UNIVERSITY OF READING

Department of Meteorology

Nonlinear data assimilation using synchronisation in a particle filter

FLÁVIA RODRIGUES PINHEIRO

A thesis submitted for the degree of Doctor of Philosophy

April 2018

DECLARATION

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Flávia R Pinheiro

PUBLICATIONS

Some of the work in Chapters 3 and 4 of this thesis has appeared in the publication:
Pinheiro et al. (2018): Pinheiro, F. R., P. J. van Leeuwen and U. Parlitz, 2018: An ensemble framework for time delay synchronization. *Quart. J. Roy. Meteorol. Soc.*, 144: 305-316, DOI: 10.1002/qj.3204.

Work contained in chapter 5 of this thesis appears in the submitted article:
Pinheiro, F. R., P. J. van Leeuwen and G. Geppert: Efficient nonlinear data assimilation using synchronisation in a particle filter (*submitted to Quart. J. Roy. Meteorol. Soc.*)

ABSTRACT

Current data assimilation methodologies still face problems in strongly nonlinear systems. Particle filters are a promising solution, providing a representation of the model 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 a useful tool to be explored. A potential choice of proposal density might come from the synchronisation theory, in which one tries to synchronise the model with the true evolution of a system using one-way coupling, via the observations, by adding an extra term to the model equations that will control the growth of instabilities transversal to the synchronisation manifold.

Efficient synchronisation is possible in low-dimensional systems, but these schemes are not well suited for high-dimensional settings. The first part of this thesis introduces a new scheme: the ensemble-based synchronisation, that can handle high-dimensional systems. A detailed description of the formulation is presented and extensive experiments in the nonlinear Lorenz96 model are performed. Successful results are obtained and an analysis of the usefulness of the scheme is made, bringing inspiration for a powerful combination with a particle filter.

In the second part, the ensemble synhronisation scheme is used as a proposal density in two different particle filters: the Implicit Equal-Weights Particle Filter and the Equivalent-Weights Particle Filter. Both methodologies avoid filter degeneracy by construction. The formulation proposed and its implementation are described in detail. Tests using the Lorenz96 model for a 1000-dimensional system show qualitatively reasonable results, where particles follow the truth, both for observed and unobserved variables. Further tests in the 2-D barotropic vorticity model were also performed for a grid of up to 16,384 variables, also showing successful results, where the estimated errors are consistent with the true errors. The behavior of the two schemes is described and their advantages and issues exposed, as this is the first comparison ever made between both filters.

The overall message is that results suggest that the combination of the ensemble synchronisation with a particle filter is a promising solution for high-dimensional nonlinear problems in the geosciences, connecting the synchronisation field to data assimilation in a very direct way.

ACKNOWLEDGEMENTS

If I had to define this PhD by a feeling, that would be gratitude. The last four years were intense, beautiful, enjoyable, hard, and incredible in so many ways that saying thanks to all the people who made this happen is never enough.

God is probably aware of how thankful I am, as I made this quite clear in our daily chats during this whole trajectory.

Two other very special people are responsible for this whole PhD:

My best friend, partner and love, Rachel, who supported me in every single second of this process, always making me feel the luckiest person in the world for having her beside me. There are no words that can express how grateful I am and she knows that.

Then, when I thought I was lucky enough, life showed me it could get better, and then I met the best supervisor a PhD student can have. The hours run fast when you are having fun and that is exactly what happened in the long, inspiring and enjoyable meetings I had with Peter Jan van Leeuwen. He transfered his passion for data assimilation to me very quickly and this PhD would be very tough without him. Thanks for sharing your knowledge and always making me feel positive until the end!

A huge thanks also goes to all those people in Brazilian Navy who helped me in many ways to start this amazing (crazy) adventure: CF Reinert, CMG Marcio Borges, Nilza, Fabiola, CMG Alenquer, CMG Carvalho, CMG Lucia, CMG Ana Claudia, CNPq for the financial support and special thanks to some others who helped me stay in this country and finish this adventure: CMG Reis Leite, Rocha and the couple Andre and Taryn (were they angels in my life?).

In the Department of Meteorology, one of the best I've ever been, my gratitude goes to Ross Reynolds, who helped me so much before coming, my Monitoring Committee, for always being so kind and helpful and, of course, all the special friends I've made in this Department. As I know I will probably be unfair if I forget someone, I will not list anyone, because they know who they are. In particular, 1U07 will always be in my heart, an office where I shared so many experiences, laughs and thoughts.

Last, but not the least, thanks to my beloved family: my mother, brother and grandma. I've always felt their support, no matter how distant we were. My friends in Brazil were also missed, but some of them were very present and supportive with their love.

This PhD is the result of being lucky to have all these people in my life. Gratitude is indeed the only word I can use to define it.

Contents

List of Figures

List of Tables

Chapter 1:

MOTIVATION

1.1 General Introduction

Numerical models are extensively used in the geosciences field to describe and predict the behaviour of physical variables through the use of mathematical equations. Most of these geophysical models are nonlinear, exhibiting a chaotic behaviour. Chaos is an existing phenomenon in nature which describes the sensitivity of a system to small changes in its initial conditions. In essence, what the theory explains is that slightly different initial conditions can lead to completely different predicted trajectories of a variable in the future (Lorenz, 1963). This effect is also known as the ”butterfly effect”. This fascinating idea was summarised by Edward Lorenz as follows:

> ”Chaos: When the present determines the future, but the approximate present does not approximately determine the future.”

Therefore, long-range predictability is a limited ability when it comes to geophysical models (Lorenz, 1969; Tribbia and Baumhefner, 2004). In numerical weather prediction, a useful technique is applied: multiple forecast runs are performed with slightly different initial conditions and physics. This way, probability is now taken into account in the process, as the different outputs will reflect how probable different events are to occur. As the majority of the forecasts tend to show a similar behaviour, then it is considered that this event has the higher probability to be closer to the true forecast to happen. An important detail in this procedure is that one needs information on the starting point of these different model runs. And in order to have multiple forecasts that are able to represent the future correctly, we need initial conditions that represent the probability distribution of the current state as good as possible.

Data assimilation is a mathematical tool that combines the information and error statistics from the observations with the prior knowledge of the model states probability, in order to obtain a better representation of the state of the system, that can also be called in this context as the posterior probability.

The most popular present-day data assimilation methodologies have their formulations commonly based in some assumptions, like e.g. considering that nature is weakly-nonlinear. In the past, this was not an issue, as the resolution of the numerical models did not allow for strongly nonlinear features such as convection, for instance. Nowadays, however, assuming an almost linear atmosphere can, in some cases, be less valid, due to the fast increase in the numerical resolution and ability of the present models to simulate nonlinear events.

An alternative methodology that is a promising solution to circumvent the non-linearity issue is a particle filter. It allows for fully nonlinear models and is able to produce samples from the full posterior probability distribution. The prior knowledge of the model states is represented by a set of model runs or particles, which are weighted by their proximity to the observations.

The very basic particle filter, however, does not work in high-dimensional cases and the reasons for that will be explained later in this thesis. The challenge during the last recent years, then, has been to improve this methodology so it can fully work in geophysical systems.

Fortunately, recent advances in particle filtering have provided several schemes, which improved its performance and applicability to geosciences. Focus has been applied in an interesting property of the particle filter's formulation: the so-called proposal density. In summary, this procedure allows one to change the model equations to bring the particles closer to the observations and consequently closer to the truth, resulting in very efficient model updates.

Towards this proposal density strategy, simple solutions could be applied, like for instance, the use of nudging. Nevertheless, solutions like these have shown to be not powerful enough to bring the particles closer to the existing observations. A potential solution might come from the synchronisation theory. The core idea in this field is to try to synchronise a model with the true evolution of a system, via the observations. In practice, this means that an extra term is added to the model equations, so it can hamper the growth of instabilities in the system.

Among the existing synchronisation approaches, the ones which use the concept of time embeddings - similar to smoothers in data assimilation - have derived remarkable results in the field. Specifically, results presented in the work from Rey et al. (2014a) brought inspirations for the topic of this thesis. Among their results, the authors performed experiments using a nonlinear toy model in which a very small proportion of the system was observed. With this setup, they achieved synchronisation between the model and the truth, reaching errors with an order of magnitude close to machine precision, i.e. 10^{-15}.

That impressive result, however, was based in a formulation that had some drawbacks. In their idealised framework, observation errors were ignored and, as soon as they were introduced, the system's performance degraded considerably. Also, that framework was computationally prohibitive for high-dimensional systems, and this will be discussed later in the present work.

1.2 Hypothesis and aims

The hypothesis considered in this thesis is the following:

- Considering that synchronisation and data assimilation are based on similar formulations, it is possible to merge them and explore the advantages of both to derive a powerful new tool towards the estimation/prediction process.

The overall aim of this thesis is to propose a fully nonlinear data assimilation method and show its potential to be used as a data assimilation scheme in the high-dimensional, chaotic models found in the geosciences field. To this end, the research contained in this thesis addresses two main questions:

1. *Is it possible to improve the existing synchronisation framework by developing a scheme which can be robust to high-dimensional systems and can also take into account realistic observation settings?*

   The synchronisation framework that motivated this work is indeed interesting, due to its powerful convergence between the model and the truth. A new scheme based on its ideas will be proposed. Investigations on how further improvements will affect the previous results will be discussed.

2. *Can we use synchronisation in a particle filter and make the whole scheme more efficient in nonlinear systems?*

   The proposal density freedom will be fully explored in this thesis. A detailed formulation will be derived to show how these two schemes can be combined. Experiments with nonlinear models will be exposed to investigate how efficient this methodology can be.

1.3 Outline

This thesis is structured as follows:

- Chapter 2 presents a brief review of the scientific background of the main data assimilation methodologies, exposing the assumptions behind these methods. A summary on the characteristics of the results produced by the variational and sequential schemes is explained. A brief overview on smoothers is also presented to prepare the reader for later comparisons with the synchronisation framework. Considerations are made on how and why the particle filters are a promising data assimilation methodology to tackle the nonlinear and non-Gaussianity problem, which is indeed a reality. Limitations of the method and the existing solutions are also exposed. Two main particle filter schemes are then described, as they will be used later in the experiments in this thesis.
- Chapter 3 gives an overview on the synchronisation phenomenon, its theory and different

methodologies. The peculiarities of the field and also its similarities with the existing data assimilation framework are presented. A description of the synchronisation framework on which this thesis is based is given and tests are performed with this scheme in a nonlinear toy model. The limitations of the scheme are exposed, giving the main motivation for the next chapter.

- Chapter 4 introduces a new scheme: the ensemble-based synchronisation. A detailed description of the formulation is presented and extensive experiments are performed to test the robustness of the scheme in a nonlinear system. An analysis of its usefulness is made, bringing inspiration for a powerful combination with a particle filter, which will be investigated in the next chapters. This chapter aims to answer the first question addressed in this thesis.
- Chapter 5 essentially presents a new nonlinear scheme, which uses the ensemble-based synchronisation in a particle filter. The formulation proposed and its implementation are described in detail. Again, a nonlinear model is used to test the efficiency of the system and an analysis of its implementation and possible improvements is made. This chapter starts to answer the second question addressed in this thesis.
- Chapter 6 aims to investigate the use of synchronisation in particle filters for larger models. A 2-dimensional model based on the barotropic vorticity equation is described and used to assess how the ensemble-based synchronisation works with two different particle filters. The behavior of the two schemes is described and their advantages and issues exposed, as a first comparison ever made between both filters. This chapter aims to close an answer for the second question addressed in this thesis.
- Chapter 7 concludes by summarising the thesis' results, while proposing suggestions for potential improvements to the system and future work to extend the research.

Chapter 2:

BACKGROUND

2.1 Introduction

Data assimilation is a complex field which aims for a ”simple” goal: improve a model output by combining it with the available observations. To this end, a state vector $x \in \Re^{D_x}$ is used to represent the state of a system and the estimate of this state is evolved forward in time using a forecast model $\mathcal{M}$:

$$x^{n+1} = \mathcal{M}(x^n) + \beta^{n+1} \tag{2.1}$$

where β^{n+1} is the stochastic part of the true evolution or the model stochastic noise. This forecast is updated while taking into account the observations of this state contained in the vector $y \in \Re^{D_y}$:

$$y = H(x) + \varepsilon \tag{2.2}$$

where $H(x)$ is the observation operator, which maps the state vector x from model space to observation space and can be linear or nonlinear. The term ε is the observation noise.

Different methodologies are currently used in the geosciences and in general they lie in two main approaches: variational and sequential methods.

Generally speaking, the first approach is based in the inverse-modelling point of view. It assumes that the prior knowledge of the model state is a Gaussian distribution and that the observation errors are Gaussian as well. The observation operator $H(x)$ can be linear or nonlinear. Then, the aim is to find the most probable state of the system, i.e. the maximum of the posterior probability density function (pdf), $p(x^n \mid y^n)$, which is also called the global mode. This can be computed by finding the minimum of the expression:

$$J(x) = -log(p(x^n \mid y^n)) \tag{2.3}$$

in which x^n and y^n are the state and observation vectors at time n respectively, and J is the so-called cost function or penalty function. Considering the Gaussian assumptions previously made, the best match of an analysis model state to a set of observations is usually found with a least-squares error minimization (Daley, 1991).

The second method in the data assimilation field involves an statistical perspective and is based in the randomness of the model and the observations, assuming that the prior knowledge of the model state is known and that the uncertainty is described by a pdf. Again, both the prior and the observation errors are assumed to be Gaussian. Bayes' theorem is used, assuming that the conditional distribution of the observations given the model state (called the likelihood) is also known. This methodology is considered a multiplication problem which has a solution defined by a full probability density function, also represented by an ensemble of model states.

During recent years, the development of the hybrid methods, which aim to gather the advantages and benefits from both approaches, has intensified, leading to a whole branch of new methodologies based on all the previous knowledge on the data assimilation methods.

In this chapter, the two main approaches will be briefly described, so the reader can have an overview of the field. Later on, focus will be given in the smoothing methods, as key connections with synchronisation will appear and will be explained in the future.

More relevant in this chapter, though, is the description of the particle filters as a data assimilation methodology. It has important advantages over the main, traditional approaches when it comes to fully nonlinear systems, as in this methodology the Gaussian assumptions assumed by the main data assimilation approaches can be relaxed. This provides one justification for the use of particle filters as a data assimilation method in nonlinear systems. The limitations and solutions of this method will be exposed and, among all the existing particle filter flavours, focus will be given on two of them: the Equivalent-Weights Particle Filter and the Implicit Equal-Weights Particle Filter.

2.2 Data assimilation

2.2.1 The problem

Data assimilation can be defined in a general way as a Bayesian inference problem, in which prior information about a system or model contained in a pdf is updated when observations, including their errors, become available.

According to Bayes' Theorem, the prior pdf $p(x^n)$, which brings all the information we know about the state without including any knowledge from the observations, and the likelihood $p(y^n \mid x^n)$, which represents the observations given the state previously known, are multiplied to form the posterior pdf $p(x^n \mid y^n)$, which is the solution to the full data-assimilation problem:

$$p(x^n \mid y^n) = \frac{p(y^n \mid x^n)p(x^n)}{p(y^n)} \tag{2.4}$$

Here, the term $p(y^n)$ is just a normalisation factor, as

$$p(y^n) = \int p(y^n, x^n)dx = \int p(y^n|x^n)p(x^n)dx \tag{2.5}$$

and so the integral of the numerator in equation 2.4. This ensures that

$$\int p(x^n|y^n)dx = 1 \tag{2.6}$$

To solve this problem, some assumptions are usually made in the data assimilation methods, such as: 1) the observations are independent; 2) the observation errors are Gaussian, unbiased and uncorrelated; 3) the prior distribution is Gaussian; 4) the numerical model is a perfect representation of the truth. The need of these assumptions in the data assimilation approaches will be explained later, together with a brief discussion on the significance of the model state generated by them under nonlinear conditions.

Another issue in data assimilation is that in most geophysical models it is expected that the observations are not taken at the exact same physical location as the model variables. The observation operator $H(x)$ is then responsible for mapping the state vector x from model space to the observation space. This can be a simple linear relationship, in which an interpolation can be used, or a complex nonlinear relation, like in the case of the radiative transfer functions. The last case is dominant in numerical weather prediction, due to the abundance of satellite observations used.

2.2.2 Variational methods

Variational methods can be three-dimensional (3D-Var) or four-dimensional (4D-Var). They are based on a maximum *a posteriori* estimate (often called MAP) approach and derive the analysis state by minimizing a cost function J. It is a function which aims to make a balance between our *a priori* knowledge of the state and the observations. In order to perform the minimization of the cost function, one can use numerical optimization techniques, which may require the gradient of the cost function. Adjoint methods provide a very efficient way to compute this gradient.

In the 3D-Var case (Lorenc, 1986), it is assumed that the background state (or prior) x_b^n and the observations y^n are obtained at the same time or over a very short time interval. Assuming that the prior and the observation errors are Gaussian distributions, the posterior is then shaped this way:

$$p(x^n \mid y^n) \propto exp\left[-\frac{1}{2}(x^n - x_b^n)^T B^{-1}(x^n - x_b^n) - \frac{1}{2}(y^n - h(x^n))^T R^{-1}(y^n - h(x^n))\right] \tag{2.7}$$

yielding the minimization of the cost function

$$J(x^n) = \frac{1}{2}(x^n - x_b^n)^T B^{-1}(x^n - x_b^n) + \frac{1}{2}(y^n - h(x^n))^T R^{-1}(y^n - h(x^n)) \tag{2.8}$$

where B and R are the background and observation error covariance matrices, respectively. We can consider 3D-Var a stationary method, as all the observations are considered at the same time. This way, B is a static, climatological covariance matrix.

If the observation operator is linear, then the likelihood is Gaussian, and so is the posterior, i.e. the model state with maximum probability is the mean of this posterior. The analysis error covariance is found by computing the second derivative of J, also called the Hessian. This covariance is actually the inverse of the local curvature of the MAP estimate, or the inverse of the Hessian and is the covariance of the full pdf, but only if we consider the assumption that it is a Gaussian distribution.

The 4D-Var case (Le Dimet and Talagrand, 1986; Talagrand and Courtier, 1987; Courtier and Talagrand, 1987) can be considered a non-stationary one, as it considers the observations over a time interval and so the system dynamics through it. This way, considering that the start of the assimilation window is at time $t = 0$, then we have the initial model state x^0, which yields the posterior:

$$p(x^0 \mid y^{1:n}) \propto p(y^{1:n} \mid x^0)p(x^0) \tag{2.9}$$

and, considering that the prior and the observation errors are still assumed to be Gaussian, the cost function is:

$$J(x) = \frac{1}{2}(x^0 - x_b^0)^T (B^0)^{-1}(x^0 - x_b^0) + \frac{1}{2}\sum_{j=1}^{n}(y^j - h(\mathcal{M}^{0\to j}(x^0)))^T R^{-1}(y^j - h(\mathcal{M}^{0\to j}(x^0))) \tag{2.10}$$

where $\mathcal{M}^{0\to j}(x^0)$ is the model equation required to propagate the initial model state x^0 to the correspondent time of the observation y^j. Note that equation 2.10 is the strong constraint version of the 4D-Var, in which it is assumed that the model is perfect and so any trajectory is fully determined by the initial state. The minimization of this cost function provides the initial condition of the model that best fits the observations. As we will need to bring the information from these observations back to the beginning of the window, backwards equations are used, in which the transposed of the model equations is integrated backwards in time through the adjoint model equations.

The previous assumptions made for the 3D-Var formulation are kept: the prior is again assumed to be Gaussian and so, for linear model equations and observation operator, the posterior will still be a Gaussian. Consequently, its mean which is the MAP estimate and the inverse of the Hessian can fully determine the pdf.

When considering the model uncertainty, we obtain the weak constraint version of the 4D-Var formulation (van Leeuwen and Evensen, 1996; Tremolet, 2006). The posterior in Bayes' theorem is now defined as:

$$p(x^{0:n} \mid y^{1:n}) \propto p(y^{1:n} \mid x^{0:n})p(x^{0:n}) \tag{2.11}$$

in which we consider the Markov assumption

$$p(x^{0:n}) = \prod_{j}^{n} p(x^j|x^{j-1})p(x^0) \tag{2.12}$$

and that the transition densities are Gaussian. The same Gaussian assumptions previously used are considered, and an extra term should now be added to the equation 2.10, taking into account the model error covariance matrix Q, which is assumed to be uncorrelated with the background and observation errors:

$$\begin{aligned} J(x) =& \frac{1}{2}(x^0 - x_b^0)^T(B^0)^{-1}(x^0 - x_b^0) + \frac{1}{2}\sum_{j=1}^{n}(y^j - h(\mathcal{M}^{0\to j}(x^0)))^T R^{-1}(y^j - h(\mathcal{M}^{0\to j}(x^0)))+ \\ &+ \frac{1}{2}\sum_{j=1}^{n}(x^j - \mathcal{M}^{0\to j}(x^0))^T Q^{-1}(x^j - \mathcal{M}^{0\to j}(x^0)) \end{aligned} \tag{2.13}$$

The minimization of the cost function now aims to determine the trajectory of states $x^{0:n}$.

An issue that arises in both of these variational methods is that, theoretically, by assuming that the likelihood does not behave as a Gaussian distribution or that the observation operator or model are not linear, we obtain a non-Gaussian posterior as a result, which can have multiple minima. The inverse Hessian, in this case, will not be able to determine the distribution, providing information on a local minimum which may not be the global one. The uncertainty, then, will be related to the error close to this local minimum, not describing the full posterior pdf properly. Also, the minimum is found using the gradient of the cost function. This gradient is computed using adjoint methods which linearise the observation operator and the model equations, as shown in the 4D-Var formulation. If the first guess is such that it is closer to the global minimum than any local minimum, then the method is likely to find the global minimum after sufficient iterations of the minimisation algorithm. If not, then the minimum found can be located anywhere around the existing local minima. In other words, as the system becomes more nonlinear, it is more probable that local minima will occur (Gauthier, 1992; Miller et al., 1994), making it more difficult to find the global minimum of the cost function.

For a unified explanation of this methodology, see Talagrand (2012).

2.2.3 Sequential methods

So far we can note that the main goal of data assimilation, from a statistical point of view, is to describe a posterior distribution as good as possible. Considering, though, that it is very hard to describe the full distribution, we aim for a good estimate of some statistical parameters such as e.g. the mean and/or the covariance. We have seen in the previous section that the MAP estimator yields the mode of the conditional distribution. Seeking for optimal estimates, among many statistical methods (Anderson and Moore, 1979; Garthwaite et al., 2002; Hogg et al., 2013; Ross, 2014), there is also the MV (minimum variance) estimator, which describes the conditional mean of the state. This estimator produces what is called the optimal interpolation (OI). OI and the Kalman Filter (KF) (Kalman, 1960) are both sequential methods.

By assuming that the model prior and the likelihood are again Gaussian distributions, both schemes find a Gaussian posterior, which can be fully defined by the normal distribution:

$$p(x^j \mid y^j) = N(x_a^j, P_a^j) \tag{2.14}$$

in which x_a^j is the mean and P_a^j is the analysis error covariance.

To find these values, a set of equations shall be used, which vary from the OI to the KF cases. The formulation for the OI method is:

$$\begin{aligned} x_a^j &= x_b^j + K(y^j - Hx_b^j) \\ K &= BH^T(HBH^T + R)^{-1} \\ P_a^j &= (I - KH)B \end{aligned} \tag{2.15}$$

whereas for the KF we have:

$$\begin{aligned} x_a^j &= x_b^j + K^j(y^j - Hx_b^j) \\ K^j &= P_f^j H^T(HP_f^j H^T + R)^{-1} \\ P_a^j &= (I - K^j H)P_f^j \end{aligned} \tag{2.16}$$

These are known as the Best Linear Unbiased Estimate (BLUE). They provide the minimum variance estimate of the model state through the following procedure: given a prior (background, or initial guess) estimate x_b at time j, we find the best update (or analysis) x_a, which is an optimal *a posteriori* estimate and a linear combination of an *a priori* estimate x_b and a weighted difference between the current observations y^j and the measurement prediction Hx_b. This difference

$$d = y^j - Hx_b^j \tag{2.17}$$

is usually called the *innovation*.

The main difference between equations 2.15 and 2.16 is that OI keeps the same static, climatological error covariance B along the time. The KF methodology, though, allows this matrix to be flow dependent, so $B = P_f^j$, which can be computed as:

$$P_f^j = MP_a^{j-1}M^T \tag{2.18}$$

considering that M is the operator responsible for the evolution of a perfect model and P_a^{j-1} is the analysis error covariance matrix at the previous observation time (at $t = 0$, it is considered $P_a^0 = B$). Also, K and K^j in the previous equations are known as the Kalman gain matrix.

To summarise all the steps in the KF methodology, we can divide the process in two main blocks:

- Propagation:
  - Generate an *a priori* or background state x_b, as $x_b^j = Mx_a^{j-1}$ (assuming the model is linear)
  - Propagate the forecast error covariance matrix P_f^j according to equation 2.18
- Update:
  - Compute the innovation d
  - Compute the Kalman gain matrix K^j
  - Update the analysis state x_a^j and the analysis error covariance matrix P_a^j

We can also assume that the model is imperfect in the KF formulation. If the model error is additive and Gaussian with zero mean, the posterior is still a Gaussian distribution. We just need to add a Q^j term to equation 2.18 to be consistent to that assumption, where Q^j is the model error covariance matrix.

A full derivation and explanation of the KF formulation can be found e.g. in Asch et al. (2016); Fletcher (2017).

From an operational perspective, however, both the OI and the KF are very computationally expensive, as one needs to store the prior error covariance matrix to feed the process and also because of the inversion contained in the Kalman gain formulation. Also, the Gaussian assumptions made for the prior, the likelihood and consequently the posterior allow the analysis state x_a^j to be considered an optimal result, as it represents the mean of the posterior. When migrating to nonlinearities in the model dynamics and the observation operator, in which case the

Gaussianity of the system gets compromised, adapted formulations like the Extended Kalman Filter (EKF) (Gelb, 1974) exist. They do allow for weakly nonlinear numerical models and observation operators. However, x_a^j cannot be considered optimal anymore, and so its meaning gets unclear.

Additionally, the size of the propagated covariance matrix in 2.18 in the EKF formulation makes it not feasible for geosciences applications. An alternative then appeared, to avoid the explicit propagation of the covariance matrix and instead estimate it, by using a sample of evolved states, also called ensemble. This alternative approach is the Ensemble Kalman Filter (EnKF) (Evensen, 1994, 2003).

By using Monte Carlo methods, the idea is that a cloud of states or points in state space can be used to represent a pdf. With this approximation, one can evolve each ensemble member i forward in time using a nonlinear model with additive noise:

$$x_i^j = Mx_i^{j-1} + \beta_i^j \tag{2.19}$$

and then compute approximate estimates for the first and second moments of the pdf. The ensemble mean, which is considered the best guess estimate, is defined as

$$\overline{x^j} = \frac{1}{N_{ens}} \sum_{i=1}^{N_{ens}} x_i^j \tag{2.20}$$

where N_{ens} is the number of ensemble members. And, by defining

$$X_f = \frac{1}{\sqrt{N_{ens} - 1}} (x_i^j - \overline{x^j}) \tag{2.21}$$

the ensemble covariance is:

$$P_f^j = X_f(X_f)^T = \frac{1}{N_{ens} - 1} \sum_{i=1}^{N_{ens}} (x_i^j - \overline{x^j})(x_i^j - \overline{x^j})^T \tag{2.22}$$

which indeed substitutes the evaluation performed in the KF for the equation 2.18. The ensemble spread defines the error variance.

The EnKF presents the following advantages:

- a flow-dependent *a priori* uncertainty is considered;
- easy implementation;
- no model linearisations are needed;
- account for model error;
- explicit covariance never has to be computed, as all manipulations can be done in ensemble

space.

Some issues are still present, though:

- the ensemble size is a sensitive factor in the system, as undersampling can occur and lead to the divergence of the filter;
- techniques like localisation and inflation should be applied to alleviate the filter divergence;
- Gaussian statistics are assumed, which may not be valid for nonlinear systems.

2.3 Smoothers

The smoothing problem consists in estimating a state or a set of states, given past, present and future observations. The posterior pdf, then, can be described as

$$p(x^j \mid y^{1:k}) \tag{2.23}$$

for k fixed and $k > j > 1$.

In another case, the posterior can be a joint pdf, where the states to be estimated form a time series, from a time index 0 to k, for instance:

$$p(x^{0:k} \mid y^{1:k}) \tag{2.24}$$

which is actually called a fixed-interval smoother, as the time interval is fixed. Using Bayes' Theorem and considering the Markov property of the system, we obtain

$$p(x^{0:k} \mid y^{1:k}) \propto p(x^{0:k-1} \mid y^{1:k-1})p(x^k \mid x^{k-1})p(y^k \mid x^k) \tag{2.25}$$

A similar formulation is the fixed-lag smoother, in which

$$p(x^{k-L:k} \mid y^{1:k}) \tag{2.26}$$

Here L is called the lag and k takes consecutive values; each state is estimated using the past, present and the same number of L future observations. That is the variant that will be of interest in this thesis, as it is comparable to the synchronisation framework.

There are several optimal linear smoothers which have been explored in geophysical applications (Cosme et al., 2012) like: the fixed-interval sequential smoother, the fixed-lag smoother, the ensemble smoother, the forward-backward smoother, the two-filter smoother, among others (Cohn et al., 1994; van Leeuwen and Evensen, 1996; van Leeuwen, 1999; Lermusiaux and Robinson, 1999; Lermusiaux et al., 2002; Cosme et al., 2010). They are all based on the KF formulation, and the main difference lies in the way the cross covariances in time are handled, while taking the future observations into account.

2.3.1 Kalman Smoother (KS)

The KS formulation is a generalization of the KF, but allowing for observations in a certain time window to be considered in the assimilation process, including future observations (Mènard and Daley, 1996).

Let us consider the case in which the time interval of the observation window is the same for each model state, i.e. the fixed-lag smoother. Define the forecast state trajectory vector as

$$\hat{x}_b^j = ((x^j)^T, (x^{j+1})^T, \cdots, (x^k)^T)^T \tag{2.27}$$

where k is the fixed-lag window. The full observation operator working on this vector is then

$$\hat{H} = diag(H^j, H^{j+1}, \cdots, H^k) \tag{2.28}$$

Note that $\hat{H}$ is a block diagonal matrix. The full observation vector can be defined as

$$y = ((y^j)^T, (y^{j+1})^T, ..., (y^k)^T)^T \tag{2.29}$$

Then we can write the update of the full trajectory vector as

$$\hat{x}_a^j = \hat{x}_b^j + P\hat{H}^T \left(\hat{H}P\hat{H}^T + R\right)^{-1} (y - \hat{H}\hat{x}_b^j) \tag{2.30}$$

where the full trajectory covariance matrix P has a block structure over the whole window, such as:

$$P = \begin{pmatrix} P^{j,j} & P^{j,j+1} & ... & P^{j,k} \\ ... & ... & ... & ... \\ P^{k,j} & P^{k,j+1} & ... & P^{k,k} \end{pmatrix} \tag{2.31}$$

where:

$$P^{j,j} = \overline{(x_b^j - \overline{x_b^j})(x_b^j - \overline{x_b^j})^T} \tag{2.32}$$

and if we apply the linearised model

$$M^{j:k} = M^k M^{k-1} ... M^j \tag{2.33}$$

at the covariance $P^{j,j}$ from time j to time k, we obtain the covariance matrix related to the whole time lag, $P^{k,k}$:

$$P^{k,k} = M^{j:k} P^{j,j} {M^{j:k}}^T \tag{2.34}$$

2.4 Particle filters

2.4.1 Basic particle filter

The main structure of the particle filters is very similar to the EnKF. Both methods represent the pdfs by an ensemble of model states (or particles), which are propagated forward in time using the model equations. In both cases, model errors can be included. The main difference between them lies on the fact that the EnKF relies on the assumptions that the prior and the likelihood are Gaussian and the observation operator is linear. This way, each model state is updated so the ensemble converges to the Gaussian posterior. These assumptions can be relaxed in the particle filters. Each particle is weighted according to the likelihood related to each particle and the linearity of the observation operator and the Gaussianity of the observation distributions are not necessarily needed, although they are often assumed in practice. The model equations are totally allowed to be nonlinear in the particle filter framework, which leads to a non-Gaussian prior. Model errors are included and actually essential to many of the particle filter formulations.

Another advantage of the particle filters is that it is a methodology which does not require an estimate of the background or prior model state error, an important covariance matrix that appears in all data assimilation methodologies described in sections 2.2 and 2.3. Results for the variational methods e.g., are quite sensitive to the quality of the covariance matrix B. This is also true for the KF and the EnKF. For the latter, this is actually the reason why so much effort has been put into inflation and localisation. Particle filters only use the model error covariance matrix Q. Additionally, when new observations are available, only the particle weights are recomputed, while the particles themselves are propagated forward in time without being modified, which makes this methodology more consistent to dynamical balances that may occur in the system.

This way, a particle filter represents the prior distribution by an ensemble of model trajectories propagated forward in time and each of these trajectories is a sample from the prior pdf. By increasing the number of particles or trajectories the representation gets closer and closer to the true prior distribution.

Considering again Bayes' Theorem, the posterior can be defined as:

$$p(x^{0:n} \mid y^{1:n}) = \frac{p(y^{1:n} \mid x^{0:n}) p(x^{0:n})}{p(y^{1:n})} \tag{2.35}$$

where $x^{0:n}$ represents the model trajectories over a period of n time steps and $y^{1:n}$ the observations, at every time step, for instance. The likelihood $p(y^{1:n} \mid x^{0:n})$ is usually considered a

Gaussian distribution, but this is not a mandatory assumption in the particle filter framework. The prior $p(x^{0:n})$ is the distribution of the model trajectories before the measurements were made.

If we consider that the observations are independent with time and conditional to the model states, i.e. the probability of an observation at time j, given a model trajectory, depends only on the model state at that time step, we note that:

$$p(y^j \mid x^{0:n}) = p(y^j \mid x^j) \tag{2.36}$$

and then the likelihood in equation 2.35 becomes:

$$p(y^{1:n} \mid x^{0:n}) = p(y^n \mid x^n)p(y^{n-1} \mid x^{n-1})...p(y^1 \mid x^1) \tag{2.37}$$

This way, the posterior density can be described as:

$$\begin{aligned} p(x^{0:n} \mid y^{1:n}) =& \frac{p(y^{1:n} \mid x^{0:n})p(x^{0:n})}{p(y^{1:n})} \\ =& \frac{p(y^n \mid x^n)}{p(y^n)}\frac{p(y^{n-1} \mid x^{n-1})}{p(y^{n-1})}...\frac{p(y^1 \mid x^1)}{p(y^1)}p(x^{0:n}) \end{aligned} \tag{2.38}$$

As the basic idea is to represent the probability density that contains information from the model using Monte Carlo methods, the prior pdf is represented as a discrete set of particles i or a sum of delta functions:

$$p(x^{0:n}) \approx \frac{1}{N}\sum_{i=1}^{N}\delta(x^{0:n} - x_i^{0:n}) \tag{2.39}$$

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

$$\begin{aligned} p(x^{0:n} \mid y^{0:n}) &= \sum_{i=1}^{N}\frac{p(y^n \mid x_i^n)}{p(y^n)}\frac{p(y^{n-1} \mid x_i^{n-1})}{p(y^{n-1})}...\frac{p(y^1 \mid x_i^1)}{p(y^1)}\frac{1}{N}\delta(x^{0:n} - x_i^{0:n}) \\ &= \sum_{i=1}^{N} w_i^n w_i^{n-1}...w_i^1\frac{1}{N}\delta(x^{0:n} - x_i^{0:n}) \end{aligned} \tag{2.40}$$

where the weights w_i^j are:

$$w_i^j = \frac{p(y^j \mid x_i^j)}{p(y^j)} = \frac{Np(y^j \mid x_i^j)}{\sum_{i=1}^{N} p(y^j \mid x_i^j)} \tag{2.41}$$

and the sum in the denominator ensures normalized weights. These weights, w_i^j, are the relative weights for each particle i, representing their importance in the posterior distribution.

For a more detailed mathematical explanation on particle filters, see Doucet et al. (2001).

2.4.2 Limitations and solutions

The weights in equation 2.41 are attached to each particle to give them a relative importance, according to how close or distant the model states are from the observations. The product of the weights over all time steps is very sensitive to these distances and can become very low, even if just one particle departs from the observations at a single time step. After some few more time steps, only one particle will be close to the observations and receive non-zero weights, whereas all the other particles will receive negligible weights (Doucet et al., 2001; Gordon et al., 1993). The posterior pdf becomes a delta function centered on the state trajectory of the particle which received the significant weight, meaning that the statistical information contained in the ensemble and so the ability of the particle filter to represent the posterior pdf are lost. This is called filter degeneracy or filter collapse and it happens in even small-dimensional systems.

Several methods have been developed to avoid this problem, based on resampling. In this technique, at each observation time the particles which receive low weights are discarded and the ones with higher weights are kept and duplicated, so the ensemble remains with the same number of particles, with same weights. These particle filters are also called sequential importance resampling (SIR) filters. Different implementations of resampling do exist, such as in Gordon et al. (1993); Kitagawa (1996); Lui and Chen (1998), aiming to reduce the variance in the weights. To this end, other schemes like in Klaas et al. (2005); Wikle and Berliner (2007); Berliner and Wikle (2007) have also been developed. Other methodologies, focusing on keeping just the first and second order moments of the posterior, relaxing the need of higher moments, have also been tested (Xiong et al., 2006; Kim et al., 2003; Eyink and Kim, 2006; Nakano et al., 2007). Snyder et al. (2008, 2015) proved, however, that resampling would not solve the degeneracy issue, even using the so-called optimal proposal density (see e.g. Doucet et al. (2000)).

Still aiming to prevent filter degeneracy to occur, other techniques such as van Leeuwen (2002); Spiller et al. (2008) focus on the improvement of the weights of the majority of the particles.

Actually, to avoid that the particles become 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: 1) guarantee that the particles are located at the high-probability regions of the posterior pdf and 2) ensure similar or equal weights to the particles, so they will never collapse. To this end, several methods have been recently developed, such as the ones described in this thesis: the EWPF (Equivalent-Weights Particle Filter, van Leeuwen (2010)) and the IEWPF (Implicit Equal-Weights Particle Filter, Zhu et al. (2016)).

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

2.4.3 Proposal densities

As mentioned earlier, each of the model trajectories in a particle filter is a sample from the prior pdf. The idea of the proposal densities is to sample from a proposed pdf. This way, one can change the trajectories of the particles, and so leading them to end up closer to the observations. By sampling from a proposed density distribution, we affect the particle weights in the posterior, ensuring that it still converges to the true posterior pdf, as the number of particles is not reduced.

Schemes like the optimal proposal density, the implicit particle filter (Chorin and Tu, 2009; Chorin et al., 2010; Morzfeld et al., 2012) and the EWPF use the idea of applying the proposal density to affect the trajectories of the particles by using information from future observations.

The proposal density theory can be explained as follows: considering the Markov property of the model, we can write Bayes' Theorem as

$$p(x^{0:n} \mid y^{0:n}) = \frac{p(y^n \mid x^n)p(x^n \mid x^{n-1})}{p(y^n)} p(x^{0:n-1} \mid y^{1:n-1}) \tag{2.42}$$

where $p(x^n \mid x^{n-1})$ is the transition density of the original model. By integrating the past out we obtain:

$$p(x^n \mid y^{0:n}) = \frac{p(y^n \mid x^n)}{p(y^n)} \int p(x^n \mid x^{n-1}) p(x^{n-1} \mid y^{1:n-1}) dx^{n-1} \tag{2.43}$$

We have the freedom to introduce a proposal transition density $q(x^n \mid x^{n-1}, y^n)$ as follows, without changing the previous equation:

$$p(x^n \mid y^{0:n}) = \frac{p(y^n \mid x^n)}{p(y^n)} \int \frac{p(x^n \mid x^{n-1})}{q(x^n \mid x^{n-1}, y^n)} q(x^n \mid x^{n-1}, y^n) p(x^{n-1} \mid y^{1:n-1}) dx^{n-1} \tag{2.44}$$

This is possible if the support of $q(x^n \mid x^{n-1}, y^n)$ is equal to or larger than the one in $p(x^n \mid x^{n-1})$. This way, we avoid dividing the previous equation by zero. Regarding the last term inside the integral which corresponds to the previous analysis, by assuming that we have equal-weight particles we can obtain

$$p(x^{n-1} \mid y^{1:n-1}) = \sum_{i=1}^{N} \frac{1}{N} \delta(x^{n-1} - x_i^{n-1}) \tag{2.45}$$

and so equation 2.44 is reduced to:

$$p(x^n \mid y^{0:n}) = \sum_{i=1}^{N} \frac{1}{N} \frac{p(y^n \mid x^n)}{p(y^n)} \frac{p(x^n \mid x_i^{n-1})}{q(x^n \mid x_i^{n-1}, y^n)} q(x^n \mid x_i^{n-1}, y^n) \tag{2.46}$$

Instead of drawing from $p(x^n \mid x_i^{n-1})$ and so using the original model, we now draw from

$q(x^n \mid x_i^{n-1}, y^n)$, a modified model. So, for instance, if we have a modified model such as

$$x^n = g(x^{n-1}, y^n) + \eta^n \tag{2.47}$$

where η^n is a Gaussian distribution $\eta^n \sim N(0, Q)$, the proposal density in this case can be

$$q(x^n \mid x^{n-1}, y^n) = N(g(x^{n-1}, y^n), Q) \tag{2.48}$$

where the mean is the deterministic part of the model proposed and the covariance is the stochastic term in this model. Note that there is no obligation to assume that the proposed transition density q is a Gaussian, although it is assumed here.

Sampling from the proposal density leads to the following:

$$p(x^n \mid y^{0:n}) = \sum_{i=1}^{N} \frac{1}{N} \frac{p(y^n \mid x_i^n)}{p(y^n)} \frac{p(x_i^n \mid x_i^{n-1})}{q(x_i^n \mid x_i^{n-1}, y^n)} \delta(x^n - x_i^n) \tag{2.49}$$

This way, the posterior at time n is now defined as:

$$p(x^n \mid y^{1:n}) = \sum_{i=1}^{N} w_i \delta(x^n - x_i^n) \tag{2.50}$$

where w_i are the weights

$$w_i \propto p(y^n \mid x_i^n) \frac{p(x_i^n \mid x_i^{n-1})}{q(x_i^n \mid x_i^{n-1}, y^n)} \tag{2.51}$$

($1/N^2$ and $p(y^n)$ were dropped, considering that they are the same for all particles). The weights are then accumulated during the m time steps between observations until time step n as:

$$w_i^j \propto p(y^n \mid x_i^n) \prod_{j=n-m+1}^{n} \frac{p(x_i^j \mid x_i^{j-1})}{q(x_i^j \mid x_i^{j-1}, y^n)} \tag{2.52}$$

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

A summary on particle filters and proposal densities can be found e.g. in van Leeuwen et al. (2015).

2.4.4 Equivalent-Weights Particle Filter (EWPF)

The EWPF is a particle filter scheme which ensures that the majority of the particles receives equivalent-weights, while they keep close to the observations. This way, the scheme prevents filter degeneracy, regardless the system dimension. It is a fully nonlinear scheme, which represents the full posterior pdf.

The scheme consists of two different steps: 1) a relaxation proposal density to bring the particles closer to the observations and 2) an equivalent-weights proposal density to attribute equivalent-weights to the particles and so prevent degeneracy. The second step is crucial to the success of the scheme, as just keeping the particles close to the observations and so sampling from the high probability region of the posterior is not enough. The posterior actually consists of a combination of the position of the particles and their weights. The equivalent-weights step will then guarantee the significance of the majority of the particles in representing this posterior, obtaining the best statistical information possible.

Regarding the relaxation proposal density, the main idea of this thesis is to use synchronisation as this proposal. The complete formulation of it will be described later in chapter 5. In this section, the original relaxation proposal used in Ades (2013) will be briefly described.

Assume that the states are evolved according to:

$$x_i^j = f(x_i^{j-1}) + \beta_i^j \tag{2.53}$$

where β_i^j is an additive, stochastic error in the model equation. The distribution of this term can be quite general, but here it is assumed to be Gaussian: $N(0, Q)$. This way, the transition density can be defined as:

$$p(x_i^j \mid x_i^{j-1}) = N(f(x_i^{j-1}), Q) \tag{2.54}$$

For the relaxation proposal density chosen for the EWPF in Ades (2013), the new model state at time j is assumed to be:

$$x_i^j = f(x_i^{j-1}) + B(\tau)[y^n - h(x_i^{j-1})] + \hat{\beta}_i^j \tag{2.55}$$

and, consequently, the proposal density is defined following equation 2.48 as:

$$q(x_i^j \mid x_i^{j-1}, y^n) = N(f(x_i^{j-1}) + B(\tau)[y^n - h(x_i^{j-1})], \hat{Q}) \tag{2.56}$$

where the covariance $\hat{Q}$ can be freely chosen, and it is assumed here that it is the same as Q. The term $B(\tau)[y^n - h(x_i^{j-1})]$ works as a slight relaxation at every time step to pull the particles towards the observations at time n and $B(\tau)$ is a strength factor in this relaxation, dependent on the distance in time τ to the next observation. In Ades (2013), this factor was chosen to be:

$$B(\tau) = b\tau Q H^T R^{-1} \tag{2.57}$$

where b is a scaling factor that controls the relaxation strength of each particle towards the observations and in that work the authors defined $b = 0.2$. The constant τ is defined as

$$\tau = (t^j - t^0)/(t^n - t^0) \tag{2.58}$$

so it is zero at the previous observation time t^0 and increases linearly to 1 at the new observation time t^n.

As mentioned earlier, between the observations, the relaxation proposal is used to pull the particles towards the observations. But in the time step immediately before the next observation the equivalent-weights proposal is used. The final weight at time n is then

$$w_i^n \propto p(y^n \mid x_i^n) \frac{p(x_i^n \mid x_i^{n-1})}{q(x_i^n \mid x_i^{n-1}, y^n)} w_i^{n-1} \tag{2.59}$$

where w_i^{n-1} are the weights obtained from equation 2.52. The final proposal $q(x_i^j \mid x_i^{j-1}, y^n)$ is used to find a target weight w^{targ}:

$$w^{targ} = p(y^n \mid x_i^\star) p(x_i^\star \mid x_i^{n-1}) w_i^{n-1} \tag{2.60}$$

Assume that the likelihood and the model error are Gaussian, then we can solve the previous equation for $x_i^\star$ as:

$$\begin{aligned} -log(w^{targ}) =& \frac{1}{2}(y^n - H(x_i^\star))^T R^{-1} (y^n - H(x_i^\star)) + \\ & \frac{1}{2}(x_i^\star - f(x_i^{n-1}))^T Q^{-1} (x_i^\star - f(x_i^{n-1})) - log(w_i^{n-1}) \end{aligned} \tag{2.61}$$

In the case of non-Gaussian likelihood and model error, this equation will naturally change, but can still be solved through iterative methods, assuming it is still possible to evaluate a sample from the density chosen.

Ideally, each particle should have the maximum possible weight, so

$$w_i^{targ} = w_i^{max} \tag{2.62}$$

but w_i^{max} is not enough to prevent filter degeneracy, as it may vary significantly in high-dimensional systems (Ades and van Leeuwen, 2012). Considering that no particle can receive a weight greater than its maximum, a common value for w^{targ} is then computed: it is equal to or less than the w_i^{max} of a chosen percentage of particles. The remaining particles for which it is not possible to solve 2.61 with this final value of w^{targ} are ignored and resampled later.

The maximum weight for each particle is then solved as:

$$w_i^{max} = w_i^{n-1} exp\left[-\frac{1}{2}(y^n - Hf(x_i^{n-1}))^T(HQH^T + R)^{-1}(y^n - Hf(x_i^{n-1}))\right] \tag{2.63}$$

and then they are sorted from highest to lowest. The target weight is chosen equal to one of these maximum weights, such that equation 2.61 can be solved for a selected percentage of particles. These particles with $w_i^{max} \geq w^{targ}$ will have equivalent weights in the posterior and the remaining ones for which equation 2.61 cannot be solved, will be resampled.

After choosing w^{targ}, one of the possible solutions to equation 2.61 can be:

$$x_i^\star = f(x_i^{n-1}) + \alpha_i K(y^n - Hf(x_i^{n-1})) \tag{2.64}$$

where

$$K = QH^T(HQH^T + R)^{-1} \tag{2.65}$$

The factor α_i is used to ensure equal weights to the chosen percentage of particles and solves the quadratic equation 2.61. It is given by:

$$\alpha_i = 1 + \sqrt{1 - b_i/a_i} \tag{2.66}$$

where

$$a_i = 1/2(y^n - Hf(x_i^{n-1}))^T R^{-1} HK(y^n - Hf(x_i^{n-1})) \tag{2.67}$$

and

$$b_i = 1/2(y^n - Hf(x_i^{n-1}))^T R^{-1}(y^n - Hf(x_i^{n-1})) + log(w_i^{n-1}) - log(w^{targ}) \tag{2.68}$$

We note the similarity of equation 2.64 to the EnKF method. The main difference is the use of the model error covariance matrix Q in the K gain. It should also be noted in this equation that the case of $\alpha_i > 1$ yields a change of model state greater than that generated by the EnKF. Another difference is that the EnKF uses perturbed observations on each particle, whereas in equation 2.64 y^n is related to the present observation.

One of the main requirements a proposal density must follow is that its support should be at least equal to that of the model prior. As equation 2.64 is a purely deterministic solution to equation 2.61, this leads to a proposal density which is a delta function, resulting in a division by zero in equation 2.59. Then, a random noise is added to the deterministic equation as:

$$\begin{aligned} x_i^n &= f(x_i^{n-1}) + \alpha_i K(y^n - Hf(x_i^{n-1})) + \beta_i^n \\ &= x_i^\star + \beta_i^n \end{aligned} \tag{2.69}$$

Ideally, the random noise should not move the state far from $x_i^\star$ and should also help to ensure the particle weights keep close to equal. The error β_i^n can be sampled from $N(0, Q)$ through

$$\beta_i^n = Q^{1/2}\eta_i^n \tag{2.70}$$

where $\eta_i^n \sim N(0, I)$ and I is the identity matrix. However, Ades and van Leeuwen (2012) have shown that by adding a Gaussian noise to the equation, filter degeneracy occurs. An useful choice could be a uniform distribution $\eta_i^n \sim \tilde{U}(0, \gamma_U)$ with a width γ_U over a small interval. In this case, considering a sample from a D_x-dimensional multivariate uniform distribution, we have for each element j of the vector η_i^n

$$\eta_{i,j}^n \sim U[-\gamma_U, \gamma_U], \quad \forall j = 1, ..., D_x \tag{2.71}$$

where $U[-\gamma_U, \gamma_U]$ is the standard uniform distribution. Thus, each element of η_i^n is drawn from:

$$U[-\gamma_U, \gamma_U] = \begin{cases} \frac{1}{2\gamma_U} & \text{if} \quad \eta_{i,j}^n \in [-\gamma_U, \gamma_U] \\ 0 & \text{otherwise} \end{cases} \tag{2.72}$$

As each of the elements in η_i^n is independent, then

$$\tilde{U}_k(0, \gamma_U) = \begin{cases} \frac{1}{(2\gamma_U)^{D_x}} & \text{if} \quad \eta_{i,j}^n \in [-\gamma_U, \gamma_U], \quad \forall j = 1, ..., D_x \\ 0 & \text{otherwise} \end{cases} \tag{2.73}$$

The vector η_i^n is then multiplied by $Q^{1/2}$, following equation 2.70, to introduce covariances between its elements. Unfortunately, as explained in Ades (2013), similar to the issue of the delta function in the deterministic solution in 2.64, choosing $q(x^n \mid x_i^{n-1}, y^n)$ as a uniform distribution does not have support equal to or greater than the Gaussian distribution $p(x^n \mid x_i^{n-1})$, leading again to a division by zero.

A solution to this problem is to draw η_i^n from a mixture density, i.e. a combination between a uniform and a Gaussian distribution, so that:

$$\eta_i^n \sim (1-\epsilon)\tilde{U}_k(0, \gamma_U) + \epsilon N(0, \gamma_N I) \tag{2.74}$$

where ϵ is a proportional factor which controls which distribution is mainly used. By using a small ϵ, we ensure that most of η_i^n are drawn from the uniform density. Additionally, if ϵ is related to the ensemble size, e.g. $\epsilon = 0.001/N_{ens}$, this will mean that drawing η_i^n from the Gaussian tails is unlikely, even if the ensemble size increases. This way, equivalent weights can be guaranteed for all particles. However, according to Ades (2013), unlike using a purely uniform distribution, the possibility of drawing from the Gaussian tails guarantees the support needed over the entire space of x_i^n.

In summary, this mixture density, combined with the values chosen for ϵ, γ_U and γ_N are designed to ensure that the weight of each particle will be predominantly inside the probability coming from the part of the proposal in which the uniform distribution is non-zero. To sample from this mixture density, a value u is draw from $u \sim U[0,1]$. If $u < \epsilon$, then η_i^n is sampled from the normal distribution, otherwise it is sampled from the uniform one.

To find the final weight, we come back to equation 2.59 where the numerator is solved as:

$$\begin{aligned} w_i^{n-1} p(y^n \mid x_i^n) p(x_i^n \mid x_i^{n-1}) =& w_i^{n-1} exp[-\frac{1}{2}(x_i^n - f(x_i^{n-1}))^T Q^{-1}(x_i^n - f(x_i^{n-1})) \\ & - \frac{1}{2}(y^n - H(x_i^n))^T R^{-1}(y^n - H(x_i^n))] \end{aligned} \tag{2.75}$$

and the denominator is:

$$\frac{1}{q(x_i^n \mid x_i^{n-1}, y^n)} = \left[1 + \frac{\epsilon}{(1-\epsilon)} \left(\frac{2}{\pi}\right)^{D_x/2} \frac{\gamma_U^{D_x}}{\gamma_N} exp\left(-\frac{1}{2}(\eta_i^n)^T(\gamma_N^2 I)^{-1}\eta_i^n\right)\right]^{-1} \tag{2.76}$$

if $-\gamma_U \leq \eta_{i,j} \leq \gamma_U$ and

$$\frac{1}{q(x_i^n \mid x_i^{n-1}, y^n)} = \left[\frac{\epsilon}{(1-\epsilon)} \left(\frac{2}{\pi}\right)^{D_x/2} \frac{\gamma_U^{D_x}}{\gamma_N} exp\left(-\frac{1}{2}(\eta_i^n)^T(\gamma_N^2 I)^{-1}\eta_i^n\right)\right]^{-1} \tag{2.77}$$

otherwise.

The weights are then normalized by the sum of all particle weights and resampled, so they all have weight equal to $1/N_{ens}$. Particles with smaller weights which lie outside the chosen percentage of particles are resampled as duplicated copies of the particles that remained. The new ensemble is now ready to be propagated forward in time until the next observation.

Note that in this thesis parameters like γ_U, γ_N and ϵ which compose the mixture density received the same values used in Ades (2013): $\gamma_U = \gamma_N = 10^{-5}$ and $\epsilon = 0.001/N_{ens}$. More details on the EWPF scheme can be found in Ades (2013).

The appendix D contains a description on how to code the EWPF step.

2.4.5 Implicit Equal-Weights Particle Filter (IEWPF)

As mentioned before, one of the main issues of the particle filters is that particles tend to collapse in large-dimensional systems (also known as curse of dimensionality or degeneracy). The Implicit Equal-Weights Particle Filter - IEWPF (Zhu et al., 2016) does not suffer from degeneracy by construction. The extra cost is that, like the EWPF, it introduces biased estimates, although numerical experiments have shown that the bias is small compared to the Monte Carlo noise. This bias happens because, as the number of particles increases, the target weight becomes smaller and smaller, so the particles end up further and further away from the posterior mode.

The main idea is to draw the samples such that each particle has a weight equal to a target weight. One way to do that is to sample implicitly, e.g. draw from a Gaussian and transform that sample to a sample in state space with weight equal to the target weight. The implicit sampling is at the heart of the implicit particle filtering.

A brief explanation of the IEWPF scheme is as follows: considering that the model is Markovian, then the numerator in equation 2.51 for the weights can be defined as

$$p(y^n \mid x_i^n)p(x_i^n \mid x_i^{n-1}) = p(x_i^n \mid x_i^{n-1}, y^n)p(y^n \mid x_i^{n-1}) \tag{2.78}$$

and so the particle weights become:

$$w_i = \frac{p(x_i^n \mid x_i^{n-1}, y^n)p(y^n \mid x_i^{n-1})}{p(y^n)q(x_i^n \mid x_i^{n-1}, y^n)} \tag{2.79}$$

The IEWPF scheme basically draws samples implicitly from a Gaussian distributed proposal $q(\xi_i)$ instead of the original $q(x_i^n \mid x_i^{n-1}, y^n)$ (Chorin and Tu, 2009), through the following relation:

$$q(x_i^n \mid x_i^{n-1}, y^n) = q(\xi_i)\left\|\frac{d\xi}{dx^n}\right\| \tag{2.80}$$

where the derivative in this equation represents the absolute value of the determinant of the Jacobian matrix for the transformation $x_i = g(\xi_i)$. Function g can be defined as:

$$x_i^n = x_i^a + \alpha_i^{1/2}P^{1/2}\xi_i^n \tag{2.81}$$

where x_i^a is the mode of $p(x_i^n \mid x_i^{n-1}, y^n)$, P is the width of this pdf and α_i is a scalar.

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^{targ}, i.e.:

$$w_i(\alpha_i) = w^{targ} \tag{2.82}$$

The particle weights are then equalized as:

$$w_i = \frac{1}{N_{ens}} \frac{p(x_i^n \mid x_i^{n-1}, y^n)p(y^n \mid x_i^{n-1})}{p(y^n)q(x_i^n \mid x_i^{n-1}, y^n)} = w^{targ} \tag{2.83}$$

Sampling implicitly from $q(\xi)$ the weights equation becomes:

$$w_i = \frac{p(x_i^n \mid x_i^{n-1}, y^n)p(y^n \mid x_i^{n-1})}{q(\xi_i)}\left\|\frac{dx^n}{d\xi}\right\| w_i^{n-1} \tag{2.84}$$

where w_i^{n-1} is related to the weight from previous time steps, as in equation 2.52 and $q(\xi)$ is sampled from a Gaussian distribution. The determinant of the Jacobian $\|\frac{dx^n}{d\xi}\|$ can be obtained from equation 2.81:

$$\left\|\frac{dx^n}{d\xi}\right\| = \left\|\alpha_i^{1/2}P^{1/2} + P^{1/2}\xi_i^n\frac{\partial\alpha_i^{1/2}}{\partial\xi_i^n}\right\| \tag{2.85}$$

Taking the logarithm of equation 2.84 leads to:

$$-2\log w_i = -2\log w_i^{n-1} + \left[-2\log\left(\frac{p(x_i^n \mid x_i^{n-1}, y^n)p(y^n \mid x_i^{n-1})}{q(\xi_i)}\left\|\frac{dx^n}{d\xi}\right\|\right)\right] \tag{2.86}$$

After some maths, Zhu et al. (2016) derived a final formulation for these weights:

$$-2\log w_i = -2\log w_i^{n-1} + \epsilon_i\gamma_i + \phi_i - 2\log\left(\alpha_i^{N_x/2}\left\|P^{1/2}\right\|\left|1 + \frac{\partial\alpha_i^{1/2}}{\partial\xi_i^n}\frac{\xi_i^n}{\alpha_i^{1/2}}\right|\right) \tag{2.87}$$

where N_x is the dimension of the system and

$$\gamma_i = (\xi_i^n)^T\xi_i^n \tag{2.88}$$

$$\alpha_i = 1 + \epsilon_i \tag{2.89}$$

$$\phi_i = (y^n - Hf(x_i^{n-1}))^T(HQH^T + R)^{-1}(y^n - Hf(x_i^{n-1})) \tag{2.90}$$

Note that this is the case where the observation and model errors are assumed to be Gaussian and the observation operator is considered linear.

By setting the weights of all particles equal to w^{targ} makes the term on the left-hand side of equation 2.87 to be equal to a constant C. This leads us to:

$$0 = -C - 2\log w_i^{n-1} + \epsilon_i\gamma_i + \phi_i - 2\log(\alpha_i^{N_x/2}) - 2\log\|P^{1/2}\| - 2\log\left(\left|1 + \frac{\partial\alpha_i^{1/2}}{\partial\xi_i^n}\frac{\xi_i^n}{\alpha_i^{1/2}}\right|\right) \tag{2.91}$$

where the term $-2\log\|P^{1/2}\|$ can also be absorbed in C.

Considering that this is an extremely hard implicit equation to solve, Zhu et al. (2016) considered the equation in the limit of a high-dimensional N_x and integrated it in this limit, deriving a much simpler formulation:

$$\epsilon_i\gamma_i - C - 2\log w_i^{n-1} + \phi_i - N_x\log(1 + \epsilon_i) = 0 \tag{2.92}$$

The authors point out that analytical solutions exist for this formulation and use the so-called Lambert W function to solve it. Among some specificities of this function, one of its characteristics is that there are two real solutions for ϵ_i: a positive and a negative one. In their work, they tested different percentages of positive ϵ_i, for instance: $100\%, 50\%$ and 0%, where the 50% case was considered the ideal, in which ϵ_i is to be randomly chosen positive or negative, with equal probability.

For a more detailed description of the IEWPF, see Zhu et al. (2016).

Like in the EWPF, this scheme is to be used at the time step immediately before the observations. During the multiple time steps that may occur between observations, a relaxation proposal density is used to pull the particles closer to these observations. Zhu et al. (2016) used the same relaxation proposal explored in the EWPF by Ades and van Leeuwen (2012). As mentioned before, the aim of this thesis is to use synchronisation as this relaxation density.

2.5 Summary

This chapter described a brief review on the current data assimilation methods. It was exposed that the main methodologies rely on assumptions such as: the linearity of the observation operator, the Gaussianity of the prior pdf, and so the linearity of the model equations. In particular, we note that variational methods do not assume the observation operator to be linear and regarding the linearity of the model, EnKF and 4D-Var do not consider this assumption. Anyway, in the context of the complexity of the actual numerical models, each of these assumptions is becoming less and less valid.

It was explained that the variational methods provide the MAP model state and posterior uncertainty related to the error close to a local minimum. It is unclear, then, what model state is obtained from these methods when the linear assumptions become less valid and the posterior, multi-modal. Also, the Hessian is extremely expensive to compute, making it difficult for 4D-Var to obtain an error estimate.

Sequential methods like the EnKF do allow for nonlinear model equations, however the assumption of Gaussian analysis updates makes it also unclear what kind of result is obtained from this scheme and if the ensemble is indeed representing the true posterior.

Particle filters are, in principle, fully nonlinear and the above assumptions are not required in its formulation. Its main issue, however, is the filter degeneracy and several adaptations based on resampling have been studied in order to prevent the collapse of the particles. While the mere use of resampling has been proven to not be a solution for this issue, the use of proposal densities has proved to be a promising solution for high-dimensional systems. Several schemes have been explored. The Equivalent-Weights Particle Filter and the Implicit Equal-Weights Particle Filter proved to be non-degenerate schemes, in which the number of independent observations is not an obstacle for their efficiency.

In this thesis, the proposal density freedom will be explored with the above mentioned particle filters, using a linear observation operator and Gaussian errors in the model equations, although these methods could be used in a more general setting. A clear link between synchronisation and the particle filters will be investigated for the first time.

Chapter 3:

SYNCHRONISATION

”... It is quite worth noting that when we suspended two clocks so constructed from two hooks imbedded in the same wooden beam, the motions of each pendulum in opposite swings were so much in agreement that they never receded the least bit from each other and the sound of each was always heard simultaneously. Further, if this agreement was disturbed by some interference, it reestablished itself in a short time. For a long time I was amazed at this unexpected result, but after a careful examination finally found that the cause of this is due to the motion of the beam, even though this is hardly perceptible.”
(Memories of Christiaan Huygens in *Horologium Oscillatorium - The Pendulum Clock, or Geometrical Demonstrations Concerning the Motion of Pendula as Applied to Clocks)*

3.1 Introduction

The synchronisation phenomenon was first discovered and described in the 17th century by Christiaan Huygens, a Dutch researcher. He observed that two pendulum clocks suspended by a wooden beam on the top of two chairs would oscillate, after some time, in opposite directions, i.e. in an anti-phase regime, but with the same frequency. This intriguing fact was independent of their start positions. Additionally, he described that if an intervention was made to any of these pendula, this agreement between the phases and frequency would be restablished after a short period of time. The scientist has attributed this ”sympathy between the two clocks” to the support they were sharing - a flexible wooden bar - and its motion.

After this brilliant explanation for the phenomenon and during the subsequent centuries, other researchers have also experienced synchronisation in other systems, such as acoustical, radio communication and biological systems. Others have focused on deriving simplified models to explain the phenomenon. Very recently, Peña-Ramirez et al. (2016) presented a detailed study of Huygens’ experiment and a mathematical model following physical and mechanical laws, confirming that the occurrence of synchronisation was due to the dynamical behaviour of the wooden beam that was coupling the two clocks. Therefore, the main idea behind synchronisation is that, if two (or more) systems share or exchange information in a suitable way, they can synchronise to each other, i.e they can match some of their characteristics.

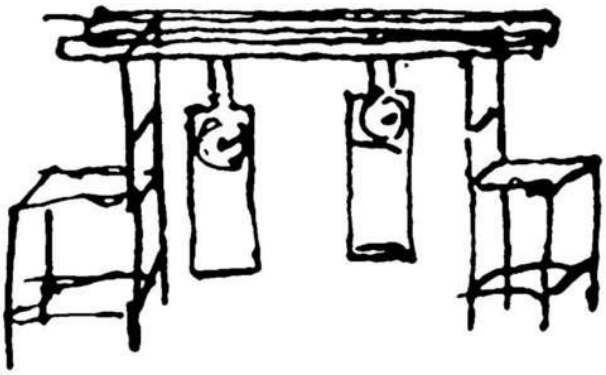


Figure 3.1: *A drawing by Christiaan Huygens of his experiment in 1665.*

3.2 Theory

Synchronisation is a dynamical process described by Pikovsky et al. (2001) as an adjustment of rhythms due to the interaction between two (or more) oscillating objects or systems. In the case of two clocks, as in the example described in the introduction, even a weak interaction between both can make them oscillate with a common frequency and a fixed phase difference. Synchronisation may then happen if this interaction's intensity or coupling strength is not zero, i.e. if the supporting frame to which the clocks are fixed is not rigid, allowing the clocks to share their vibrating movements with each other.

In this thesis, the main focus is obviously not to investigate the synchronisation phenomenon in clocks, but in nonlinear, chaotic systems. These systems are well-known to be very sensitive to small perturbations in initial conditions, which lead to a limited predictability of its dynamics. Oscillation in this kind of environment is viewed in the state space as more complex structures, called strange attractors, which can be defined as the path traveled by a point in the mathematical (state) space. This concept can be better understood when we come back to the first ideas of chaos theory, conceived by Edward Lorenz. He was a pioneer in this field with his work in Lorenz (1963). While studying convection rolls in the atmosphere, he produced an illustration of what was then called the ”Lorenz strange attractor” (figure 3.2). It is a 3-dimensional dynamical system which exhibits a chaotic behavior, despite its clear pattern of order. We note that the trajectories revolve around two main points in space and time. These are examples of strange attractors.

Instabilities along the strange attractor can be measured by a quantity called the Lyapunov exponent. Positive values of this quantity correspond to a growth of instability, negative values

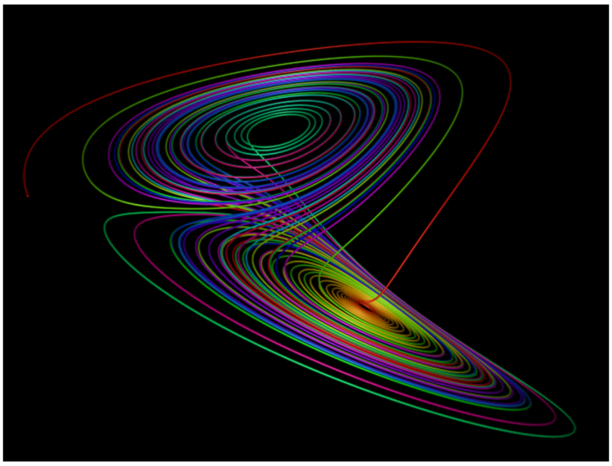


Figure 3.2: *An example of the Lorenz strange attractor, first reported in 1963.*

are related to a decay, i.e. a convergence of the trajectories towards the attractor, and zero Lyapunov exponents correspond to a transition between these behaviours. Regarding the distance δx between neighboring trajectories, one can compute its mean growth rate, which is called the leading finite-time Lyapunov exponent, $\lambda(t)$. It can be computed for a time t as:

$$\lambda(t) \simeq \frac{1}{t} ln \frac{\|\delta x(t)\|}{\|\delta x(0)\|} \tag{3.1}$$

Synchronisation theory investigates the stability of the synchronisation manifold, which is a set of subspaces to which states are attracted to, i.e. the D-dimensional linear subspace defined by $x_S = x_M$, where x_S and x_M denote states of the D-dimensional true system and the D-dimensional model, respectively. To achieve synchronisation, this manifold has to be transversally stable, such that

$$\lim_{t\to\infty} \|\mathbf{x}_S(t) - \mathbf{x}_M(t)\| = 0 \tag{3.2}$$

The stability of the synchronisation manifold is then defined as

$$\|\mathbf{x}_S(t) - \mathbf{x}_M(t)\| \leq \varepsilon \quad \forall t \tag{3.3}$$

in which ε is a threshold to be defined. This stability can be obtained with a suitable coupling term added to the model equations. The complete coincidence of the states, while they still perform chaotic dynamics in time, can happen if both x_S and x_M are identical chaotic systems. This is what is called complete synchronisation. The coupling strength plays an important role in this process: if it is too small, the exponential chaotic divergence will be only slowed down, as the attraction between the states will start to happen, whereas if it is strong enough, this attraction will lead to a decreasing difference between the states, reaching zero, after some time. It is interesting to note the relationship that exists between the coupling strength and the Lyapunov exponents, considering that the latter determine the largest growth rate of the exponential divergence. Actually, in order to reach complete synchronisation, the coupling term may exceed a threshold, somehow proportional to the Lyapunov exponents.

When the systems are not identical, however, we observe that the states do not coincide completely, i.e.

$$\lim_{t \to \infty} \|\mathbf{x}_S(t) - \mathbf{x}_M(t)\| \approx 0 \tag{3.4}$$

or x_S and x_M just get very close to each other. This phenomenon is then called generalised synchronisation (the event of complete synchronisation can be considered a specific case of the generalised one). Typically, the coupling involved in the generalised synchronisation is considered unidirectional, i.e. one system (master) influences the other (slave), without a response back. After the stabilisation of the synchronisation manifold, the slave system follows the master's behaviour. Considering that the geosciences field deals with nonidentical systems - usually numerical models and the true state of nature, which can only be approximated by observations - the generalised synchronisation is the phenomenon to be expected in such researches and so, the one to be studied in this thesis.

3.3 Different methodologies

The aim to stabilise a synchronisation manifold and synchronise a set of chaotic systems can be considered a type of chaos control problem, a field which is actively studied in many different areas, from engineering to biology. Many techniques have been developed during the past decades, aiming to control chaotic dynamical behaviour of nonlinear systems.

The first experiments on synchronisation inside a chaos perspective were based on the combination between two chaotic systems, in which one was defined as the transmitter and the other was the receiver. Each of them was composed of sub-systems with dynamical variables. The idea was to choose a correct variable to be sent from the transmitter to the receiver - which did not have all the same variables compared to the transmitter - and drive both systems towards synchronisation.

Researchers like Fujisaka and Yamada (1983) and Pecora and Carroll (1990) then started to focus on experiments using physical systems composed of a set of differential equations, aiming to reach synchronisation in a more realistic environment. To this end, systems like Lorenz, Rössler, Chua's circuit and some others are still widely used for synchronisation experiments. By considering configurations like the following Lorenz sub-systems:

$$\begin{aligned} &\dot{X} = \sigma(Y - X) & & \\ &\dot{Y} = \rho X - Y - XZ & &\dot{Y_1} = \rho X - Y_1 - XZ_1 \\ &\dot{Z} = -\beta Z + XY & &\dot{Z_1} = -\beta Z_1 + XY_1 \end{aligned} \tag{3.5}$$

one could synchronise the slave sub-systems Y_1 and Z_1 to the master sub-systems X, Y and Z, regardless their sensitivity to the initial conditions.

Fujisaka and Yamada (1983) were pioneers on publishing work on chaotic synchronisation with coupled-oscillator systems, by using a diffusive coupling constant D in a set of ordinary differential equations (ODEs) such as

$$\frac{dx_i}{dt} = f(x_i) + D\sum_{j=1}^{N}(x_j - x_i) \tag{3.6}$$

where $f(x_i)$ is a function which defines the chaotic systems and i and j are different spatial locations in the system. This scheme can be considered a simple type of nudging and worked by varying the values of the constant D, so that synchronisation can be obtained. Pikovsky (1984) has also contributed to the first investigations of synchronisation in chaotic attractors. But after Pecora and Carroll (1990) have shown the relationship between negative Lyapunov exponents in the sub-systems and synchronisation, they have brought attention to the synchronisation phenomenon and made clear that designing synchronising chaotic systems was possible.

As another important step in the field, Stojanovski et al. (1996) have pointed out that some issues still existed in the formulation proposed by Fujisaka and Yamada (1983), like e.g. the desynchronisation of the systems as soon as the coupling ceased and the dependence on the driver signals for synchronisation to happen. They also discuss in their work that a slave system behaves independently from a master, except in certain times t_n, when synchronisation between a slave component y_1 and a master component x_1 could happen. This way, the slave is driven by a sequence of samples or subsets $x_1(t_n)$. They showed, then, that synchronisation of the master-slave systems, i.e. $x(t) = y(t)$ was possible when using a finite number of samples.

Other work was focused on mutual couplings exchanged between equations, also acting as dissipative terms, making the systems synchronise, depending on the coupling strength. Mutual and unidirectional couplings have always been subject of study among scientists in the field. But Johnson et al. (1998) have shown that it was possible to optimize those linear couplings in a tunable way, in order to reach synchronisation. By considering two chaotic systems $x(t)$ and $y(t)$ where:

$$\begin{aligned} \frac{dx}{dt} &= f(x) \\ \frac{dy}{dt} &= f(y, x) \end{aligned} \tag{3.7}$$

the approach used was:

$$\begin{aligned} \frac{dx}{dt} &= f(x) \\ \frac{dy}{dt} &= f(y) - BK^T(y-x) \end{aligned} \tag{3.8}$$

in which vectors B and K were tunable and, if correctly set, were able to make the response (second equation in (3.8)) to follow the dynamics of the drive (first equation in (3.8)) and so reaching synchronisation. Further work on dynamical coupling was performed by Junge and Parlitz (2001), in which they used the contraction properties of the system and the aid of singular vectors to suppress the exponential divergence in state space and so ensuring a stable synchronisation regime. Some other authors have focused on finding coupling strategies to better understand the syncronisation phenomenon, like Guan et al. (2006).

Along the years, more sophisticated approaches have been developed, including regularisation techniques and also the concept of time embeddings, or time-delays. Such concept consists of including observations either from the past or from the future, in order to bring additional information to the present state. If the embedding space is large enough, it is possible to obtain the same characteristics of the original phase space trajectory in the reconstructed one (Letellier et al., 2005). A coupling scheme exploiting the inclusion of the delay embedding of the observations in the forcing term has been suggested by e.g. Abarbanel (2013), Rey et al. (2014a), Rey et al. (2014b) and Parlitz et al. (2014). These methods use the Jacobian of the forward model to move observation influence to the time instant of interest. A more simplified approach of a delay embedding nudging scheme has been developed by Pazó et al. (2016), where the computation of the Jacobian is avoided, although weaker synchronisation is achieved. The main idea behind these methods is that observations from other time steps can be used to increase the observability of a system that would otherwise not synchronise.

3.4 Synchronisation and data assimilation

A first discussion towards a clearer equivalence between the concepts of synchronisation and data assimilation was pointed out by Duane et al. (2006). The main goal of synchronisation is the convergence between the model and the truth in the future, allowing a predictable behaviour between chaotic systems. Generally speaking, data assimilation aims to *synchronise* the model evolution with the true evolution of a system, finding the best estimate of the state or its evolution, including its uncertainty. This best state will then produce the best prediction, assuming that the model is unbiased.

To reach their goals, both methodologies make use of a coupling term, which in general is unidirectional (a characteristic of the generalised synchronisation, as described previously in this chapter). In the data assimilation field, this coupling occurs from the truth to the model system. Additionally, it is incomplete, as observations (which are the real knowledge we have from the truth) are typically sparse and contain errors, likewise the model. Typically, as described previously, the model is extended with a nudging or relaxation term that forces it towards the observations. In synchronisation, this coupling has pretty much the same characteristics, but is specifically designed to control the spread of a probability distribution, centered on the synchronisation manifold $x_S = x_M$.

Duane et al. (2006), on a first attempt to find equivalences between synchronisation and data assimilation methods, have presented derivations in which they concluded that, in linear cases, synchronisation could be considered equivalent to 3D-Var or the Kalman filtering, depending on the background error used. For the nonlinear case, they have performed a hypothetical experiment with a three-variable system that switched, periodically, between the Lorenz set of equations and that system with the variables X and Z reversed:

$$\begin{aligned} \dot{X} &= \sigma(Y - X) & \dot{X} &= -\beta X + ZY \\ \dot{Y} &= \rho X - Y - XZ & \dot{Y} &= \rho Z - Y - ZX \\ \dot{Z} &= -\beta Z + XY & \dot{Z} &= \sigma(Y - Z) \end{aligned} \tag{3.9}$$

A comparison was made between a Kalman filtering approach and a synchronisation scheme. The latter used an alternate coupling in the slave sub-systems as follows:

$$\begin{aligned} \dot{X}_1 &= \sigma(Y_1 - X_1) + k(X - X_1) & \dot{X}_1 &= -\beta X_1 + Z_1 Y_1 \\ \dot{Y}_1 &= \rho X_1 - Y_1 - X_1 Z_1 + k(Y - Y_1) & \dot{Y}_1 &= \rho Z_1 - Y_1 - Z_1 X_1 + k(Y - Y_1) \\ \dot{Z}_1 &= -\beta Z_1 + X_1 Y_1 & \dot{Z}_1 &= \sigma(Y_1 - Z_1) + k(Z - Z_1) \end{aligned} \tag{3.10}$$

while the Kalman filter used observations of the X, Y and Z master system to drive the slave. For synchronisation to happen, the coupling parameter had a threshold value, i.e. synchronisation did not increase monotonically with k. Their results also showed that Kalman filtering presented bursts of desynchronisation during the transitions between Lorenz and the reversed system that were not happening in the coupling set used for synchronisation, although the latter presented greater errors on average. They suggested that the use of a time-dependent synchronisation coupling would lead to a reduction on the average error and so represent an optimal synchronisation scheme. They also argued that the Kalman filter method tended to be better than any other coupling scheme described in the synchronisation literature by that time. It is interesting to note though, that these methodologies, the Kalman filtering and synchronisation, have different purposes. The Kalman filter approach aims to compute an optimal analysis for a better departure of the model evolution, through a linear combination of an *a priori* state (forecast) and a weighted difference between observations and the predicted observations. Synchronisation tries to find an optimal forecast, a model trajectory that follows the observations as closely as possible, aiming for the fastest convergence of the model with the true future trajectory. This is done by combining the forecast with a weighted difference between a set of observations and a set of forecasts.

Still on the equivalences between synchronisation and data assimilation methods, Yang et al. (2006) have also tested synchronisation using a three-variable Lorenz system and a similar type of nudging as described in (3.10) for the slave system, finding what they called an optimal coupling strength, which could generate comparable results to the ones produced by 3D-Var. That was considered indeed a remarkable result, considering the simplicity of the nudging scheme used, in comparison with the 3D-Var framework. They also tested synchronisation with bred vectors and singular vectors, suggesting that learning the relationship between the coupling parameter and the bred vector growth rate could help to reduce the error growth. Their overall conclusion at that time was that the development of a hybrid synchronisation-data assimilation methodology in the future would improve weather forecasts. Interestingly, that is exactly the aim of this thesis.

The use of time embeddings or time delays has also allowed significant advances in the synchronisation field. It consists of incorporating in the synchronisation coupling dynamics the existing information from different time delays. Note that here the term ”delay” can refer to information contained in the past and/or in the future. This concept comes from Taken’s delay embedding theorem (Kantz and Schreiber, 1997), which explains that, in the absence of enough observations, the delayed information can be used to reconstruct the topology of the attractor. Work from Abarbanel et al. (1994); Zhan et al. (2003); Dhamala et al. (2004); Feng et al. (2007); Chen and Kurths (2007); Huang et al. (2011); Ghosh et al. (2012); Pazó et al. (2016) and other researchers related to different fields, from engineering to neural dynamics, focused on reaching synchronisation and/or predicting physical variables by using the concept of time-delay embeddings. The main structure of the time embeddings shares similarities with the one found in a strong constraint 4D-Var, as the main idea of these systems is to use a sliding observation window, bringing back information from time-advanced observations. Some crucial differences between these methods are noted, though. In synchronisation, the analysis is propagated in small time increments dt, while in 4D-Var this is done throughout the observation window. Additionally, in synchronisation the observations are re-used as we slide the window (note that not all observations inside this window are used, only specific observations are chosen at once). Another noticeable difference is that these time embedding synchronisation frameworks usually utilize a truncated singular value decomposition, while 4D-Var uses a background term to perform regularization.

Among these time embedding formulations, Rey et al. (2014a,b) have achieved remarkable synchronisation results with the coupling term designed in their framework. They performed tests considering a twin experiment using the Lorenz96 chaotic model (Lorenz, 1995), with 20 state variables to be estimated and only 1 observed state. Among their results, they were able to find synchronisation errors with magnitudes of the order of 10^{-15}, i.e. close to machine precision. Furthermore, the authors achieved successful prediction ranges of around 2,000 time steps after an estimation period of 10,000 time steps. It should be mentioned, however, that these results assumed that the observations contained no errors, which is unrealistic in the data assimilation context. In their setting, even small observational errors can have a large impact on the synchronisation results. Furthermore, their method is far too expensive to be used in realistic high-dimensional geophysical problems as it needs the propagation of the Jacobian matrix of the forward model, with size the model dimension squared. Despite these issues, their methodology has inspired the research in this thesis, given the powerful convergence between the model and the truth obtained. This framework will be described in more details next.

3.5 Synchronisation framework

3.5.1 Overview

The first part of this thesis is inspired by the synchronisation ideas of Rey et al. (2014a,b). The methodology used by them explores the concept of time-advanced embeddings, aiming to bring additional information from observations in the future back to present time. The main goal of this methodology is to stabilise the synchronisation manifold and, after reaching synchronisation, produce very precise estimates of all variables, including the unobserved ones. Following these estimates, accurate predictions are obtained, over a significant forecast period.

The synchronisation framework proposed by Rey et al. (2014a,b) can be summarised as follows: consider the state $x \in \Re^{D_x}$ and observations $y \in \Re^{D_y}$ at each time step of the model:

$$x^{j+1} = f(x^j) \tag{3.11}$$

where $f(x^j)$ is the nonlinear model at time j.

The framework is constructed by following the next steps:

(i) Define D_d as the delay dimension, containing the time embeddings to include information from D_y observations at different times in a time interval $[j, j + (D_d - 1)\tau]$, where τ is a constant time interval. Similarities with a fixed-lag smoother can be noted. A remark to be made is that the constant τ has an important role in this formulation, related to the minimum observation frequency needed to synchronise the system. This interesting feature will be explained further along this thesis.

(ii) Define an embedding dimension $D_e = D_d * D_y$ and construct vectors $S \in \Re^{D_e}$ and

$Y \in \Re^{D_e}$, related to the states and the observations, respectively as:

$$S(x^j) = ((H(x^j)^T, (H(x^{j+\tau}))^T, ..., (H(x^{j+(D_d-1)\tau}))^T)^T \tag{3.12}$$

and

$$Y^j = ((y^j)^T, (y^{j+\tau})^T, ..., (y^{j+(D_d-1)\tau})^T)^T \tag{3.13}$$

in which $S(x)$ is a map from physical to the embedding space and $H(x)$ is the observation operator at each observation time, a map $\Re^{D_x} \rightarrow \Re^{D_y}$. This operator can be linear or nonlinear. For simplicity, it is assumed that H is the same at each time instant.

(iii) Compute the Jacobian matrix $\frac{\partial S(x^j)}{\partial x^j}$. To this end, we must evaluate

$$\frac{\partial H(x^{j+\tau})}{\partial x^j} = \mathbf{H}\boldsymbol{F}(x)_{j\rightarrow j+\tau} \tag{3.14}$$

in which $\mathbf{H}$ is the Jacobian of the map H and $\boldsymbol{F}(x)_{j\rightarrow j+\tau}$ is the linearised model, i.e. the Jacobian of the nonlinear model, from time j to $j+\tau$. The Jacobian matrix $\frac{\partial S(x^j)}{\partial x^j}$ is of size $D_e \times D_x$, where D_x is the number of dimensions of the system:

$$\frac{\partial S(x^j)}{\partial x^j} = \begin{pmatrix} \mathbf{H}(x) \\ \mathbf{H}\boldsymbol{F}(x)_{j\rightarrow j+\tau} \\ \vdots \\ \mathbf{H}\boldsymbol{F}(x)_{j\rightarrow j+(D_d-1)\tau} \end{pmatrix}$$

(iv) Compute the pseudoinverse $\frac{\partial S(x^j)}{\partial x^j}^{\dagger}$ of the Jacobian matrix, using a truncated singular value decomposition (SVD) (the singular-value decomposition of a $D_e \times D_x$ matrix is a factorization of the form $U\Sigma V^T$, where U is a $D_e \times D_e$ unitary matrix, Σ is a $D_e \times D_x$ diagonal matrix with non-negative values on the diagonal, and V is a $D_x \times D_x$ unitary matrix. The pseudoinverse of the Jacobian matrix using this factorised form is then: $V\Sigma^{-1}U^T$).

An issue that appears in this methodology is that the computation of this pseudoinverse can be prohibitively expensive for high-dimensional systems, as $\frac{\partial S(x^j)}{\partial x^j}$ is a $D_e \times D_x$ matrix, and so depending on the size of the system.

(v) Finally, compute the variable evolution with time, by using the following equation:

$$x^{j+1} = f(x^j) + g\frac{\partial S(x^j)}{\partial x^j}^{\dagger}(Y^j - S(x^j)) \tag{3.15}$$

where g is a coupling constant, a useful tuning parameter that will be explained further in this thesis.

(vi) Keep iterating the whole process (steps (i) to (v)) from time $j+1$ on, until the last observation.

The pseudoinverse $\frac{\partial S(x^j)}{\partial x^j}^{\dagger}$ is responsible for spreading the information from the observed variables to all unobserved variables of the model, at time j. Note that not all observations which are available within the time interval $[j, j+(D_d-1)\tau]$ are used, but only those τ time steps apart are selected. The main idea is that these specific observations are sufficient to bring valuable information to the system and those in between do not carry much additional information.

An important point to be highlighted is that $S(x)$ must be recalculated at every time step. This means that we need to linearise the model at every time step over the time interval $[j, j+(D_d-1)\tau]$, i.e if, for example, the dimension of our system is $D_x = 1000$ and a delay dimension $D_d = 5$ with a constant time interval $\tau = 10\Delta t$ is used, a propagation of a 1000×1000 matrix for 40 time steps at each time step will be needed. This is a weakness of this framework, if designed to high-dimensional systems, as this procedure would be extremely expensive for large models and that is the main motivation to turn to an ensemble framework. That is, indeed, one of the goals of this thesis and a description of an ensemble-based idea to make this framework more suitable to high-dimensional systems will be presented in the next chapter.

The appendix A contains a pseudocode of the synchronisation scheme as described here.

3.5.2 A connection between synchronisation and the Kalman smoother

At this point the reader must be wondering how close this synchronisation framework is to the well-known data assimilation methodologies. A first connection between the two methods becomes apparent when the dynamics of the system is assumed to be linear. In this case, similarities can be found between synchronisation and the Kalman Filter, as it was also discussed in Duane (2015). The main difference is that synchronisation uses a tunable parameter to set the strength of the relaxation term, without taking into account any observation or model error covariance matrices, while the Kalman Filter includes a Kalman gain in its formulation, which is determined completely by the error covariances of the model and the observations. This way, the Kalman Filter provides an optimal gain, with minimal unbiased posterior errors. Still in a linear system, but considering the time embeddings as in the present framework, the same equivalence is valid, but now the optimal gain comes from the Kalman Smoother.

Another point to be mentioned is that in synchronisation, observations are re-used in the time embedding framework, when the observation window is moved as a slidding fixed-lag smoother. Including the re-use of observations in a Kalman Smoother scheme would be possible, however its formulation would become rather complex, as the correlations between model and observation errors would have to be taken into account. Synchronisation does not include these correlations in its formulation, though.

For a mathematical comparison between the synchronisation framework presented and the Kalman Smoother, we can start by reducing the synchronisation formulation to a non-lagged version. By not considering the time embedding extension, equation 3.15 reduces to:

$$x^{j+1} = f(x^j) + gH^\dagger(y^j - H(x^j)) \tag{3.16}$$

where $H^\dagger$ is a generalized inverse of H (considering that the direct inverse of H does not exist, as the state vector is partially observed in geophysical systems). This idea can then be extended to a lagged version, where future observations are also taken into account:

$$x^{j+1} = f(x^j) + g\tilde{H}^\dagger(Y^j - \tilde{H}(x^j)) \tag{3.17}$$

where

$$\tilde{H} = (H^j F^j, H^{j+1} F^{j:j+1}, ..., H^k F^{j:k})^T \tag{3.18}$$

The matrix $\tilde{H}$ is a $(k - j + 1) * D_y \times D_x$ matrix, also called $\frac{\partial S(x^j)}{\partial x^j}$ in the previous section.

We then remember equation 2.30 for the Kalman Smoother and rewrite it as:

$$\hat{x}^{j+1} = f(\hat{x}^j) + P\hat{H}^T \left(\hat{H}P\hat{H}^T + R\right)^{-1} (Y^j - \hat{H}\hat{x}^j) \tag{3.19}$$

We note that it is possible to write the product $\hat{H}P\hat{H}^T$ as follows:

$$\hat{H}P\hat{H}^T = \tilde{H}P^{j,j}\tilde{H}^T \tag{3.20}$$

This way, we have moved the propagation of the model from $P^{j,j}$ to the observation operator. Furthermore, we can write $\hat{H}\hat{x}^j = \tilde{H}x^j$, as $\hat{H}$ is a large matrix with $H^j, H^{j+1}, ..., H^k$ as blocks on the diagonal. Since we are only interested in the new state at time j, we can write:

$$x^{j+1} = f(x^j) + P^{j,j}\tilde{H}^T \left(\tilde{H}P^{j,j}\tilde{H}^T + R\right)^{-1} (Y^j - \tilde{H}x^j) \tag{3.21}$$

which allows us a direct comparison with the time-embedded synchronisation formulation. In that case, we implicitly assume that $R = 0$ and $P^{j,j} = I$, leading to:

$$x^{j+1} = f(x^j) + \tilde{H}^T \left(\tilde{H}\tilde{H}^T\right)^{-1} (Y^j - \tilde{H}x^j) \tag{3.22}$$

Indeed, this is one form of the pseudo-inverse when $\tilde{H}$ is broader than it is tall. This corresponds to the case that the total number of embedded observations is larger than the dimension of the state vector.

If we apply the Woodbury identity to equation 3.21, we find:

$$x^{j+1} = f(x^j) + \left((P^{j,j})^{-1} + \tilde{H}^T R^{-1} \tilde{H}\right)^{-1} \tilde{H}^T R^{-1} (Y^j - \tilde{H} x^j) \tag{3.23}$$

This equation has the advantage over the covariance form of the Kalman Smoother in (3.21) that we now have to multiply the propagation matrix with the block-diagonal matrix R, so the propagation of $P^{j,j}$ is not performed. If we assume that $R = \sigma^2 I$ and multiply the numerator and denominator with the observation noise squared σ^2, we obtain:

$$x^{j+1} = f(x^j) + \left(\sigma^2 (P^{j,j})^{-1} + \tilde{H}^T \tilde{H}\right)^{-1} \tilde{H}^T (Y^j - \tilde{H} x^j) \tag{3.24}$$

If we then take the limit of small observation noise:

$$x^{j+1} = f(x^j) + \left(\tilde{H}^T \tilde{H}\right)^{-1} \tilde{H}^T (Y^j - \tilde{H} x^j) \tag{3.25}$$

which corresponds to the pseudoinverse when $\tilde{H}$ is taller than broader. These Kalman Smoother equations show us how observation errors can be included into the synchronisation framework, and the innovation form of the Kalman Smoother (equation 3.24) shows that the observation noise acts as a regularisation term for the pseudoinverse.

3.5.3 Experiment configuration and results

3.5.3.1 Lorenz96 model

All experiments performed in chapters 3, 4 and 5 used a D_x-dimensional Lorenz96 model, a periodic dynamical system proposed by Edward Lorenz (Lorenz, 1995) to investigate the predictability of chaotic systems like the atmosphere. It is governed by the following formulation:

$$\frac{dx_a}{dt} = (x_{a+1} - x_{a-2}) x_{a-1} - x_a + f \tag{3.26}$$

where $a = 1, ..., D_x$ are variables corresponding to equidistant grid points along a circle. Periodic ”boundary conditions” are defined, as: $x_{-1} = x_{a-1}$, $x_0 = x_a$ and $x_{a+1} = x_1$. The constant f is a forcing parameter and different solutions can be obtained, depending on the value chosen for it. In all experiments shown here a value of $f = 8.17$ was used, which is in the chaotic regime.

The linear term $-x_a$ in the equation can be considered equivalent to a damping term in the atmosphere, so one can use this fact to interpret the time units in the model, which are dimensionless. If we consider a synoptic atmospheric dissipation time scale as 5 days, then this time would be the equivalent to 1 time unit in the model. This way, a $6h$ period of time in a real atmosphere would be equivalent to 0.05 dimensionless time units in Lorenz96 model.

3.5.3.2 Configuration

The metric typically used in synchronisation to monitor the synchronisation error (SE) at every time t is the root mean square error (RMSE), defined by:

$$SE(t) = \sqrt{\frac{1}{D_x}\sum_{k=1}^{D_x}(x^k_{true}(t) - x^k(t))^2} \tag{3.27}$$

Note that all RMSE values shown in this work are global synchronisation errors, i.e. they are computed considering both observed and unobserved variables.

In this thesis twin experiments are performed, i.e. the truth is artificially generated by the numerical model, which allows 'x_{true}' to be used in eq. (3.27). In real cases, however, our knowledge about the truth relies on the observations, which means that x_{true} would be substituted by y and x^k would be replaced by $H(x)$ in the computation of this evaluation metric.

Note that tests with different initial conditions and random number realisations were performed in all the experiments, all of them leading to qualitative and quantitative results similar to the ones presented in this work. Therefore, these results can be considered reproducible and typical for the behaviour of the system at hand.

Tests were performed for different system sizes: $D_x =$ 20, 100 and 1000, in order to follow the sensitivity of the scheme to an increase in the system dimension. In all the experiments performed in this chapter we have observations covering only 25% of the system, all equally distributed along the Lorenz ring. 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 if compared with the number of its unstable directions is a challenging problem.

A constant observation time interval $\tau = 10\Delta t$ is adopted and only observations that appear at every τ time steps inside the embedding window are used, i.e. not all of the observations available are included in equation (3.15). Also, an observation noise sampled from a normal distribution with standard deviation $\sigma_{obs} = 0.1$ is used. Note that a typical size for the state variables in Lorenz96 model is between -10 and 10.

A time step $\Delta t = 0.01$ is used and a 4th-order Runge-Kutta scheme (also known as RK4) is applied. The RK4 is a method used for numerical integrations of ordinary differential equations. The following formulation was used: considering equation (3.11) and starting from an initial condition $x_{old} = x(j_0)$ and a time increment h:

$$\begin{aligned} k1 &= f(x_{old}, j_0) \\ k2 &= f(x_{old} + (k1)\frac{h}{2}, j_0 + \frac{h}{2}) \\ k3 &= f(x_{old} + (k2)\frac{h}{2}, j_0 + \frac{h}{2}) \\ k4 &= f(x_{old} + (k3)h, j_0 + h) \end{aligned} \tag{3.28}$$

where the k are the slope estimates. In order to find $x(j_0 + h)$, an approximation of the weighted average of these slopes is computed:

$$x(j_0 + h) = x_{old} + \frac{k1 + 2(k2) + 2(k3) + k4}{6} h \tag{3.29}$$

Tests using the coupling constant g which appears in equation (3.15) equal to 1 lead to synchronised results. However, for values of $g = 0.1$, corresponding to a reduction of the influence of the whole coupling term in equation (3.15) by a factor of 10, produces slightly better results. We note that a maximum value for g does not yield a better synchronisation, since by increasing this coupling factor we also increase the amount of noise that is introduced to the system (Yang et al., 2006). Actually, these authors argue that this noise may also help to determine the optimal value of g and that this optimal coupling strength is roughly the minimum value used to obtain synchronisation in a noiseless system. Therefore, in all synchronisation experiments shown in this chapter the value of $g = 0.1$ is used.

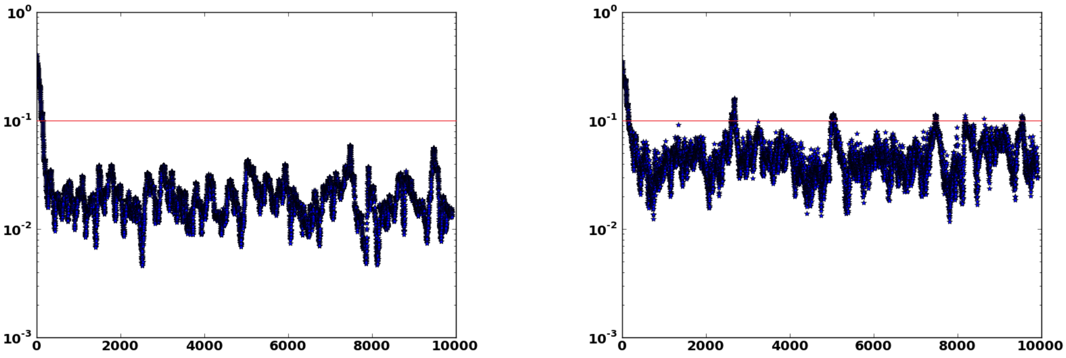
Additionally, after exhaustive tests with the computation of the SVD for the pseudoinverse of the Jacobian matrix, it was observed that the number of singular values to be used in order to produce the most stable results was equal to $2 * D_y$. Hence, this value was chosen for all the experiments shown in this subsection.

All experiments in chapters 3 and 4 were performed on a compute cluster machine and coded in Python language.

3.5.3.3 20-variable case

In this case, 5 variables were observed equidistantly on the Lorenz96 ring and the total length of the experiment was of 10,000 time steps (comparable with Rey et al. (2014a,b)).

A first comparison is made between results for the synchronisation framework using a time embedding $D_d = 5$ and the Kalman Smoother scheme presented in section 3.5.2. The observation noise introduced is $\sigma_{obs} = 0.1$. Figure 3.3 shows global RMSEs for synchronisation and the KS methods. Table 3.1 describes the error mean and the error variance for these 2 methods. Results for this KS implementation were slightly inferior if compared to synchronisation results. In terms of computation, some rank issues were found in the KS calculations. Different initial conditions were tested for this comparison, showing the same results: synchronisation performs better than this version of the KS. Duane et al. (2006) have found opposite results while comparing their simpler synchronisation framework with a Kalman Filter for the Lorenz63 model. The initial results shown in figure 3.3 can suggest a possible advantage of the use of time embeddings in the synchronisation formulation to improve its performance over the Kalman Smoother formulation.



(a)

(b)

Figure 3.3: *Global RMSEs for 10,000 time steps for: (a) synchronisation and (b) KS. The horizontal red lines represent the standard deviation of the observation noise.*

	SYNCHRONISATION	KS
Error MEAN	0.0196	0.0467
Error VARIANCE	6.44e-05	3.51e-04

Table 3.1: Error mean and error variance for synchronisation and the KS.

After this first experiment on the comparison between the KS and synchronisation, all the next results shown in this chapter will be focused on the synchronisation framework presented in equation (3.15).

Firstly, it is important to note that the size of the delay dimension D_d is a crucial factor in this framework, as it needs to be large enough to provide useful and additional information to compensate the lack of observations, but small enough to keep the numerical stability of the pseudoinverse calculation. Figure 3.4 shows the synchronisation errors (RMSE) for different embedding intervals. In Rey et al. (2014a) the authors have reached RMSEs of around 10^{-15}, but in that work they did not include observational noise to the system. The magnitude of the RMSEs found here is comparable to the results in Rey et al. (2014b), although those authors have performed experiments using observation noise sampled from a uniform distribution, for standard deviations up to $\sigma_{obs} = 0.01$ only. They found that the fluctuations of the RMSEs were roughly of the same order of magnitude of the noise introduced.

The system can be defined as synchronised when $RMSE < \sigma_{obs}$ and stabilised when no sharp peaks in the synchronisation error are found along the run. Figure 3.4 illustrates the

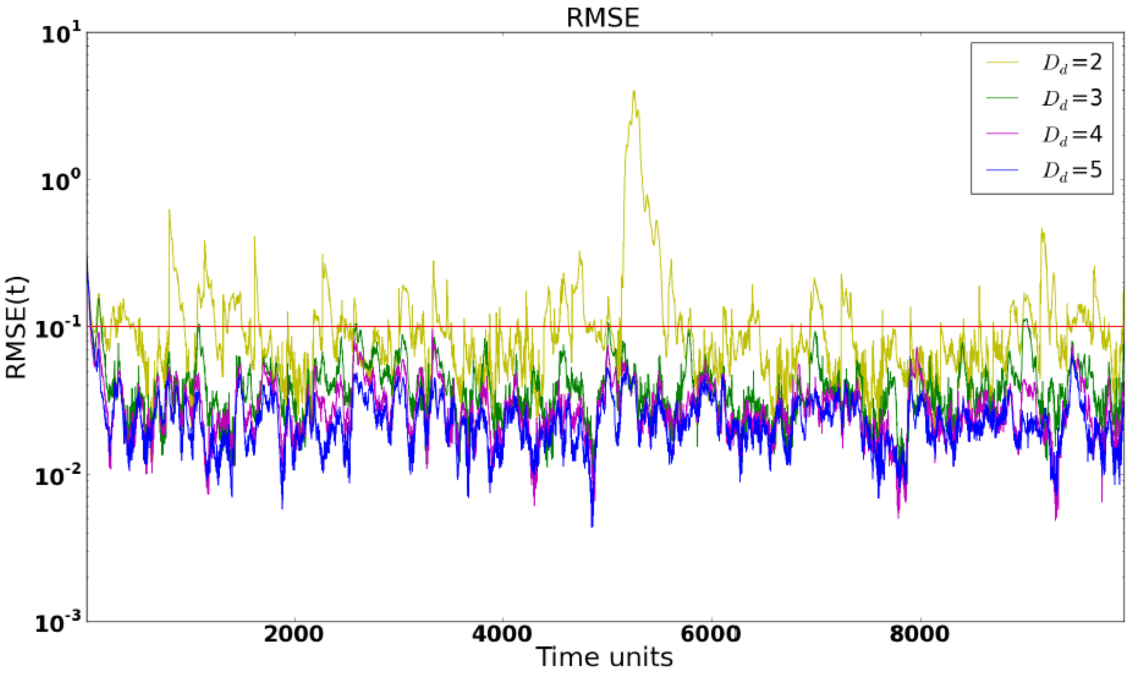


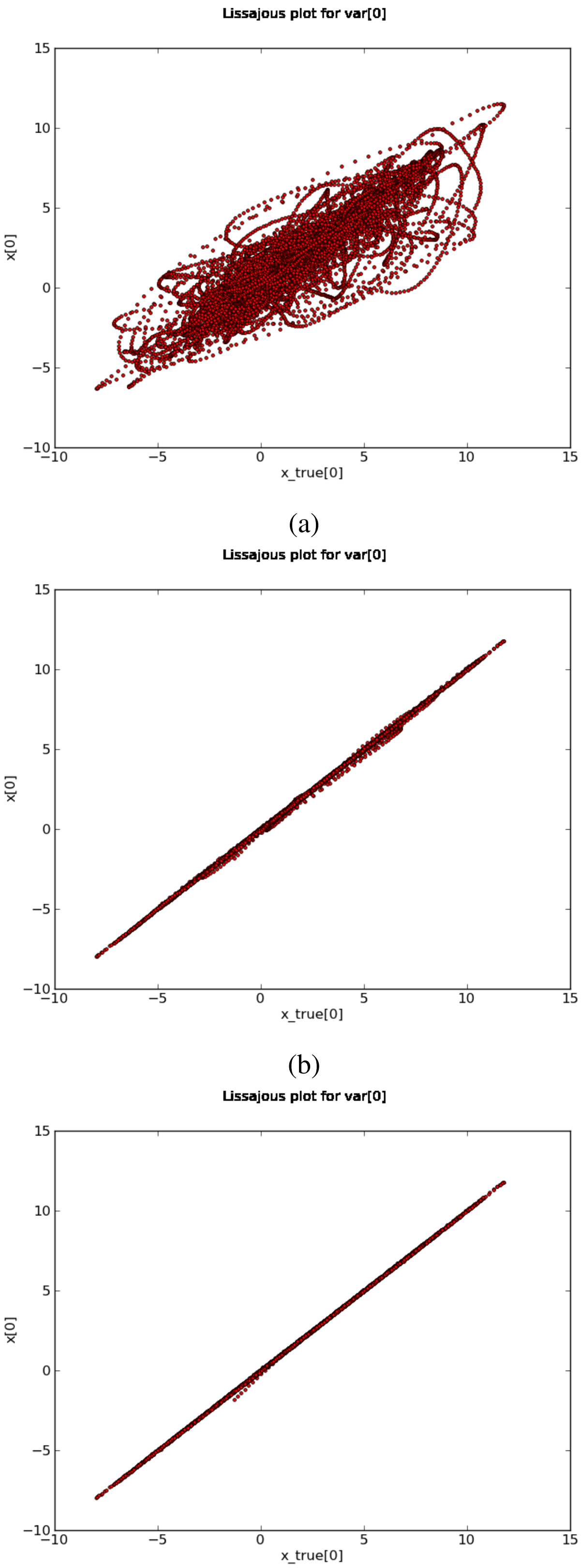
Figure 3.4: *Global Synchronisation Error (RMSE) for different delay dimensions in a 20-variable system with 5 measured variables sampled equidistantly on the Lorenz96 ring. The horizontal red line represents the standard deviation of the observation noise.*

following: 1) a delay dimension of at least $D_d = 3$ is needed to reach RMSE values $< \sigma_{obs}$ and so achieve synchronisation; 2) for $D_d = 2$, the system does not stabilize and shows moments of desynchronisation; 3) for $D_d = 4$ and $D_d = 5$ low RMSE values are obtained and convergence seems to have been reached.

Figure 3.5 is a Lissajous-type plot, which describes the behaviour of a variable during a period of time in a plane containing x and x_{true}. The points far from the diagonal in figure 3.5 (a) represent desynchronisation, showing that in the case of $D_d = 1$, i.e. in the absence of time embeddings, the system does not synchronise. By increasing the delay dimension, the values for x and x_{true} get closer to each other, depicting a diagonal line. In other words, by increasing the delay dimension, a stabilised synchronisation manifold is reached (figure 3.5 (c)).

Figure 3.6 shows the leading Lyapunov exponents for 10,000 time steps for different delay dimensions. The red crosses show positive values, providing information on which instances a perturbation will potentially increase in an exponential way in finite time. For instance, the red sequence found between around $t = 5000$ and $t = 5500$ in figure 3.6 (a) clearly denotes the pronounced peak shown in figure 3.4 for $D_d = 2$ at the same period of time. A delay dimension $D_d = 5$ (figure 3.6 (b)) avoids any growth of instabilities, converging trajectories towards the attractor.

The effect of observation noise in this type of system is also tested. Similar experiments in lower-dimensional systems were performed by Rey et al. (2014b), showing that, as noise levels increase, estimate's accuracy degrades significantly. In order to understand the impact of the

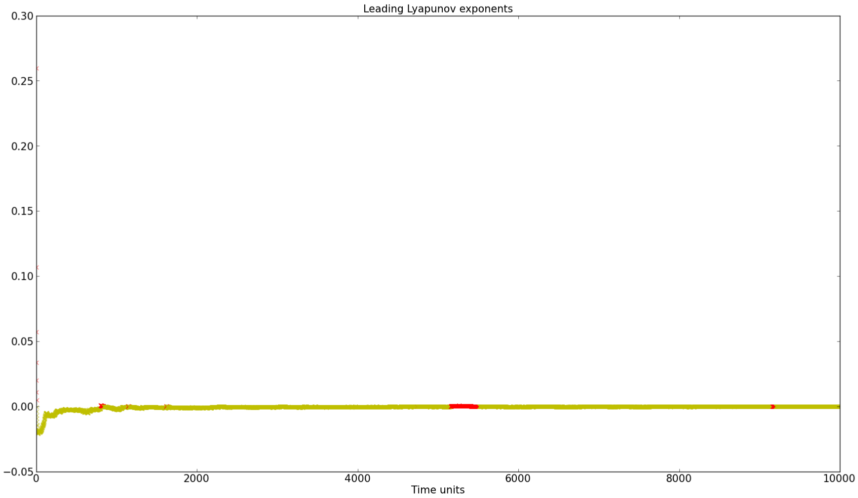


(a)

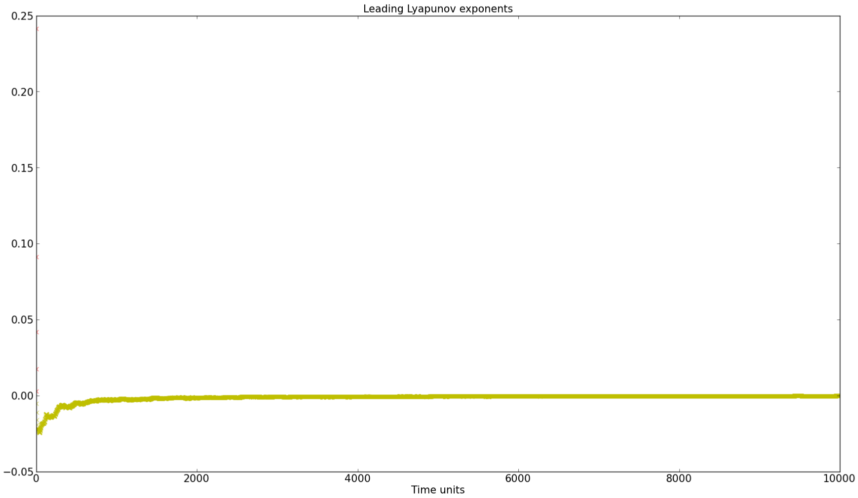
(b)

(c)

Figure 3.5: *Lissajous-type plots for variable $[0]$ during 10,000 time steps, using different delay dimensions: (a) $D_d = 1$, (b) $D_d = 2$ and (c) $D_d = 5$.*



(a)



(b)

Figure 3.6: *Leading lyapunov exponents along 10,000 time steps for $D_x = 20$: (top) $D_d = 2$ and (bottom) $D_d = 5$. The red crosses are positive lyapunov exponents and the yellow ones are negative lyapunov exponents.*

observation noise in the system used in this work, figure 3.7 shows results for σ_{obs} between 0.1 and 0.001. It is clear that the measurement noise affects the quality of synchronisation, particularly in the first 500 time steps, although synchronisation has been reached for all cases. The same realisations for the observation noise were used in all experiments. A general conclusion from these results is that $RMSE \approx (1/5)\sigma_{obs}$.

After 9,960 time steps (10,000 time steps minus $(D_d - 1)\tau$, for $D_d = 5$), the system is set to run freely, i.e no synchronisation scheme is used. Figure 3.8 shows the estimations and predictions for trajectories of the first 10 variables. The blue curves are the truth and the green ones are the estimates/predictions. The estimates are so close to the truth that both curves practically lie on

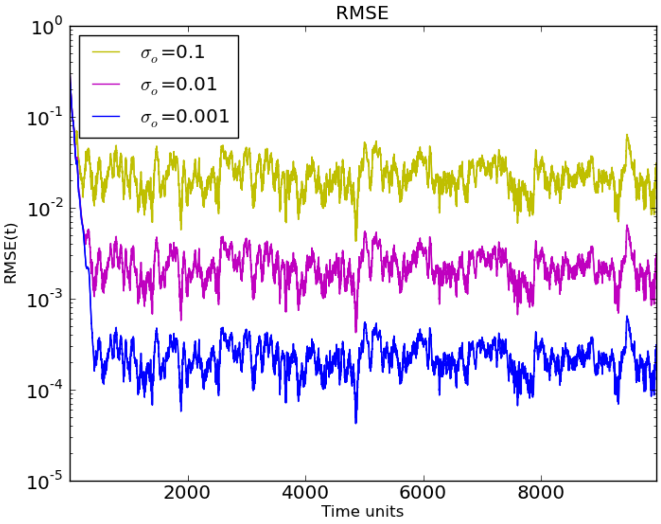


Figure 3.7: *Global Synchronisation Error (RMSE) for different standard deviations (σ) of observation noise in a 20-variable system with 5 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 5$).*

top of each other. Figure 3.9 shows a zoom-in at the end of the estimation period for one of the variables to highlight the differences between the truth, the estimate and the observations. Note that these experiments use noisy observations, and the estimates fit the truth, not the observations, as can be seen in figure 3.9. This shows that synchronisation tries to find a model trajectory that follows the observations closely, in order to synchronise with the true trajectory. Also in figure 3.8, during the prediction stage (after the red line), precise values are obtained for almost 500 time steps and after that, predictions start to diverge from the truth, given the chaotic nature of the model.

Regarding the prediction range, it depends on how well synchronised the system is at the starting point of the prediction. Figure 3.10 shows that if, for instance, we start prediction at time step 4,855, where we find minimal RMSE values, it is possible to predict an unobserved variable precisely for around 400 time steps. If, however, we start predicting at time step 9,454, when the RMSE value points out a less synchronised moment of the system, the prediction range for the variable is significantly reduced, reaching around 150 time steps.

Possible reasons that could explain a reduced prediction range in the second case are:

(1) poor initial condition, due to inferior synchronisation, as mentioned before, i.e. starting far away from the truth and

(2) the fact that, even if a tiny amount of observation noise is included, states will (at the end of the synchronisation period) end up in a different part on the chaotic attractor, where local divergence of neighbouring trajectories might be larger. This is related to the specific sensitivity of the system and its location in the attractor and it depends on the specific realization. To exclude this effect, one should average over many simulations with different

initial conditions and/or many prediction attempts, at different times.

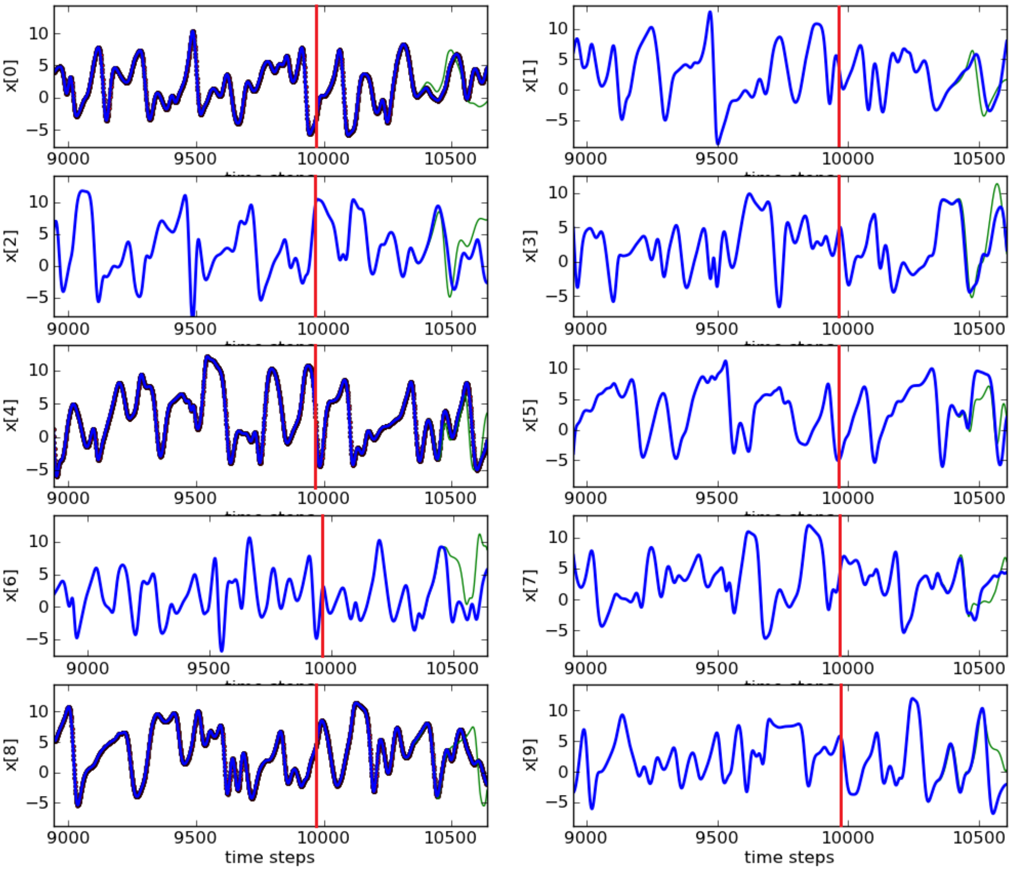


Figure 3.8: *Trajectories of the first 10 variables (y axis) and the* ***estimates*** *(until 9,960 time steps - red lines)* ***/predictions*** *(after 9,960 time steps). The blue lines are the truth and the green ones are the estimates/predictions. The observed variables have small red dots in their trajectories (variables 0, 4 and 8).*

It is interesting to note that it was not possible to reach synchronisation with only one observed variable, like in Rey et al. (2014a), due to the inclusion of observation errors in this system, something that was not considered in that work. Even increasing D_d to up to 20 was not useful to synchronise. The conclusion is that there is not enough information in 1 noisy observation to synchronise the system. Nevertheless, this situation changes dramatically if the number of observed variables is increased to 5, for example, which means that 25% of the system is observed. Also, as soon as synchronisation is then reached with this set of observations, it is possible to reduce the time embedding window to 40 (i.e. $D_d = 5$), if compared to the larger embedding interval needed in Rey et al. (2014a), i.e. $D_d = 8$. This important point will be discussed further in the next chapter.

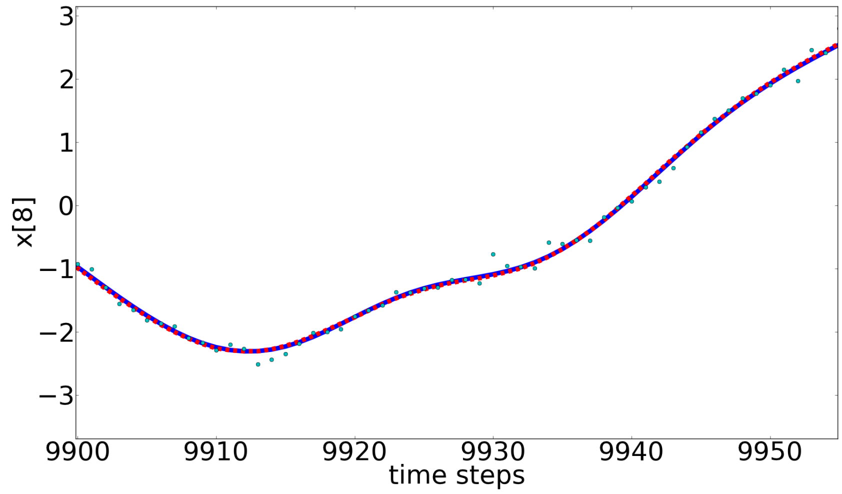


Figure 3.9: *Trajectory of an observed variable in figure 3.8 at the end of the estimation period. The blue line is the truth, the red dashed line is the estimate and the green dots are the observations. Note the near perfect synchronisation.*

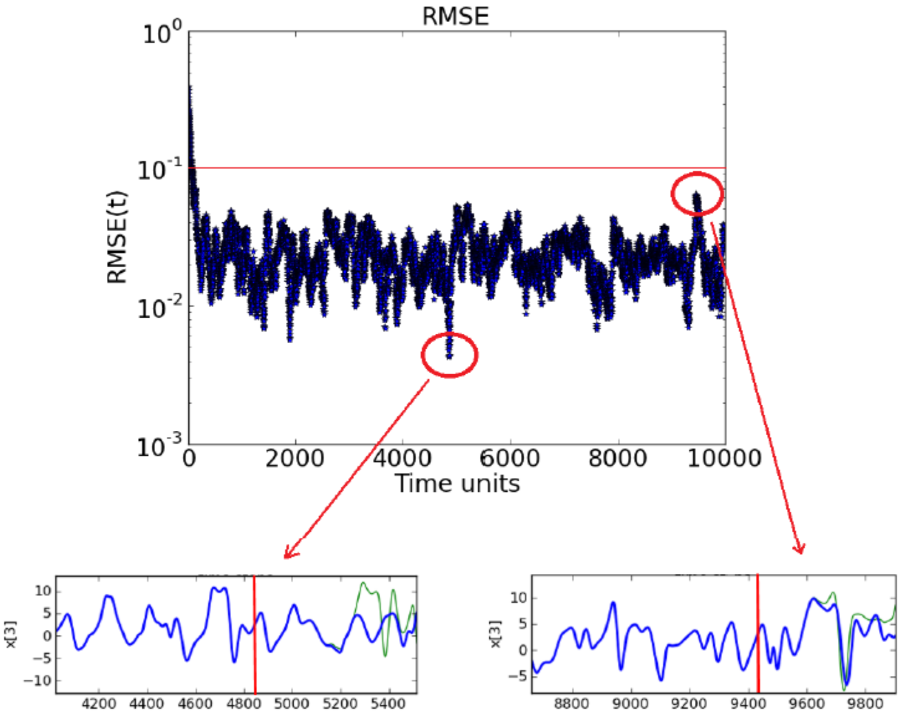


Figure 3.10: *Global Synchronisation Error (RMSE) for a 20-variable system with 5 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 5$) (top). The arrows point to unobserved variables estimates/predictions when the estimation period ends at (left) 4,855 time steps and (right) 9,454 time steps. The prediction range in the first case is approximately 400 time steps, while in the second case it is approximately 150 time steps. The horizontal red line represents the standard deviation of the observation noise.*

3.5.3.4 100-variable case

In this case, 25 variables were observed at every $\tau = 10$ time steps. The same observation noise previously used is introduced into the system. As explained in the previous case, after the initial transient period towards stabilisation, the length of the estimation period is not the crucial point in finding synchronisation, so in this and the next cases the length of the experiments are reduced to 1,000 time steps, without any loss of information.

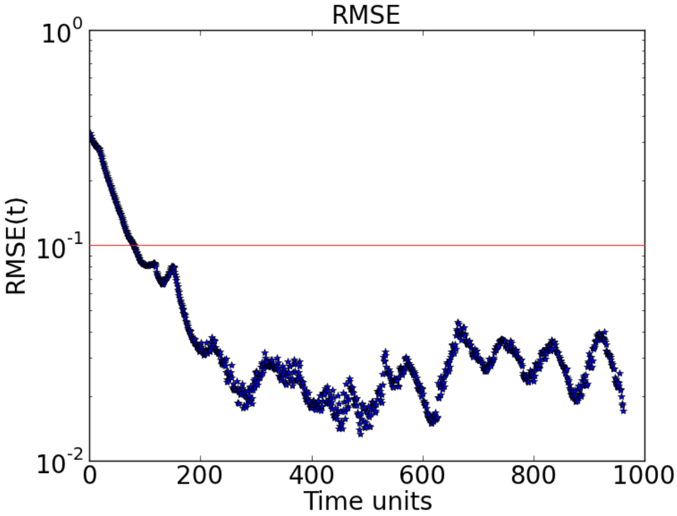


Figure 3.11: *Global Synchronisation Error (RMSE) for a 100-variable system with 25 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 5$). The horizontal red line represents the standard deviation of the observation noise.*

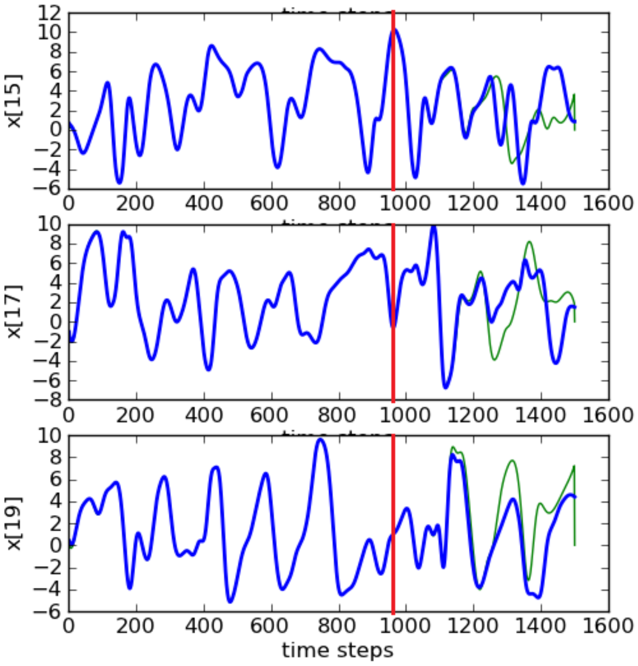


Figure 3.12: *Trajectories of 3 unobserved variables in a 100-dimensional system. The blue lines are the truth and the green ones are the estimates. Predictions start at time step 960 (red lines).*

Keeping the previous optimal value for the delay dimension $D_d = 5$, figure 3.11 shows the RMSE values decreasing to around 10^{-2}, i.e. below σ_{obs}, deriving good estimates for all the variables throughout the estimation period. This is shown for a few unobserved variables in figure 3.12. Note that the estimation period in this case goes until time step 960, after which the prediction begins. The figure also shows that accurate predictions are obtained for around 250 time steps. It is interesting to observe that the increase in the dimension of the system has little impact on the magnitude of the RMSEs obtained and no influence on the delay dimension needed to reach synchronisation.

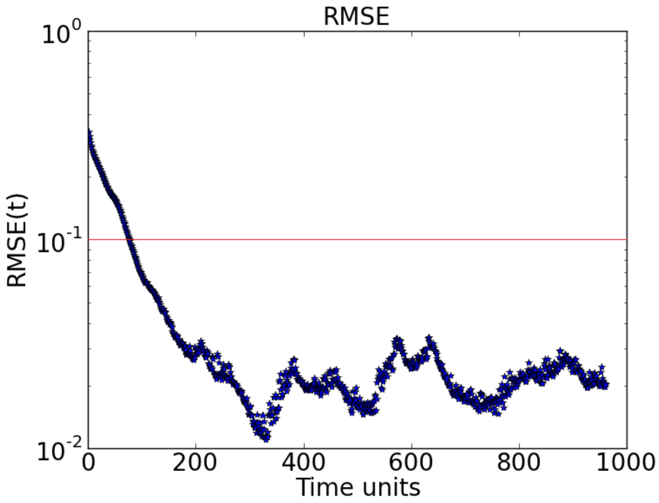


Figure 3.13: *Global Synchronisation Error (RMSE) for a 1000-variable system with 250 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 5$). The horizontal red line represents the standard deviation of the observation noise.*

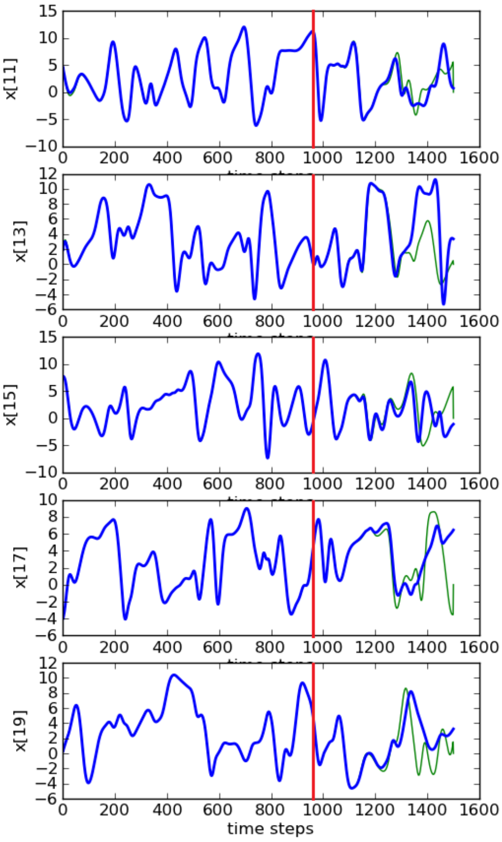


Figure 3.14: *Trajectories of 5 unobserved variables in a 1000-dimensional system. The blue lines are the truth and the green ones are the estimates. Predictions start at time step 960 (red lines).*

3.5.3.5 1000-variable case

In this case 250 variables were observed and the estimation period was also 960 time steps. Although the number of variables has increased, the delay dimension to compensate the lack of observations is still the same, i.e. $D_d = 5$. Figure 3.13 shows that synchronisation errors obtained are of the same order of magnitude compared to the previous case exposed, i.e. they are still below the observation noise introduced. Again, good estimates of all variables are produced and also precise predictions are obtained for more than 250 time steps, as it is shown in figure 3.14. Yet, comparing with the 20 and 100-dimensional cases, there is no dependence on the system size.

3.6 Summary

Although its first discovery dates back to centuries ago, synchronisation is still a phenomenon which requires further understanding and study. In particular, chaotic systems pose a challenge on the predictability of their dynamics, a problem that can be solved with the aid of synchronisation methodologies.

To this end, different techniques have been developed and, noticeably, a convergence of interests between synchronisation and data assimilation has become clearer. Applications of techniques focused on the stabilisation of a synchronisation manifold have been applied to geophysical systems and their performace compared to some well-known data assimilation methodologies. A combination between these two mathematical areas is an interesting and worthwhile endeavour, as both have much to contribute to the advance of problems like the predictability of nonlinear systems.

A framework proposed by Rey et al. (2014a,b) is the inspiration of the research exposed in this thesis. It successfully achieves synchronisation, regardless of the increase of the system dimension, as tested in the present work. Stabilised global RMSEs were obtained for a relatively large-dimensional system of $D_x = 1000$, reaching values below the magnitude of the observation noise introduced, confirming its excellent performance.

A clear obstacle to use this framework in a realistic system like a geophysical one, for example, is the expensive propagation of the Jacobian needed. That was, therefore, the main motivation to develop an ensemble-based framework that could suit geophysical interests, avoiding the propagation of large matrices by the linearised model.

Chapter 4:

ENSEMBLE-BASED SYNCHRONISATION

4.1 Introduction

Chapter 3 has presented a methodology based in the synchronisation theory deriving successful results in the experiments performed. The scheme counts on the time-embedding concept, which improves the observability of the system, an additional motivation to study its application in high-dimensional systems. A main drawback in this formulation, however, is the construction of the Jacobian matrix, which requires the propagation of a $D_x \times D_x$ matrix forward in time. In higher dimensional systems this is clearly prohibitive, as the dimension of this important matrix would make the whole scheme computationally intractable, not to mention the storage issues it would bring.

To overcome this main obstacle, this thesis explores an ensemble framework for the scheme previously presented. This will allow time-embedding synchronisation concepts to be feasible and applicable to high-dimensional systems.

4.2 Ensemble synchronisation framework

4.2.1 Formulation

The ensemble-based methodology proposed here is essentially described by the following steps:

(i) At initial time j, generate an ensemble of N_{ens} members by randomly perturbing x^j with Gaussian perturbations using a prescribed covariance matrix. In the present implementation, this matrix is chosen to be the identity matrix, leading to isotropic perturbations.

(ii) Take the mean of these i perturbed states:

$$\bar{x}^j = \frac{1}{N_{ens}} \sum_{i=1}^{N_{ens}} x_i^j \tag{4.1}$$

and the difference between each member and the mean:

$$\boldsymbol{X}_i^j = x_i^j - \bar{x}^j \tag{4.2}$$

where $\boldsymbol{X}_i^j$ is the i-th column of the ensemble perturbation matrix $\boldsymbol{X}^j$, with dimension $D_x \times N_{ens}$, and N_{ens} is the number of ensemble states or members.

(iii) Propagate forward in time each full ensemble member for τ time steps, where τ is a constant time interval, to form the ensemble perturbation matrix $\boldsymbol{X}^{j+\tau}$ with the same dimensions as $\boldsymbol{X}^j$. Repeat this process for $(D_d - 1)$ times to build ensemble perturbation matrices along the full time window $[j, j + (D_d - 1)\tau]$. As mentioned in the previous chapter, τ is closely connected to a minimal observation frequency required for synchronisation to be reached. This will be explained further at the end of this chapter.

(iv) Generate the augmented D_e-dimensional vectors S and Y, such as in equations (3.12) and (3.13). Note that the states x in $S(x^j)$ now are the ensemble means computed at each observation time.

(v) To compute the Jacobian $\frac{\partial S(x^j)}{\partial x^j}$, we can consider that, approximately:

$$H(\boldsymbol{X}^{j+\tau}) \approx \mathbf{H}\boldsymbol{F}(x)_{j \to j+\tau}\boldsymbol{X}^j \tag{4.3}$$

where, again, $\mathbf{H}$ is the Jacobian of the map H and $\boldsymbol{F}(x)_{j \to j+\tau}$ is the linearised model, i.e. the Jacobian of the nonlinear model, from time j to $j + \tau$. Note that in the nonlinear case, $H(\boldsymbol{X}^j)$ is defined by:

$$H(\boldsymbol{X}^j) = (H(x_1^j) - \overline{H(\boldsymbol{X}^j)}, \cdots, H(x_{N_{ens}}^j) - \overline{H(\boldsymbol{X}^j)}) \tag{4.4}$$

where

$$\overline{H(\boldsymbol{X}^j)} = \frac{1}{N_{ens}} \sum_{i=1}^{N_{ens}} H(x_i^j) \tag{4.5}$$

Following equation (4.3), it is possible to approximately compute the Jacobian $\boldsymbol{F}(x)_{j \to j+\tau}$ as:

$$\mathbf{H}\boldsymbol{F}(x)_{j \to j+\tau} \approx H(\boldsymbol{X}^{(j+\tau)})(\boldsymbol{X}^j)^\dagger \tag{4.6}$$

where the dagger in $(\boldsymbol{X}^j)^\dagger$ means the pseudoinverse of $\boldsymbol{X}^j$. Indeed, this specific pseudoinverse will not be calculated explicitly, as it is shown below. The full Jacobian matrix $\frac{\partial S(x^j)}{\partial x^j}$ is composed by block matrices as shown in chapter 3, as:

$$\frac{\partial S(x^j)}{\partial x^j} = \begin{pmatrix} \mathbf{H}(x) \\ \mathbf{H}\boldsymbol{F}(x)_{j\to j+\tau} \\ \vdots \\ \mathbf{H}\boldsymbol{F}(x)_{j\to j+(D_d-1)\tau} \end{pmatrix}$$

where each row represents a sub-matrix with dimension $D_y \times N_{ens}$. Following the approximation described in equation (4.6), this can be approximated using the ensemble as:

$$\frac{\partial S(x^j)}{\partial x^j} \approx \begin{pmatrix} H(\boldsymbol{X}^j)(\boldsymbol{X}^j)^\dagger \\ H(\boldsymbol{X}^{(j+\tau)})(\boldsymbol{X}^j)^\dagger \\ \vdots \\ H(\boldsymbol{X}^{(j+(D_d-1)\tau)})(\boldsymbol{X}^j)^\dagger \end{pmatrix}$$

This way, it is now possible to compute the pseudoinverse of $\frac{\partial S(x^j)}{\partial x^j}$, considering that each row in the matrix described above has the factor $(\boldsymbol{X}^j)^\dagger$ in common, so deriving the following:

$$\frac{\partial S(x^j)}{\partial x^j}^\dagger \approx \boldsymbol{X}^j \begin{pmatrix} (H(\boldsymbol{X}^j)) \\ (H(\boldsymbol{X}^{(j+\tau)})) \\ \vdots \\ (H(\boldsymbol{X}^{(j+(D_d-1)\tau)})) \end{pmatrix}^\dagger$$

This way, the computation of the pseudoinverse of a $D_e \times N_{ens}$ measured ensemble perturbation matrix, is now needed. This can be done via a truncated SVD, as explained in chapter 3.

(vi) Use equation (3.15), which contains the synchronisation term to propagate the model states forward in time:

$$x^{j+1} = f(x^j) + g\frac{\partial S(x^j)}{\partial x^j}^\dagger (Y^j - S(x^j)) \tag{4.7}$$

(vii) Keep iterating the whole process (stages (i) to (vi)) from time $j+1$ on, whilst there still exists an available observation at the end of the time window.

For a clearer computational description of the ensemble-based synchronisation scheme proposed here, refer to Appendix B, which presents a pseucode of these main steps.

By using the ensemble synchronisation scheme, it is possible to reduce the computational cost by a factor of at least D_x/N_{ens}, compared to the synchronisation scheme described in chapter 3, where the computationally demanding propagation of a matrix of size $D_x \times D_x$ was required for $(D_d - 1) \times \tau$ time steps, at each model time step.

Additionally to this huge improvement, computational costs are also reduced after the implementation of a localisation method. Localisation schemes appear as one of the solutions for the rank-deficiency issues brought by the use of a small number of ensemble members, compared to the size of the system it is applied to (Houtekamer and Mitchell, 1998). As the number of members, in practice, is much smaller than the system dimension, unstable degrees of freedom can lead the system to divergence and instability. Furthermore, spurious correlations at long distances are generated, leading to artificial updates. A solution to overcome this issue is the implementation of localisation, aiming to reduce the influence of observations which are far away from the variables, considering that they both are mainly uncorrelated and independent (assuming, at least, a short-range scale). To this end, only observations which are within a pre-fixed localisation radius from the variable point are assimilated. Note that the radius should be set small enough to avoid the spurious correlations, but large enough to allow for physically meaningful correlations in the system (e.g. Asch et al. (2016)).

Assuming that a position in physical space (e.g., a location on the Lorenz ring) is attributed to each observation, the implementation of localisation in the ensemble synchronisation system follows these steps:

a) For each variable i, compute the distance $d_{i,l}$ (along the ring for the case of Lorenz96 model, for example) between its position in physical space and each observation l.

b) Define a threshold loc for this distance $d_{i,l}$. If this limit is exceeded, consider the influence of this observation in the variable as equal to zero. Otherwise, compute the influence $iloc_{i,l}$ for the variable through:

$$iloc_{i,l} = e^{-(d_{i,l}^2/(2(loc^2)))} \tag{4.8}$$

and store each value in a vector related to i: **iloc** $\in \Re^{D_e}$ (remember that $D_e = D_d * D_y$).

c) Calculate the Schur product (denoted by a $\circ$) between **iloc** and the innovation $(Y^j - S^j)$ and right-multiply this vector by the i-th row of $\frac{\partial S(x^j)}{\partial x^j}^\dagger$ to compose each element of the D_x-dimensional coupling vector

$$x_i^{j+1} = f(x_i^j) + g\left(\frac{\partial S(x^j)}{\partial x^j}^\dagger\right)_i (\mathbf{iloc} \circ (Y^j - S(x^j))) \tag{4.9}$$

Regarding implementation procedures, it is worth to mention that, in order to make steps (v) and (vi) in this ensemble-based scheme more computationally efficient, one may first calculate:

$$[((H(\boldsymbol{X}^j)), (H(\boldsymbol{X}^{(j+\tau)})), \cdots, (H(\boldsymbol{X}^{(j+(D_d-1)\tau)})))^T]^\dagger (\mathbf{iloc} \circ (Y - S)) \tag{4.10}$$

which results in a vector (with dimension proportional to N_{ens}). Then, right-multiply this vector by $\boldsymbol{X}^j$ to complete the computation of the whole pseudoinverse of the Jacobian. This procedure is particularly more efficient in higher dimensional cases.

4.2.2 A connection between ensemble synchronisation and other data assimilation methods

The methodology described in the previous section has strong similarities with an Ensemble Smoother (van Leeuwen and Evensen, 1996; van Leeuwen, 1999; Evensen and van Leeuwen, 2000). Basically, the Ensemble Smoother update can be formulated as:

$$x^{j+1} = f(x^j) + \boldsymbol{X}(H(\boldsymbol{X}))^T(H(\boldsymbol{X})(H(\boldsymbol{X}))^T + R)^{-1}(Y^j - S(x^j)) \tag{4.11}$$

The main differences between Ensemble Smoothers and the Ensemble-based Synchronisation are the following:

1. Synchronisation ignores observation errors in the gain, i.e. it does not include the observation error covariance matrix R in this term. This allows for an inversion of $H(\boldsymbol{X})$ directly, instead of $(H(\boldsymbol{X})(H(\boldsymbol{X}))^T + R)$. The latter is a larger matrix, but efficient implementations are available, in which the matrix to be inverted in an ensemble smoother is of size $N_{ens} \times N_{ens}$. Localisation, though, makes the matrices in both methods to have similar sizes. The fact that the ensemble synchronisation ignores the observation error covariance matrix is maybe a disadvantage in the scheme, but this weakness is partly compensated for by a very useful tool, the tuning factor g which appears in equation (4.9). (In the experiments performed in this chapter and in the previous one, this powerful feature of the coupling constant g is not fully explored. However, the importance of this tuning parameter will be highlighted in the following chapters, where the ensemble synchronisation scheme will be effectively used as an important piece in a particle filter).
2. Observations are used multiple times in this time-embedded framework. That would be considered a forbidden practice in an ensemble smoother, as using the same observations multiple times would introduce correlations between the errors of the state iterates and the observation errors, bringing significant complications to the scheme. Ensemble synchronisation, however, is not hampered by this issue, and observations inside the time embedding interval are used several times, helping to increase the observability of the system. This is a crucial advantage of the ensemble synchronisation scheme, which will be used when this new method is inserted in the context of proposal densities in a particle filter, following e.g. van Leeuwen (2009), who shows that observations can be used in a proposal density as often as one would like to.

There are also relations between the ensemble-based synchronisation and variational methods that employ ensembles to avoid adjoint models, like, e.g. 4DEnsVar (Liu et al., 2008; Fairbairn et al., 2014; Gustafsson and Bojarova, 2014), and iterative Ensemble Smoothers like the IEnKS (Iterative Ensemble Kalman Smoother) of Bocquet and Sakov (2014), as these systems also explore the space-time correlations from ensembles, as pioneered by van Leeuwen and Evensen (1996).

The formulation of the 4DEnsVar is based on the use of an ensemble to estimate sensitivities to observations within a data assimilation window, i.e. replace the calculation of adjoint models with an ensemble of perturbations (Liu et al., 2008). The drawback of this strategy is that the method inherits the rank-deficiency issue, when applied to high-dimensional systems. Consequently, localisation is required as well, although a spatial localisation in a 4D environment is not that trivial, as one will need to localise covariances inside the data assimilation window between state variables and observations at different times (asynchronously). Additionally, regularisation in this case is not straight-forward, as explained in detail in Asch et al. (2016).

The IEnKS is a methodology derived from Bayes' rule, which also estimates sensitivities by using an ensemble and so preventing adjoint calculations. Like the ensemble synchronisation scheme, this method also assimilates observations ahead in time, aiming for stabilisation in the direction of the unstable modes. However, differently from the scheme presented in this work, the IEnKS reuses all the observations inside the data assimilation window, whereas the ensemble synchronisation uses only the observations that appear at every τ time steps in the $[j, j+(D_d-1)\tau]$ window. Additionally, the IEnKS scheme uses observations multiple times, but with inflated covariances. In practice, this approach factorises the likelihood and assimilates the resulting sequence of likelihoods sequentially. Note that, differently from the ensemble synchronisation, this reuse of observations does not change the observability of the system.

It is important to keep in mind that the ensemble synchronisation scheme targets a different purpose, when compared to the formal data assimilation methodologies. While data assimilation methods aim to approximate the true posterior pdf, the ensemble synchronisation technique proposed here tries to find a model trajectory that follows the observations closely, to synchronise with the truth. Therefore, as mentioned earlier, ensemble synchronisation is not a complete data assimilation method, as uncertainties in model and observations are not incorporated explicitly in the formulation. An efficient use for this scheme will be shown later in following chapters, where it is included as part of a proposal density, in a more comprehensive data-assimilation method, a particle filter.

The results to be presented in the next sections of this chapter are a motivating engine to extend the use of this scheme to the particle filtering world, as it properly fits to the needs of this data assimilation methodology. Pre-existing data assimilation methods were, so far, not good enough to be joined as a proposal density in a particle filter. Authors like Browne (2016) did use a proper Ensemble Kalman smoother as a proposal, without obtaining satisfying results. In this thesis, a new and exciting field is explored, in conjunction with a particle filter, a data assimilation method that receives an increasing attention in the geosciences field.

4.2.3 Experiment configuration and results in Lorenz96 model

For these experiments, the basic configuration used in the experiments in chapter 3 is kept: a 4th-order Runge-Kutta scheme is used, with Δt = 0.01, a constant observation time interval $\tau = 10\Delta t$ and a measurement noise with a standard deviation of σ_{obs} = 0.1. The coupling constant used in the following experiments is $g = 100$. Tests were again performed for systems with 20, 100 and 1000 variables.

The same proportion of the system is observed (25%), with equally distributed observations along the Lorenz96 ring. Again, it is important to remember that, inside the embedding window, only the observations that appear at every τ time steps are used.

Extensive tests were performed, including different ensemble sizes, delay dimensions D_d and measurement noise standard deviations σ_{obs}. Ensemble members were slightly perturbed with a normal distribution of $N \sim (0, 0.01)$. Tests were performed for different ensemble sizes: 5, 15, 20, 50, 100 and 200 members. However, the main results shown in this thesis will correspond to the minimal number of ensemble members needed in each case, characterising the cheapest configurations, computationally speaking. In order to reach computationally stable results, the number of singular values used in the calculation of the SVD in this framework is equal to the number of ensemble members used. This is valid for all the ensemble cases shown next.

4.2.3.1 20-variable case

A main point to start with in these experiments is the number of ensemble members used to achieve synchronised results, as the main purpose of implementing an ensemble version of the synchronisation framework is to make it more computationally feasible for realistic systems. For this 20-variable system case, an ensemble of only 5 members was used, presenting very precise results, as it will be described in this section. This is a first positive point in the proposed ensemble synchronisation scheme.

Regarding the minimum embedding interval to make the system stabilize and synchronise, figure 4.1 shows the synchronisation errors (RMSE) for different D_ds. In these results, it is noticeable that for $D_d = 2$, the system does not synchronise (i.e., RMSEs are not below the magnitude of the measurement noise), nor stabilise. For $D_d = 3$ stabilisation is not reached, as RMSEs show peaks of desynchronisation during the run. For $D_d = 4$ and $D_d = 5$, lower RMSE values are reached. It is clear, however, that for $D_d = 5$ the system tends to stabilize faster, synchronising more consistently. Tests were also performed for delay dimensions varying from $D_d = 6$ to $D_d = 12$, showing synchronisation in all of these cases. Nevertheless, for $D_d = 11$

and $D_d = 12$, desynchronisation was observed at some time steps during the run, although their RMSEs kept mean values of the order of 10^{-2}. This behavior may suggest that the choice of the embedding interval must obey a size limit, as information from observations in a distant future time may not be correlated to the present time anymore, thus contributing less efficiently in the system's results. (Note that the implementation of a space-time localisation could minimise this problem). Also, this behaviour agrees with Rey et al. (2014b), who observed that, for the Lorenz96 model, regardless the size of the system, a threshold of $\approx D_d = 12$ should be obeyed to prevent the ill-conditioning of the Jacobian.

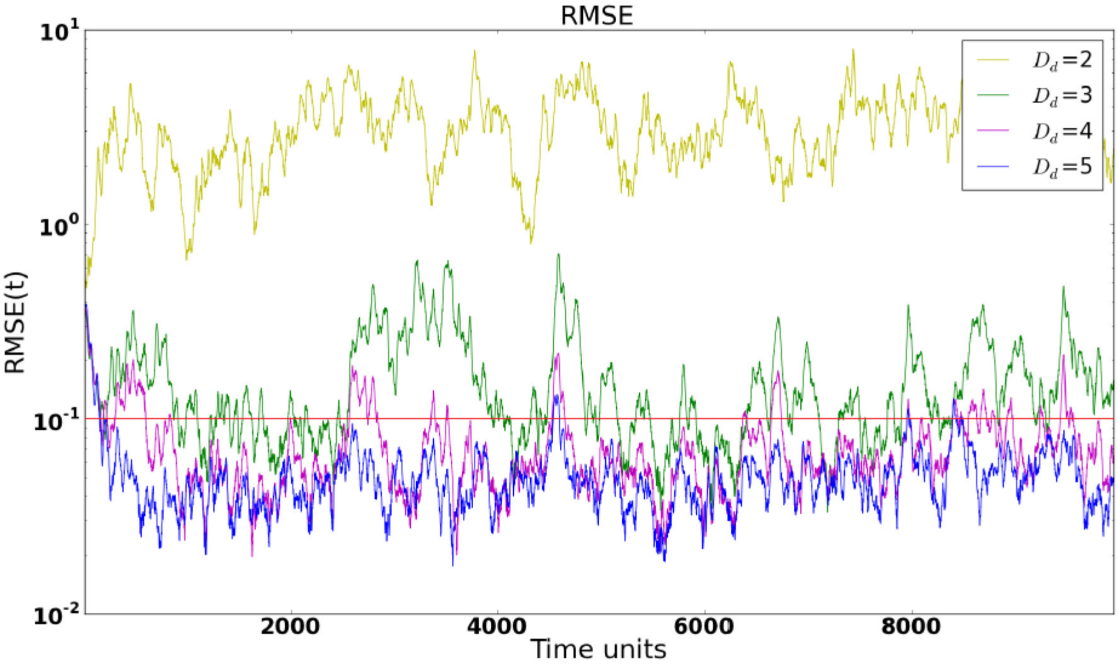
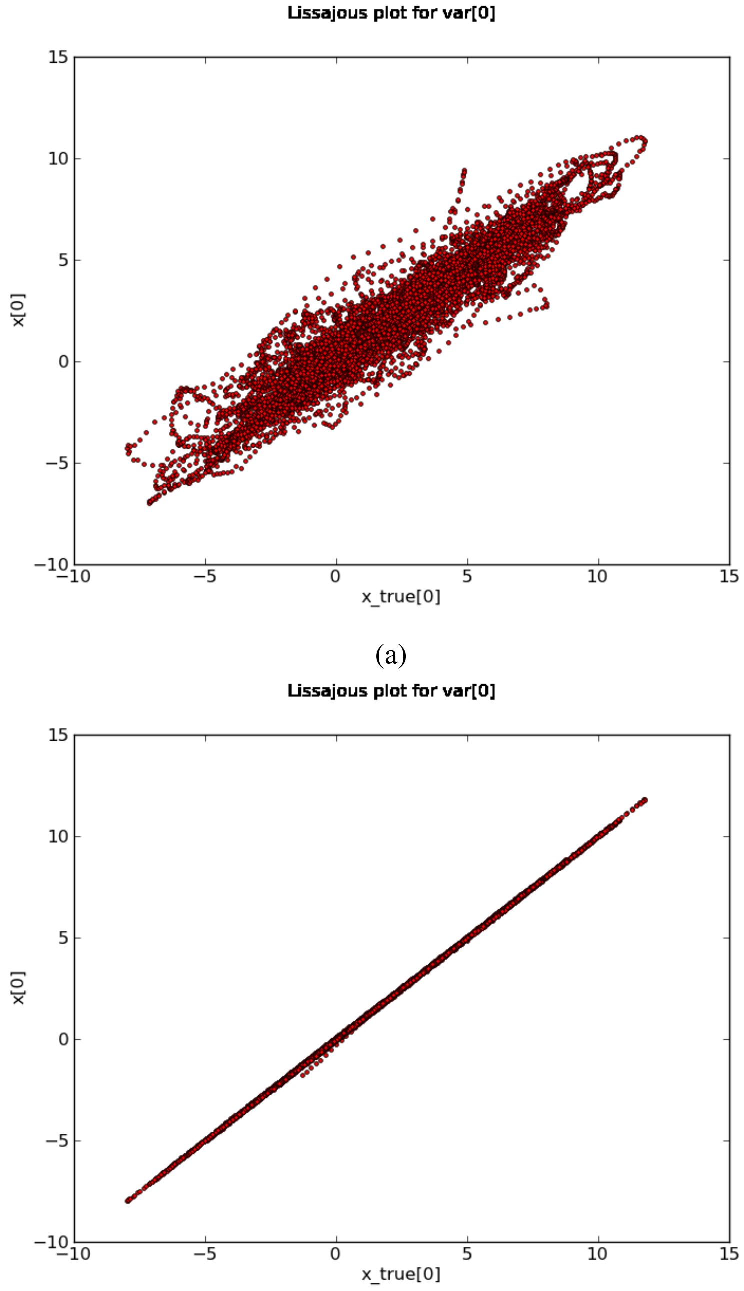


Figure 4.1: *Global Synchronisation Error (RMSE) for different delay dimensions (D_d) in a 20-variable system with 5 measured variables sampled equidistantly on the Lorenz96 ring. Ensemble Synchronisation was used with 5 members. The horizontal red line represents the standard deviation of the observation noise.*

A noticeable point is that the synchronisation results presented in the previous chapter showed slightly lower RMSEs, compared to this ensemble-based scheme. This may be closely related to the small number of ensemble members used to run the system. Yet, the ensemble-based results reach RMSE values well below the observation noise added, showing a synchronised and stabilised system, if a suitable delay dimension is chosen.

Figure 4.2 shows how generalised synchronisation happens when the delay dimension D_d is increased from 2 to 5. A comparison with figure 3.5 shows that, for the ensemble case, a time embedding $D_d = 2$ is still not enough to reach synchronisation, while the framework presented in the previous chapter using the propagation of the Jacobian obtained values which were closer to generalised synchronisation in this case. Figure 4.3 justifies this result by showing a larger amount of positive leading Lyapunov exponents along the run, describing a pattern of instability growth



(a)

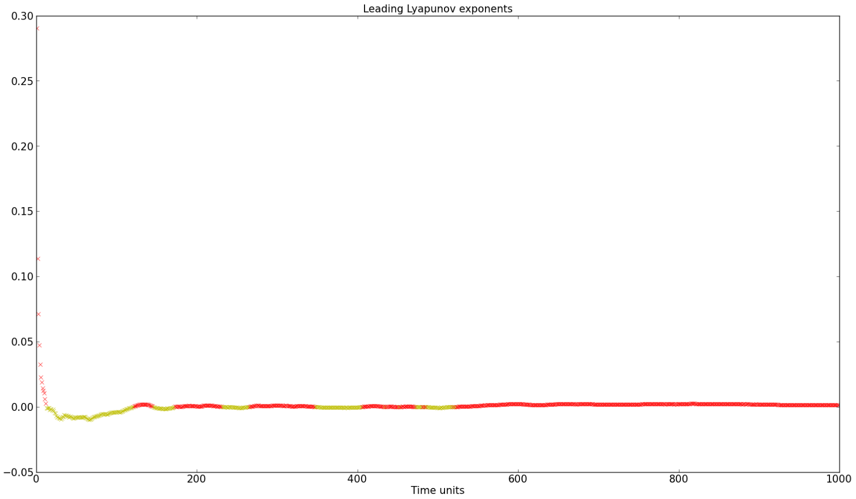
(b)

Figure 4.2: *Lissajous-type plots for variable* $[0]$ *during 10,000 time steps, using different time embeddings: (a)* $D_d = 2$ *and (b)* $D_d = 5$.

in the system, a different behaviour, if compared to the one presented in figure 3.6. We note that figure 4.3 (b) clearly shows that negative leading lyapunov exponents are related to generalised synchronisation, when we analyse it with figure 4.2 (b).

In figure 4.4 another type of experiment is performed, in which the number of ensemble members is varied. The figure shows how it is possible to decrease the number of positive leading Lyapunov exponents by increasing the ensemble size.

Another useful test was performed, showing the impact of the magnitude of the measurement



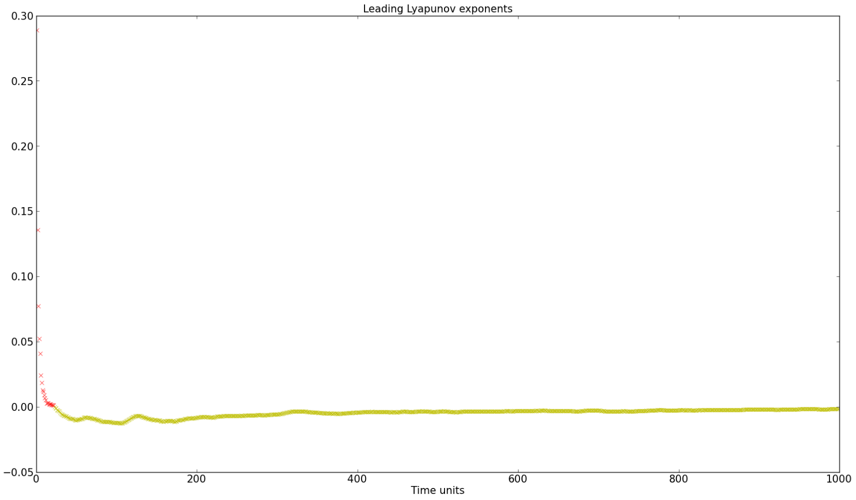
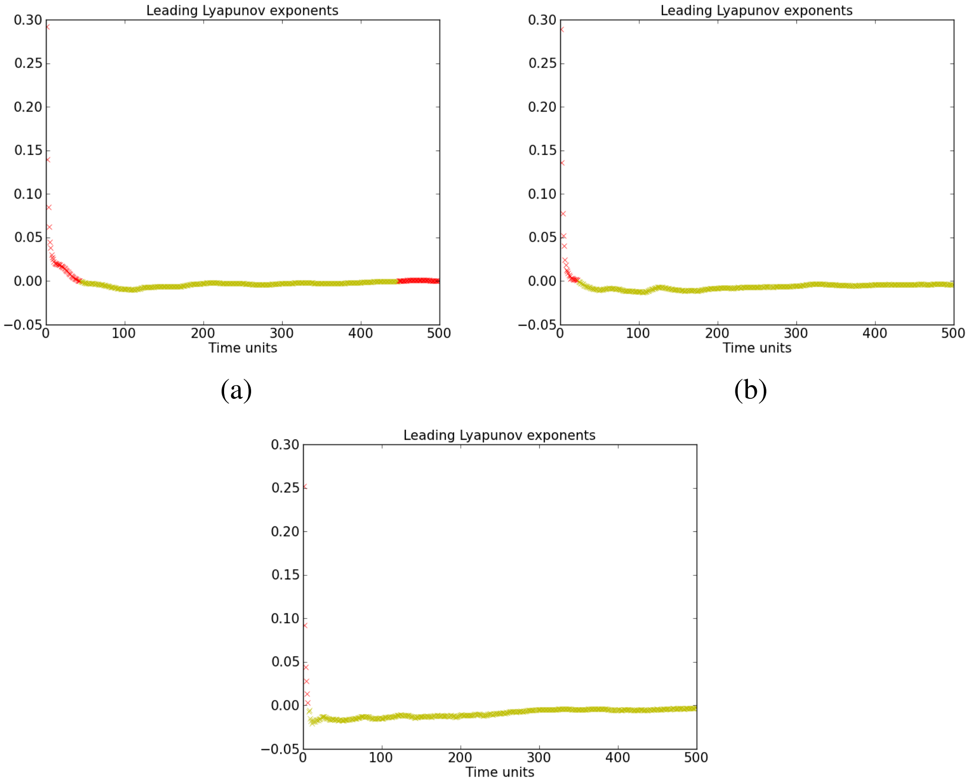


Figure 4.3: *Leading lyapunov exponents along 1,000 time steps for $D_x = 20$ and $N_{ens} = 5$: (top) $D_d = 2$ and (bottom) $D_d = 5$. The red crosses are positive lyapunov exponents and the yellow ones are negative lyapunov exponents.*

noise in combination with the ensemble size. Figure 4.5 shows that, for 5 members, the system reaches the lowest RMSEs for standard deviations of $\sigma_{obs} = 0.01$ or below this value. This suggests that Monte-Carlo noise is an important contributor to the RMSE in this system. To investigate this further, the same case was run with 100 ensemble members (Figure 4.6). As the number of members increase, the system tends to get more stabilised at $\sigma_{obs} = 0.1$, reaching smaller RMSE values for all measurement noise levels, compared to the results in figure 4.5. These results are also linked to what was mentioned in the previous paragraph, where a relationship between the number of members in the ensemble and the minimum RMSEs reached was pointed out.

In figure 4.7 we can analyse the trajectories of the first 10 variables and the estimates and predictions resulted from the ensemble-based system. Similarly to results presented in chapter



(a)

(b)

(c)

Figure 4.4: *Leading lyapunov exponents along 500 time steps for $D_x = 20$ using: (a) $N_{ens} = 3$, (b) $N_{ens} = 5$ and (c) $N_{ens} = 10$. The red crosses are positive lyapunov exponents and the yellow ones are negative lyapunov exponents.*

3, the estimates are precise and close to the truth. During prediction stage, the range of precise results have reduced, compared to the ones shown in figure 3.8, but accurate values are reached for around 300 time steps, which is still a considerable period of prediction time in this case.

4.2.3.2 100-variable case

In this experiment, 25 observed variables are considered, with an estimation period of 1,000 time steps. The delay dimension tested is, again, $D_d = 5$.

By increasing the number of states in the system to 100, the minimal number of ensemble members required to stabilize the system and make the model synchronise with the truth also increases, from 5 to 15. Figure 4.8 (left) shows good results for this configuration, with RMSE values decreasing to a magnitude of 10^{-2}.

As mentioned before, the main purpose of this work is to build a framework that is applicable to high-dimensional systems. Therefore, there is an increase need to implement a localisation method to reduce the influence of spurious correlations arising from using a proportionally small ensemble size, compared to the increased dimension of the system. To this end, a localisation radius of influence $rad = 3$ is chosen, so that any observations located further than the threshold

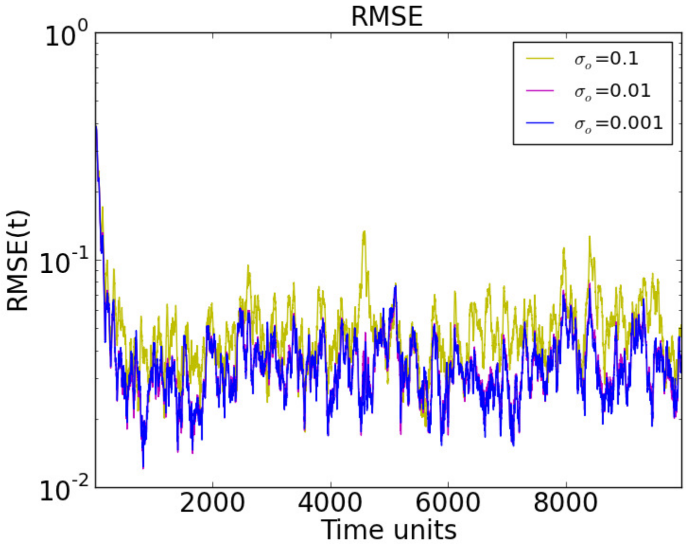


Figure 4.5: *Global Synchronisation Error (RMSE) for different standard deviations (σ) of observation noise in a 20-variable system with 5 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 5$). Ensemble Synchronisation was used with 5 members.*

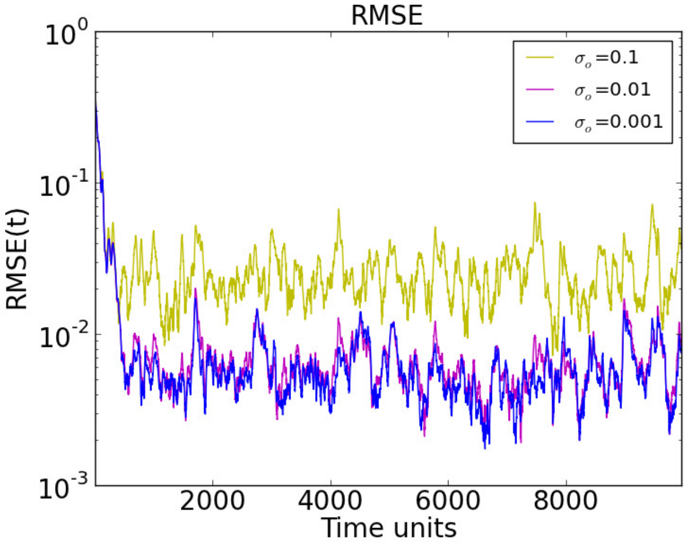


Figure 4.6: *Global Synchronisation Error (RMSE) for different standard deviations (σ) of observation noise in a 20-variable system with 5 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 5$). Ensemble Synchronisation was used with 100 members.*

$loc = 3 * rad$ are ignored by the variable. Note that the localisation function has an exponential shape. Equation ((4.9)) is then used to evolve the variables with time.

By localising the effect of these distant observations in the variables, it is possible to reduce the number of ensemble members from 15 back to 5. Figure 4.8 (right) shows that localisation allows a factor of 3 reduction of the number of ensemble members previously used in the non-localised experiment, and besides that, it additionally makes the system synchronise and stabilise even faster. Good estimates and predictions for 3 unobserved variables in this localised case are shown in figure 4.9.

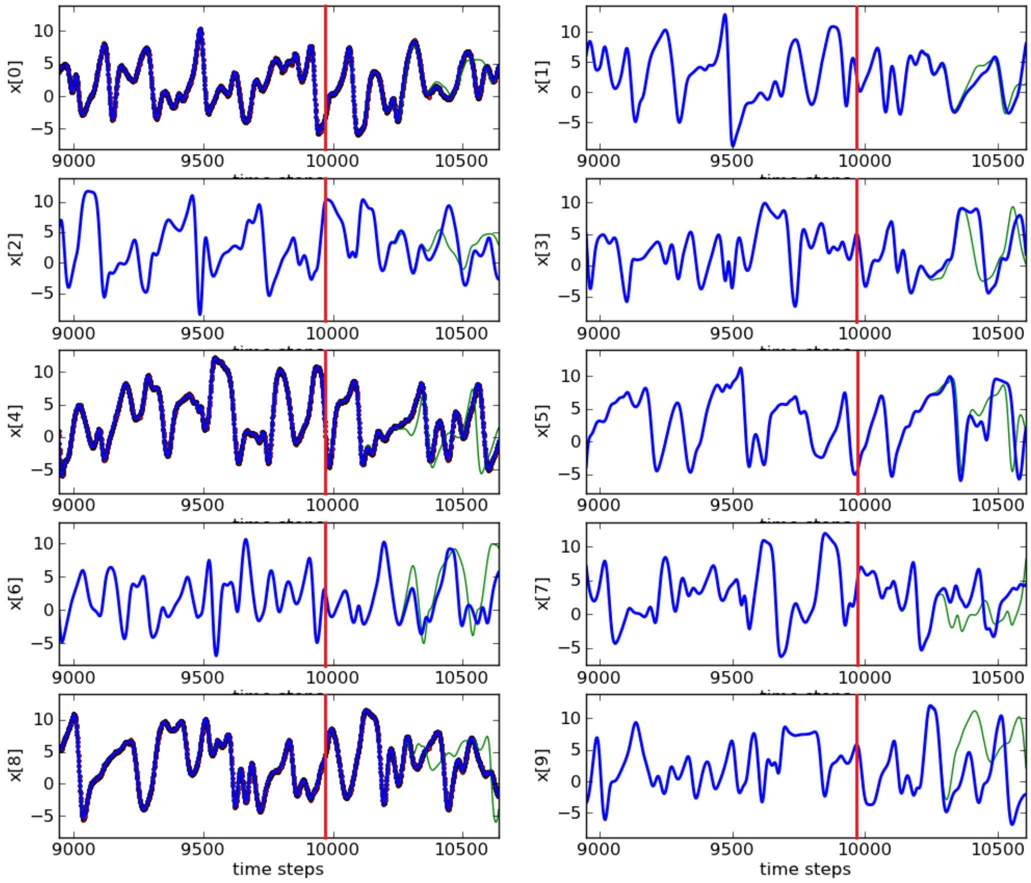


Figure 4.7: *Trajectories of the first 10 variables (y axis) and our* ***estimates*** *(until 9,960 time steps - red lines)/****predictions*** *(after 9,960 time steps), for the ensemble-based synchronisation system. The blue lines are the truth and the green ones are the estimates/predictions. The observed variables have small red dots in their trajectories (variables 0, 4 and 8).*

Figure 4.10 shows how much influence localisation has in the system, comparing runs using the same number of ensemble members, but turning localisation on and off. Clearly, the synchronisation manifold is far from being stabilised without the use of localisation.

4.2.3.3 1000-variable case

For this experiment, 250 variables are observed for the same estimation period used in the 100-variable case. As the dimension of the system increases, localisation becomes crucial. By running the configuration without localisation, the number of ensemble members required to make the system synchronise increases significantly, to 100 members. Additionally, the delay dimension needs to increase to $D_d = 10$ for synchronisation to set in (figure 4.11 (left)).

By using localisation with a radius of influence $rad = 10$, the number of ensemble members is drastically reduced from 100 to 20. Also, it is possible to reduce the embedding interval, from $D_d = 10$ back to a desirable $D_d = 5$. Figure 4.11 (right) shows a more synchronised and stabilised result, reaching lower RMSEs, if compared to the one without localisation (figure 4.11

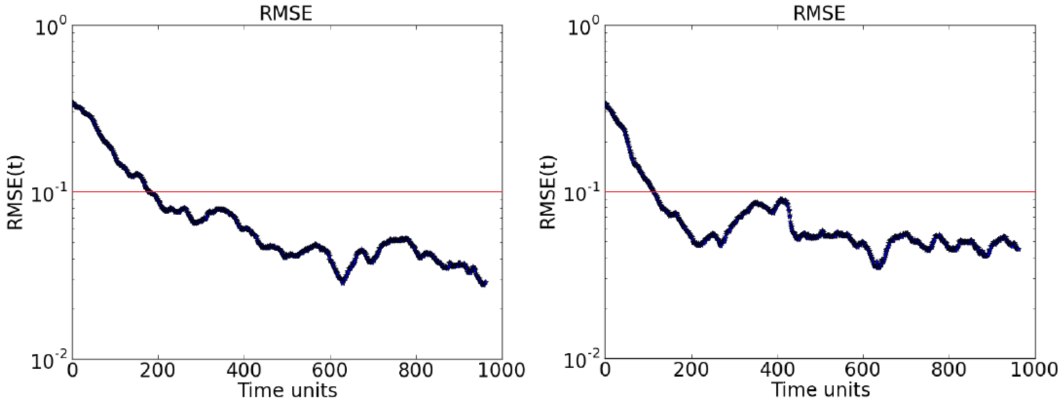


Figure 4.8: *Global Synchronisation Error (RMSE) for a 100-variable system with 25 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 5$). Left: 15 ensemble members, without localisation. Right: 5 ensemble members, with localisation applied. The horizontal red lines represent the standard deviation of the observation noise.*

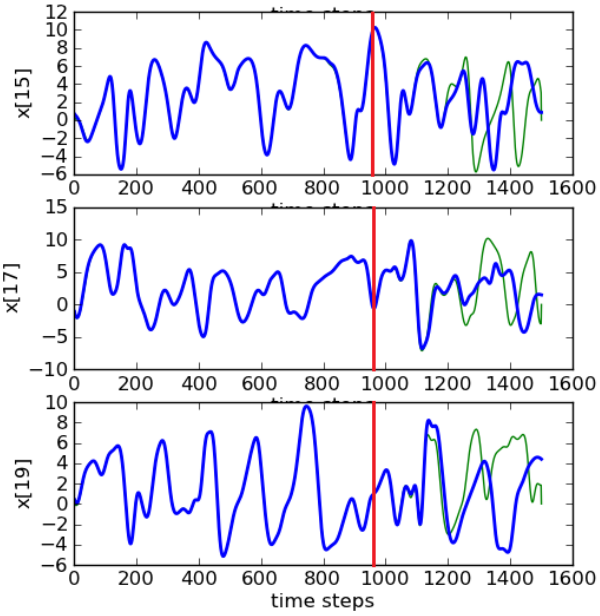
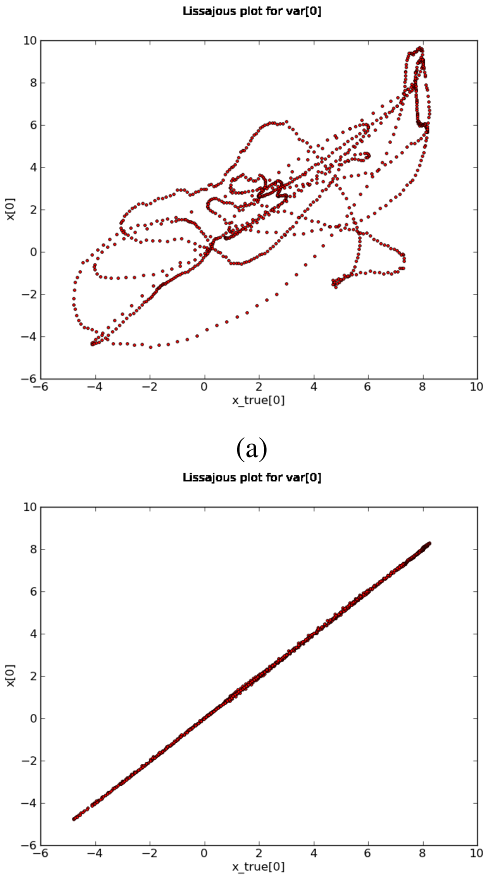


Figure 4.9: *Trajectories of 3 unobserved variables in a 100-dimensional system for the ensemble-based synchronisation scheme with localisation and 5 ensemble members. The blue lines are the truth and the green ones are the estimates. Predictions start at time step 960 (red lines).*

(left)). This localised experiment also produces excellent estimates and predictions, as can be seen in figure 4.12, which shows trajectories for 5 unobserved variables in the system.

So far, the experiments performed considered that variables were measured at every time step. We now consider a system in which observations occur less frequently, e.g. at every other time step. The methodology used in this case to evolve the states in time is:

1. at observed time steps: compute the Synchronisation Term $g\frac{\partial S(x^j))}{\partial x^j}^{\dagger}(\mathbf{iloc} \circ (Y^j - S^j))$ and update the variables using equation 4.9.
2. at unobserved time steps: the same Synchronisation Term computed in the observed time



(a)

(b)

Figure 4.10: *Lissajous-type plots for variable* $[0]$ *during 1,000 time steps using* $N_{ens} = 5$*: (a) without localisation and (b) with localisation.*

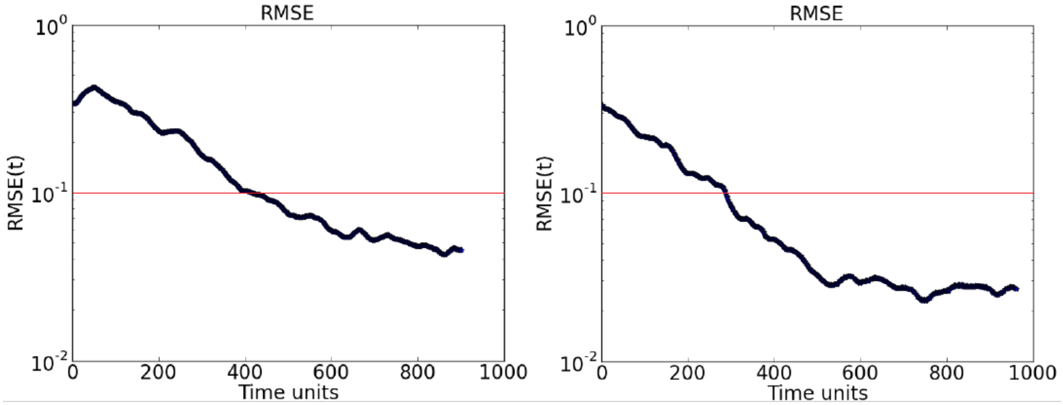


Figure 4.11: *Global Synchronisation Error (RMSE) for a 1000-variable system with 250 measured variables sampled equidistantly on the Lorenz96 ring. Left: 100 ensemble members, without localisation (*$D_d = 10$*). Right: 20 ensemble members with localisation applied (*$D_d = 5$*). The horizontal red lines represent the standard deviation of the observation noise.*

step is used to update the variables.

By adopting the strategy of reusing the synchronisation term previously computed, the computational cost of the system gets smaller, as this term is computed only half of the time.

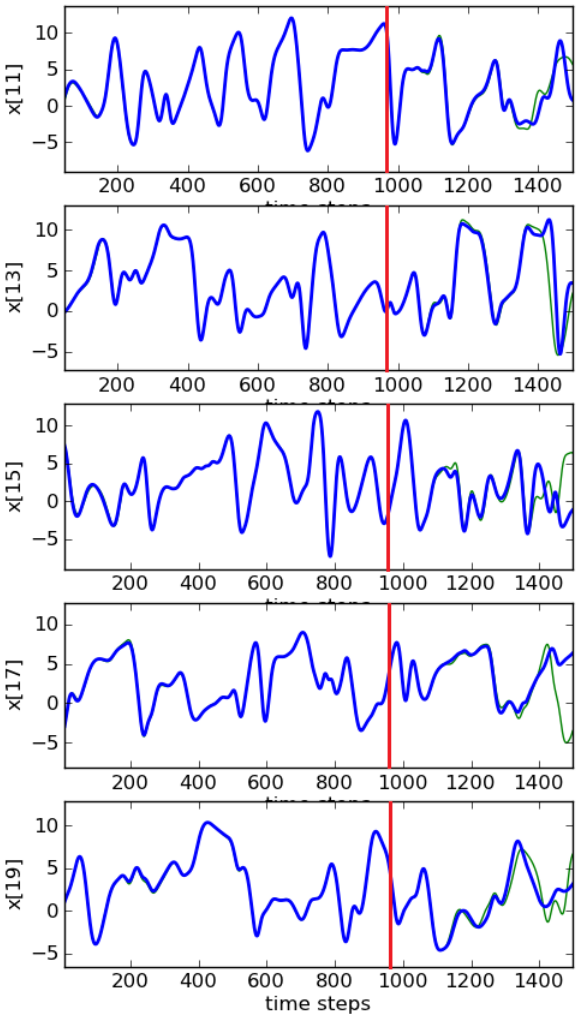


Figure 4.12: *Trajectories of 5 unobserved variables in a 1000-dimensional system for the ensemble-based synchronisation scheme with localisation and 20 ensemble members. The blue lines are the truth and the green ones are the estimates. Predictions start at time step 960 (red lines).*

Figure 4.13 shows a result for the synchronisation error with practically the same quality obtained previously in figure 4.11 (right), now in a less frequently observed system. This result reflects how information contained in the Synchronisation Term is still valid and useful for the next time step.

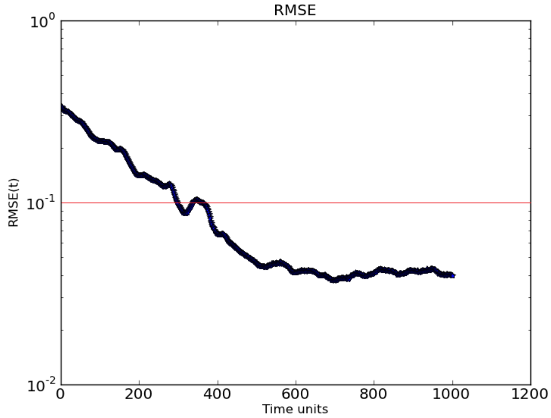


Figure 4.13: *Global Synchronisation Error (RMSE) for a 1000-variable system with 250 measured variables sampled equidistantly on the Lorenz96 ring and* ***observed at every other time step*** *($D_d = 5$). Ensemble Synchronisation was used with 20 members. The horizontal red line represents the standard deviation of the observation noise.*

Aiming to increase the complexity of the problem even more, the frequency of observations is significantly decreased. Now they are sparser and occur at every 10 time steps. The methodology used in this case to evolve variables in time is as follows: at observed time steps we compute the coupling term $g\frac{\partial S(x^j))}{\partial x^j}^{\dagger}(\textbf{iloc}\circ(Y^j - S^j))$ from equation (4.9). To update the variables at unobserved time steps, we use a strength factor $gtau$, which is progressively increased by n along the interval between observations: $g*(n*gtau)\frac{\partial S(x^j)}{\partial x^j}^{\dagger}(\textbf{iloc}\circ(Y^j - S(x^j)))$. Therefore, during the time steps without observations, n is increased from $n = 1, 2, ...,$ (observation interval -1), until the last unobserved time step in the interval. The idea is to have a combination between g and $gtau$ such that the coupling term is strong enough to bring the model towards the observations and weak enough to prevent the model to lose its trajectory around the attractor. This strategy is different from the one adopted by e.g. Yang et al. (2006), who tested time interpolating the observational increments in an infrequent observations case.

Different ensemble sizes were tested, but in this case, optimal results were generated for $N_{ens} = 30$, the cost of having much less available observations in the system.

As mentioned before, the coupling constant g is a tuning parameter in the synchronisation framework. In the experiments presented here for sparse observations, a coupling constant $g = 70$ is set. The strength factor $gtau$ used at unobserved time steps is $gtau = 0.3$ for the results shown here. Although these values seem very specific for a experiment, we will see later that there is a range of possible combinations between these two parameters to obtain synchronisation.

Apart from the details described above and the fact that the estimation period was slightly extended (until 1,000 time steps), this experiment for observations at every 10 time steps uses the same settings as the previous ones, including the size of the delay dimension, which is $D_d = 5$.

Figure 4.14 shows the global RMSEs during the estimation period. It is possible to note how the RMSEs decay exponentially until the order of magnitude of the observation noise added to the system. As observations appear less frequently in this test, some considerations can be made: 1) 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 $\leq \sigma_{obs}$; 2) the proportional reuse of the coupling term in the gaps between the occurrence of the observations is indeed useful to keep a decaying RMSE and 3) more members in the ensemble are needed.

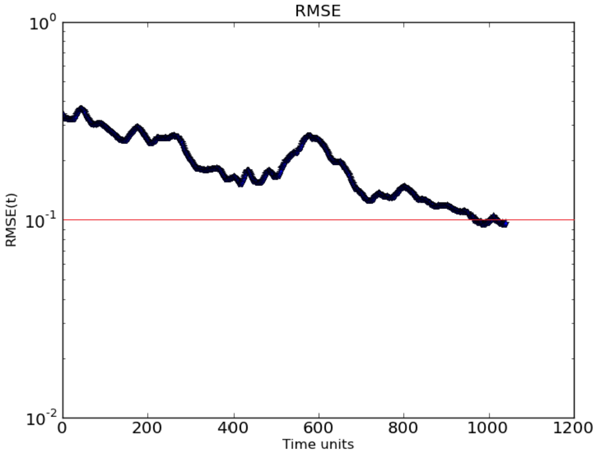


Figure 4.14: *Global Synchronisation Error (RMSE) for a 1000-variable system with 250 measured variables sampled equidistantly on the Lorenz96 ring and **observed at every 10 time steps**. Ensemble Synchronisation was used with 30 members. ($D_d = 5$). The horizontal red line represents the standard deviation of the observation noise.*

For the assimilation period, figure 4.15 shows estimations and predictions for 5 unobserved variables in the system. Despite the lack of frequent observations, the estimates still almost perfectly match the truth and during the prediction period, trajectories keep following the truth for around 200 time steps, followed by divergence, related to the chaotic nature of the model, as explained before.

Figure 4.16 shows the leading lyapunov exponents 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 the figure). From $t = 200$ until the end of the estimation period, only negative exponents are observed (not shown in the figure, for the sake of clarity), which means that the growth of instabilities are supressed and trajectories tend to converge towards the attractor, as can be seen during the assimilation period in figure 4.15.

Figure 4.17 describes the relationship between parameters g and $gtau$ for the occurrence of synchronisation, observed for different initial conditions. The green zones in the graph are the optimal combinations to reach RMSE $\leq \sigma_{obs}$. i.e synchronisation and stabilisation can be achieved. The yellow zones determine a combination in which global RMSEs are above σ_{obs}, yielding synchronisation with possible bursts of desynchronisation. The shaded gray zones are combinations between g and $gtau$ that derive a final coupling term which is too weak or too strong

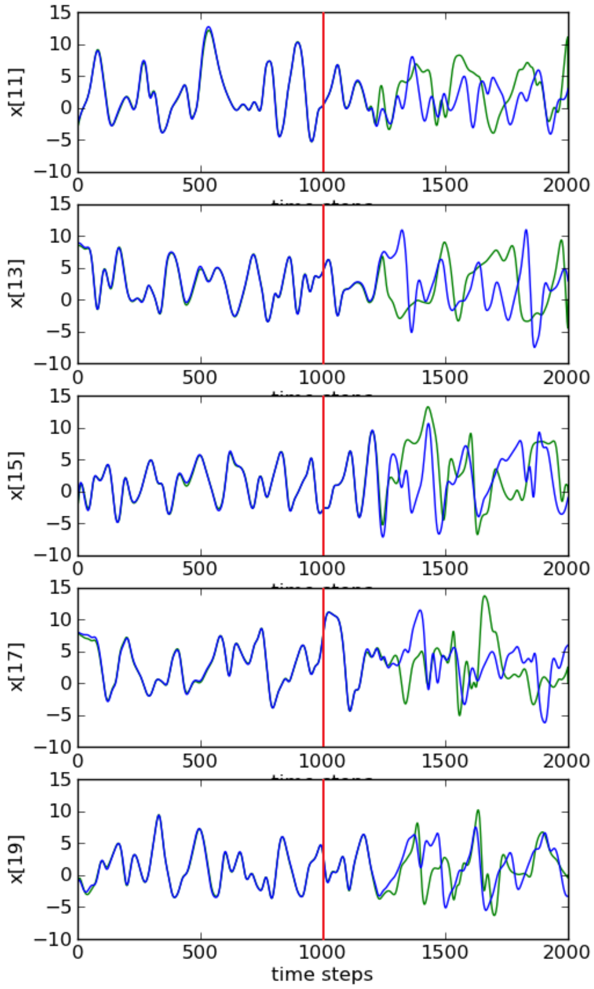


Figure 4.15: *Trajectories of 5 unobserved variables in a 1000-dimensional system* ***observed at every 10 time steps*** *for the ensemble-based synchronisation scheme with localisation and 30 ensemble members. The blue lines are the truth and the green ones are the estimates. Predictions start at time step 1,000 (red lines)*

to synchronise. An overall analysis of the figure also shows that $g * gtau \approx 20$.

These results show that the ensemble-based synchronisation scheme is very useful in steering model states to the truth, working well in a moderately high dimensional system and 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.

4.2.3.4 Further tests

To better investigate the importance of time embeddings for increasing the observability of a system and improve its space-time correlations when an ensemble is used, tests were performed considering $D_d = 1$, i.e. a case in which no time embeddings are used. Figure 4.18 demonstrates that the estimates do not synchronise below observation noise levels with only 25% of the system observed. Even without considering the use of an ensemble, which would be the case of the

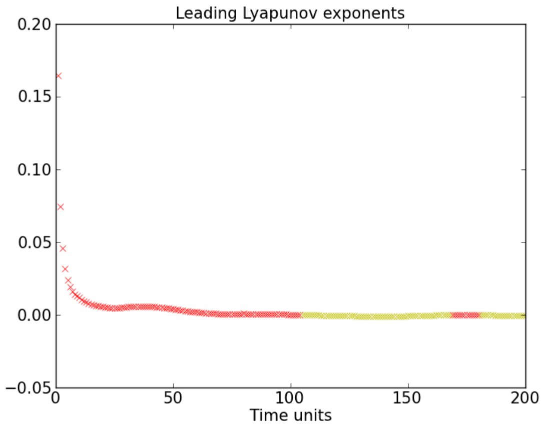


Figure 4.16: *Leading lyapunov exponents along 200 time steps in a 1000-dimensional system* ***observed at every 10 time steps****. The red crosses are positive lyapunov exponents and the yellow ones are negative lyapunov exponents.*

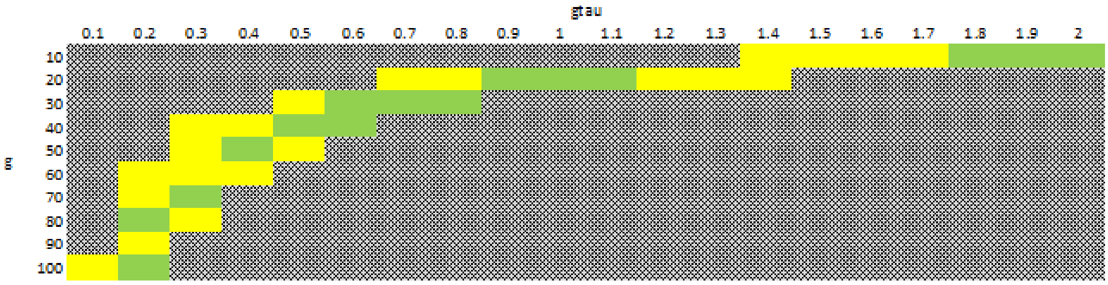


Figure 4.17: *Synchronisation for different combinations of parameters g and* $gtau$*. Green: occurrence of RMSEs* $\leq$ σ_{obs}*. Yellow: occurrence of synchronisation, with bursts of desynchronisation. Shaded gray: desynchronisation.*

matrix propagation synchronisation framework shown in chapter 3, the same unsuccessful result is found (not shown here).

For this $D_d = 1$ case to succeed, more observations are indeed needed, as we do not have the additional and auxiliary information brought from the time embeddings. RMSEs below noise levels are reached (figure 4.19) by increasing the number of observations to 50% of the system dimension. Additionally, to make this configuration successfully work in the ensemble setting, an increase of the ensemble size by a factor 2 was needed. The figure shows an example for the 100-variable system, now with an ensemble size of 10 members, so it can obtain synchronisation. This is the minimal number of members required, which is visible by the difficulty to stabilise and synchronise completely.

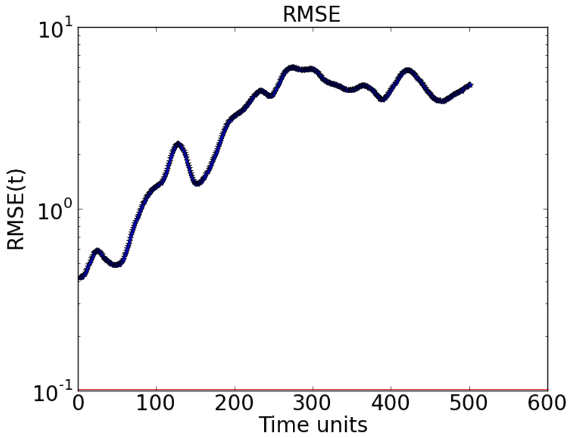


Figure 4.18: *Global Synchronisation Error (RMSE) for a 100-variable system with 25 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 1$). Ensemble Synchronisation was used with 5 members and localisation applied. The horizontal red line in the x-axis represents the standard deviation of the observation noise.*

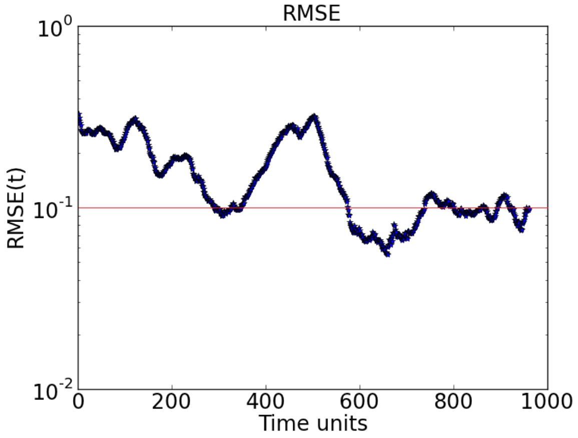


Figure 4.19: *Global Synchronisation Error (RMSE) for a 100-variable system with 50 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 1$). Ensemble Synchronisation was used with 10 members and localisation applied. The horizontal red line represents the standard deviation of the observation noise.*

When it comes to the relationship between a poor choice of time embedding interval and its sensitivity to observation noise, figures 4.20 and 4.21 show results similar to figures 4.5 and 4.6, but now for an inefficient delay dimension: $D_d = 2$. Figure 4.20 shows results using 5 ensemble members and figure 4.21 used 100 ensemble members. A comparison with figures 4.5 and 4.6 exposes a different impact of the magnitude of the measurement noise on the RMSEs. For 5 ensemble members, different observation noise strengths produced approximately comparable results, simply because synchronisation is not reached for any of the observation noise amplitudes tested. An increase on the size of the ensemble is required and a different sensitivity is found, with desynchronisation observed during most of the run period at a larger observation noise. Note that, even by using 100 members, the system struggles to stabilise and synchronise with an insufficient size of time embedding $D_d = 2$, even for smaller noise magnitudes, in comparison with the configuration in figure 4.6. Another interesting fact is that, despite the high RMSEs presented in figure 4.20, these values do not grow indefinitely, as the attractor occupies a limited area in the phase space and so the difference between x and x_{true} does not increase beyond a certain threshold.

In summary, a bad choice of the delay dimension generates a poor observability of the system. In this situation, if one desires to reach synchronisation, that will happen at the expense of increasing the number of available observations and/or increase the ensemble size of the system.

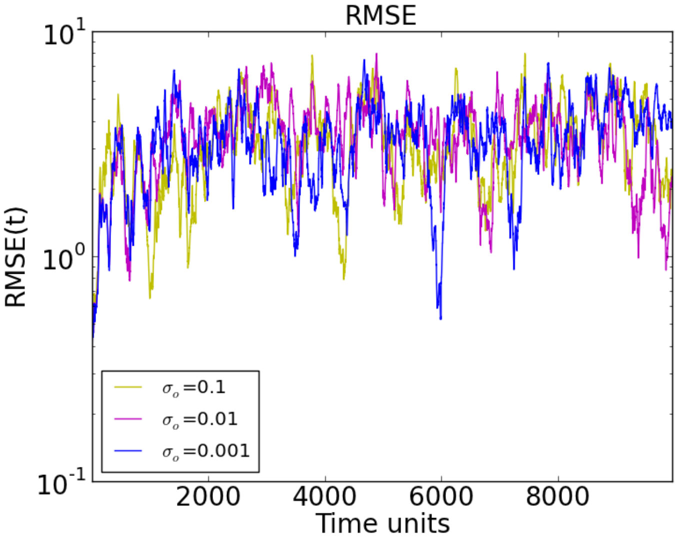


Figure 4.20: *Global Synchronisation Error (RMSE) for different standard deviations (σ) of observation noise in a 20-variable system with 5 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 2$). Ensemble Synchronisation was used with 5 ensemble members.*

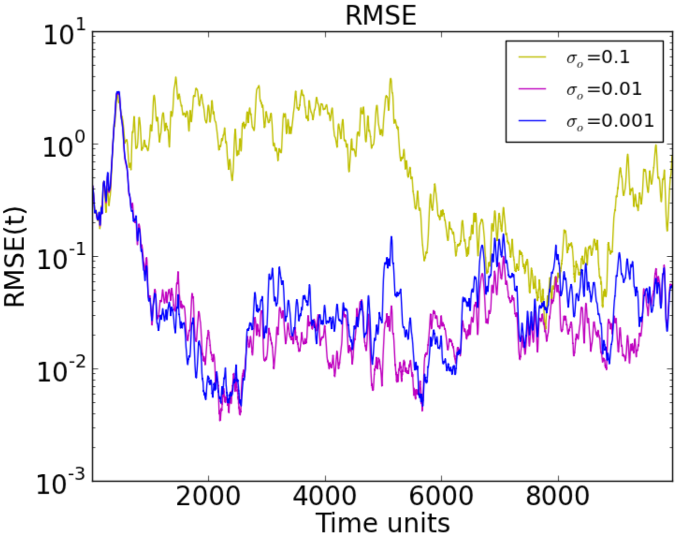


Figure 4.21: *Global Synchronisation Error (RMSE) for different standard deviations (σ) of observation noise in a 20-variable system with 5 measured variables sampled equidistantly on the Lorenz96 ring ($D_d = 2$). Ensemble Synchronisation was used with 100 ensemble members.*

Another interesting point to mention is the role of the variable τ in this system, as an indicator of the temporal frequency of observations needed for the achievement of accurate estimates and forecasts. A case in which $\tau = \Delta t$, e.g. means that all observations within a window will indeed be used. This window, however, needs to be large enough to catch additional information. If the window is too small, though, the observations will still be closely correlated and no gain is obtained, from a time embedding point of view.

To better understand the relationship between τ and the frequency of observations required for synchronisation to happen, an experimental simulation was performed, setting $\tau = \Delta t$ and the window length equivalent to $(D_d - 1)10\Delta t$. In this experimental configuration, the length of the observation window used is kept the same as in previous experiments, but now all the observations within it are used, instead of only using observations at every 10 time steps (configuration used in all previous experiments). Figure 4.22 shows a comparison between RMSEs resulted from using only the observations that appear at every 10 time steps (left) and using all the observations available in the window (right). Results are roughly equivalent and comparable, which suggests that τ can be an interesting indicator of the minimum observation frequency needed to achieve successful estimates/forecasts and so, synchronise the system. One could infer, e.g. that the present configuration, i.e. using observations at every 10 time steps seems to be enough to reach synchronisation. Observing the same variables more frequently would be useless, as it would not add much more information to this system. This way, τ could be considered an approximate lower bound for the observation frequency in a system.

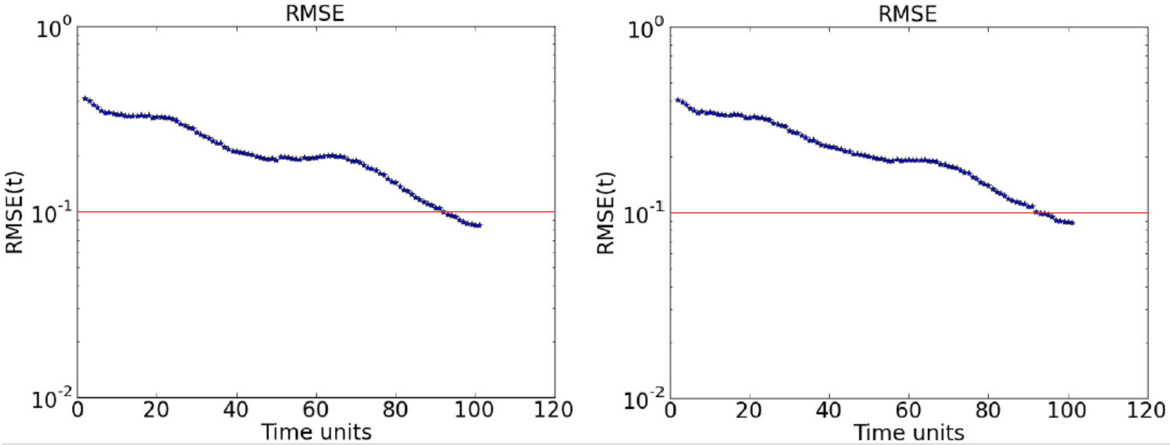


Figure 4.22: *Global Synchronisation Error (RMSE) for a 20-variable system with 5 measured variables sampled equidistantly on the Lorenz96 ring. Left: $\tau = 10\Delta t$ (configuration used along this thesis). Right: $\tau = \Delta t$. The window length in both cases is $(D_d - 1)10\Delta t$. The horizontal red lines represent the standard deviation of the observation noise.*

4.3 Summary

This chapter generalizes the time delay synchronisation technique proposed by Rey et al. (2014a) to an ensemble setting, a more suitable configuration for high dimensional systems. Numerical experiments with Lorenz96 model show comparable performance between both configurations in terms of the accuracy of state estimations, considering spatially sparse observations of about 25% of the model state. Different initial conditions and random-number realisations were tested in all the cases presented in this work, deriving qualitatively and quantitatively similar results.

This ensemble approximation is a huge step forward in synchronisation with time embedding, opening that methodology to a feasible framework for high-dimensional applications. This new scheme has already draw the attention from the physics field, for cardiac dynamics control, for instance.

While synchronisation is not a formal data assimilation methodology, as explained earlier, it provides deeper insights on how data assimilation works, and how one can, in principle, compensate the lack of observations via the time embedding concept. Results show that this also holds in the ensemble framework proposed, also opening a new way of viewing the reuse of observations in this kind of formulation.

It is important to mention the computational complexity of the two methods, the synchronisation through matrix propagation and the ensemble-based synchronisation. The number of model runs needed is strongly reduced in the ensemble-based method: at least by a factor of D_x/N_{ens}. Beyond that, the matrix method needs the linearised model equations, and these are typically more expensive than fully nonlinear model runs, as more terms appear in the equations.

Another positive feature of the innovative methodology proposed in this work is that it performs well for a desirable small number of ensemble members. We note that the initial values of the ensemble members are chosen randomly, in an isotropic distribution around the initial state, which could probably aid the method to work, as the particles are immediately attracted by the most expanding directions, so they represent, after a short transient, the unstable dynamics transversal to the synchronisation manifold, which is what is needed for a stabilisation method.

Table 4.1 shows the total RMSE means along the run periods, summarising some of the previous results from the experiments with 100 and 1000-variable systems, for the matrix propagation synchronisation (denoted as ”Synch” in the table) and the ensemble-based synchronisation (denoted as ”EnSynch” in the table), with and without localisation. It is noticeable that the synchronisation scheme with a matrix propagation produces better synchronised results, with lower RMSEs. However, it cannot be extended to high-dimensional systems, due to the need of propagation of a matrix with the size of the system, by the linearised model. The ensemble method overcomes this issue and still provides good synchronisation results, even for a high-dimensional system, when localisation is used. Note that inflation, another mechanism extensively used in some data assimilation methods and projected to deal with spurious correlations, is not needed in this type of framework. It should be mentioned that the scheme was only tested with a linear observation operator.

Table 4.1: Total RMSE means for 1,000-time step runs.

D_x	$Synch$	$EnSynch$ (no loc)	$EnSynch$ (loc)
100	0.02	0.05	0.05*
1000	0.02	0.09**	0.05

$*N_{ens}$ decreased from 15 to 5
$**D_d = 10$, in this case

An interesting remark is the relationship between the time embedding dimension and the number of observations at each observation time. As Rey et al. (2014a) have noticed, to stabilise all growing modes off the synchronisation manifold in a 20-dimensional Lorenz 96 system, one could use an 8-dimensional delay embedding of a single measurement, instead of (simultaneously) measuring 8 different state variables of the system. These conclusions were obtained for experiments with noiseless data. In the present work, it was found that with noisy observations, increasing the embedding dimension $D_e = D_d * D_y$ by just increasing D_d is not enough to achieve synchronisation, but the number of observations D_y must be chosen larger than 1. By adding measurement noise, it is not possible to synchronise a 20-dimensional system with only 1 observed variable, even with a small observation error, of the order of 0.001 and D_d as large as 20. The only way to synchronise this system is to increase the number of observations at each observation time. In all simulations presented here for $D_d = 5$, good results were obtained when $D_y = 0.25D_x$. By increasing the number of available observations to $D_y = 0.5D_x$, it was possible to synchronise the system with a delay dimension $D_d = 2$ (results not shown here), confirming this close relationship between D_d and D_y. Besides that, the ensemble framework has an additional parameter to be considered: the ensemble size. As shown in figures 4.20 and 4.21, the size of the ensemble also plays a role in the stabilisation of the unstable modes. Therefore, more research is needed on this intriguing relation between the time embedding dimension, the number of observations and the ensemble size, to better characterise the observational requirements to reach a synchronised system.

Additionally, results suggest that the embedding dimension D_e needs to be of the order of the system dimension, so one can obtain RMSE values below the measurement noise. Again, further research is needed to confirm this remark. Note that all results presented here show global RMSEs, i.e. RMSE values of the whole system, including observed and unobserved variables. Finally, an interesting investigation that is not included in this thesis would be to test the influence of a non-uniform observation network and also the use of different localisation radii.

Back to the data assimilation context, results shown in this chapter suggest that the ensemble-based synchronisation scheme is a valuable tool to steer model states to the true evolution of the system. However, although good estimates and predictions of the states are obtained, the method provides no uncertainty estimates. Therefore, this methodology should not be used as a stand-alone data assimilation method, also due to the fact that observations within the window $[j, j + (D_d - 1)\tau]$ are used more than once, which would lead to complexities in a conventional data assimilation methodology.

Both issues, however, are solved simultaneously by inserting the ensemble synchronisation in a more comprehensive data assimilation method, like a particle filter. The following chapters will present an investigation on the usefulness of the ensemble synchronisation scheme as (part of) a proposal density in a fully nonlinear particle filter. This freedom of manipulating the likelihood as part of a proposal follows previous work from, e.g. Papadakis et al. (2010) and van Leeuwen (2009), where it is discussed how an Ensemble Kalman Filter (EnKF) can be used with smaller observation errors or even multiple times, as it is only part of a proposal. van Leeuwen (2010) has introduced a simple relaxation term to future observations in his particle filter in order to steer the particles towards the high-probability region of the posterior between observations. This methodology was applied to a high-dimensional system in Ades and van Leeuwen (2014), and also in a climate model, by Browne and van Leeuwen (2015). The latter study concluded that the relaxation term used was not working satisfactorily, as the model drifted too far from the truth between observations. Browne (2016) found a similar problem, both with the simple relaxation term and with an Ensemble Kalman Smoother as a proposal between observations. Investigating the application of the ensemble synchronisation in a particle filter is the main motivation of this thesis. The results of this interesting combination will be exposed and discussed in the following chapters.

Chapter 5:

SYNCHRONISATION AND THE PARTICLE FILTER IN A LOW DIMENSIONAL SYSTEM

5.1 Introduction

The previous chapter described a new scheme developed, the ensemble-based synchronisation. It is an ensemble-based time embedding method, similar to an ensemble smoother or a 4DEnsVar, which avoids the need for tangent-linear models and adjoint calculations. Aiming to improve the nonlinear data assimilation, the main idea of this thesis is to merge the efficiency of synchronisation described previously with the data assimilation formalism. For this, synchronisation will be used indirectly, as a proposal density, in a nonlinear data assimilation method: a particle filter.

The filter chosen for our experiments is the Implicit Equal-Weights Particle Filter - **IEWPF** (Zhu et al., 2016). As described before, in chapter 2, this filter does not suffer from degeneracy by construction.

Regarding the proposal density freedom, it is well-known that several methods can be used for this purpose, including traditional methods like 4D-Var and ensemble smoothers. However, depending on the method chosen, performing this procedure can become rather expensive, as it usually involves solving a problem similar to a 4D-Var on each particle. Thus, typically, simpler schemes are employed between observation times. These schemes are expected to be less efficient, although we can ensure that Bayes theorem is fulfilled exactly for each particle.

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 (the same used in Ades and van Leeuwen (2012)). The authors concluded 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 of the method. That is the main goal of this thesis. In addition, the experiments shown here aim to test an environment with a higher complexity compared to the work performed by Zhu et al. (2016).

5.2 Formulation

The first point to be noted is that before it was implemented in the particle filter, the ensemble synchronisation scheme described in chapter 4 was adapted, to reduce its computational cost and also to adjust it to the particle filtering framework, as follows:

In order to avoid the propagation of the ensemble for $(D_d - 1)$ times at every time step, the initial set of ensemble perturbation matrices $\boldsymbol{X}$ computed at time step $j = 0$ is stored. This way, we have:

$$\boldsymbol{X} = (\boldsymbol{X}^{0}, \boldsymbol{X}^{\tau}, ..., \boldsymbol{X}^{(D_d - 1)\tau}) \tag{5.1}$$

Remember that these perturbation matrices are obtained to construct the Jacobian $\frac{\partial \mathbf{S}(\mathbf{x}^j)}{\partial \mathbf{x}^j}$. Additionally, the ensemble means are also stored at each of the time embedding intervals described in equation 5.1.

At the next time steps in which observations exist, these procedures are followed:

1) rescale all perturbation matrices in $\boldsymbol{X}$ with a factor:

$$\gamma(k) = \frac{\boldsymbol{\sigma}(k)_{prior}}{\boldsymbol{\sigma}(k)_{new}} \tag{5.2}$$

where k refers to the variables and

$$\boldsymbol{\sigma}(k) = \sqrt{\frac{1}{N_{ens} - 1} \sum_{i=1}^{N_{ens}} (x(k)_i - \bar{x}(k))^2} \tag{5.3}$$

is the formulation to compute the spreads. The $\boldsymbol{\sigma}_{prior}$, then, is the spread of the perturbation matrix for the actual time step propagated from the previous observation time, i.e the prior, whereas $\boldsymbol{\sigma}_{new}$ is the spread of the perturbation matrix newly computed at the actual time step.

2) recenter the ensembles at each time embedding interval, by adding to the means previously stored the difference between the mean of the new ensemble and the prior ensemble.

3) propagate for an additional τ time steps the last rescaled and recentered ensemble, i.e the one at $j + (D_d - 1)\tau$ that was computed at the previous observation time step. This way, we now have a complete $\boldsymbol{X}$ at this actual observed time step.

After performing these steps and considering an observation period of f (in which, for example, having available observations at every other time step corresponds to $f = 2$), two situations can occur:

- $f = \tau$: in this case we just follow the procedures described above.
- $f \neq \tau$: after rescaling and recentering the perturbation matrices, we propagate each component of $\boldsymbol{X}$ for only f time steps, to obtain a new $\boldsymbol{X}$. Perform procedure 3) mentioned above at the end.

(Note that we can have e.g. $f = 2$, meaning that observations occur at every other time step and use a constant time interval $\tau = 10\Delta t$.)

After having the rescaling and recentering steps performed, one can start the implementation of the ensemble-based synchronisation scheme as a proposal density in the IEWPF.

Consider time steps j and $j+1$ between observation time steps n. We will assume here that the model errors and the proposal are Gaussian. Therefore, our proposal density formulation is the following:

The transition density $p(x^{j+1} \mid x^j)$ for the prior is related to the original model via:

$$x_i^{j+1} = f(x_i^j) + \boldsymbol{\beta}_i^{j+1} \tag{5.4}$$

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

$$p(x^{j+1} \mid x_i^j) \sim N(f(x_i^j), \mathbf{Q}) \tag{5.5}$$

This probability can be computed by the following:

$$p(x^{j+1} \mid x_i^j) \propto exp\left[-\frac{1}{2}\left(x_i^{j+1} - f(x_i^j)\right)^T \mathbf{Q}^{-1}\left(x_i^{j+1} - f(x_i^j)\right)\right] \tag{5.6}$$

The proposal density $q(x^{j+1} \mid x_i^j, y^n)$ is related to a proposed model, in our case, the Ensynch equation:

$$x_i^{j+1} = f(x_i^j) + g\left(\frac{\partial S(x^j)}{\partial x^j}\right)_i^\dagger (Y^j - S(x^j)) + \hat{\boldsymbol{\beta}}_i^{j+1} \tag{5.7}$$

where $\hat{\boldsymbol{\beta}}_i^{j+1}$ in our case is drawn from the same distribution as $\boldsymbol{\beta}_i^{j+1}$, so we assume here that $\hat{\boldsymbol{Q}} = \mathbf{Q}$. The augmented observation and state vectors Y^j and $S(x^j)$ use future observations in the time-embedding window, described in equations (3.13) and (3.12), respectively.

Hence,

$$q(x^{j+1} \mid x_i^j, Y^j) \sim N(x_{synch}, \boldsymbol{Q}) \tag{5.8}$$

where $x_{synch} = f(x_i^j) + g\left(\frac{\partial S(x^j)}{\partial x^j}\right)_i^\dagger (Y^j - S(x^j))$, the deterministic part of the right-hand side of equation (5.7).

The distribution of this proposal density is computed as:

$$q(x^{j+1} \mid x_i^j, Y^j) \propto exp\left[-\frac{1}{2}\left(x_i^{j+1} - x_{synch}\right)^T \boldsymbol{Q}^{-1}\left(x_i^{j+1} - x_{synch}\right)\right] \tag{5.9}$$

which we can relate to the model error added in equation (5.7) as:

$$q(x^{j+1} \mid x_i^j, Y^j) = exp\left[-\frac{1}{2}(\hat{\boldsymbol{\beta}}_i^{j+1})^T \boldsymbol{Q}^{-1}\hat{\boldsymbol{\beta}}_i^{j+1}\right] \tag{5.10}$$

We avoid computing $\mathbf{Q}^{-1}$ in equation (5.10), as in Ades and van Leeuwen (2014), by noting that the error $\hat{\boldsymbol{\beta}}_i^{j+1}$ is sampled from $N(0, \mathbf{Q})$ through $\hat{\boldsymbol{\beta}}_i^{j+1} = \mathbf{Q}^{1/2}\hat{\boldsymbol{\eta}}_i^{j+1}$, where $\hat{\boldsymbol{\eta}}_i^{j+1} \sim N(0, I)$. This leads to:

$$(\hat{\boldsymbol{\beta}}_i^{j+1})^T \boldsymbol{Q}^{-1}\hat{\boldsymbol{\beta}}_i^{j+1} = (\hat{\boldsymbol{\eta}}_i^{j+1})^T \hat{\boldsymbol{\eta}}_i^{j+1} \tag{5.11}$$

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

$$w_i = \frac{p(x^{j+1} \mid x^j)}{q(x^{j+1} \mid x^j, Y^j)} \tag{5.12}$$

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

$$w_i^{n-1} = \prod_{j=1}^{n-1} \frac{p(x_i^{j+1} \mid x_i^j)}{q(x_i^{j+1} \mid x_i^j, Y^j)} \tag{5.13}$$

This way, the ensemble synchronisation will pull the particles towards the observations, between observation time steps.

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.

This is executed by drawing samples implicitly from a Gaussian distributed proposal $q(\boldsymbol{\xi})$ instead of the original $q(x^{j+1} \mid x^j, Y^j)$. Note that x_i^a in equation 2.81 is the mode of $p(x^n \mid x_i^{n-1}, y^n)$.

$\boldsymbol{\xi}_i$ is then drawn from $q(\boldsymbol{\xi})$, leading to:

$$q(x_i^n \mid x_i^{n-1}, Y^j) = q(\boldsymbol{\xi}_i)\left\|\frac{\partial \boldsymbol{\xi}}{\partial x^n}\right\| \tag{5.14}$$

The particle weights are then computed according to equation 2.84.

5.3 Implementation

Regarding the implementation of the ensemble synchronisation 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, the 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 that one wants to test, while minimal changes in the model are needed. For full instructions and documentation on this platform, see: http://www.met.reading.ac.uk/ darc/empire/index.php.

For this work, a list of steps was needed to implement the system into EMPIRE:

1. convert the pure synchronisation code which was coded in Python to Fortran (EMPIRE's main programming language);
2. add some of the synchronisation parameters, new to the existing ones in EMPIRE; and
3. code a group of new subroutines and modules to configure the new parameters and also to allow the correct communication between the new data assimilation code and the model.

For the experiments so far, the system was run on a compute cluster and for the combined system presented in this chapter, both the ensemble synchronisation and the particle filter codes were written in Fortran90.

Appendix C contains a description on how to code the IEWPF using the ensemble-based synchronisation scheme as a proposal density.

5.4 Experiments and results in the Lorenz96 model

At this point and before showing any experiment results, it is useful to emphasize to the reader the importance of data assimilation in a nonlinear system. Figure 5.1 aims to highlight

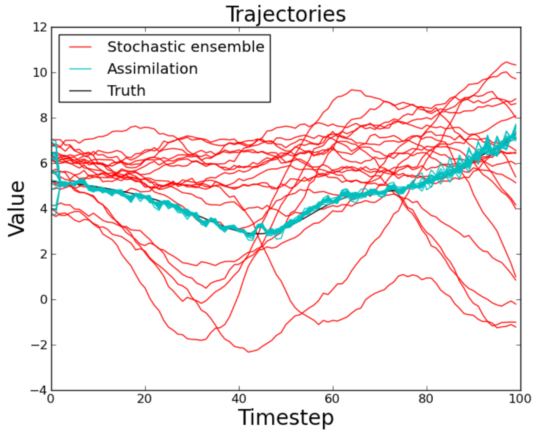


Figure 5.1: *Trajectories of the ensemble members with data assimilation in blue and in a stochastic run in red. The truth is in black.*

this. It is a simple run showing trajectories of the ensemble members of a particle filter with observations at every 2 time steps (in blue) and trajectories of a stochastic run (in red). It is clear how the filter brings trajectories towards the truth and keep them around it, with the aid of the observations. In a nonlinear stochastic run, an ensemble of trajectories can follow any path away from the truth.

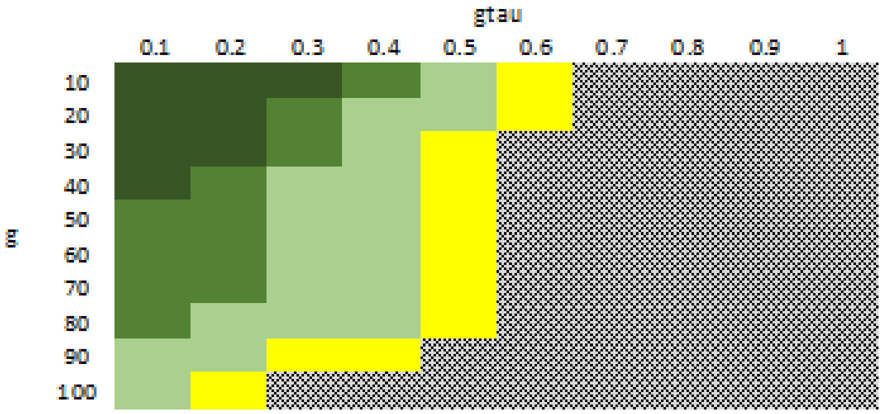
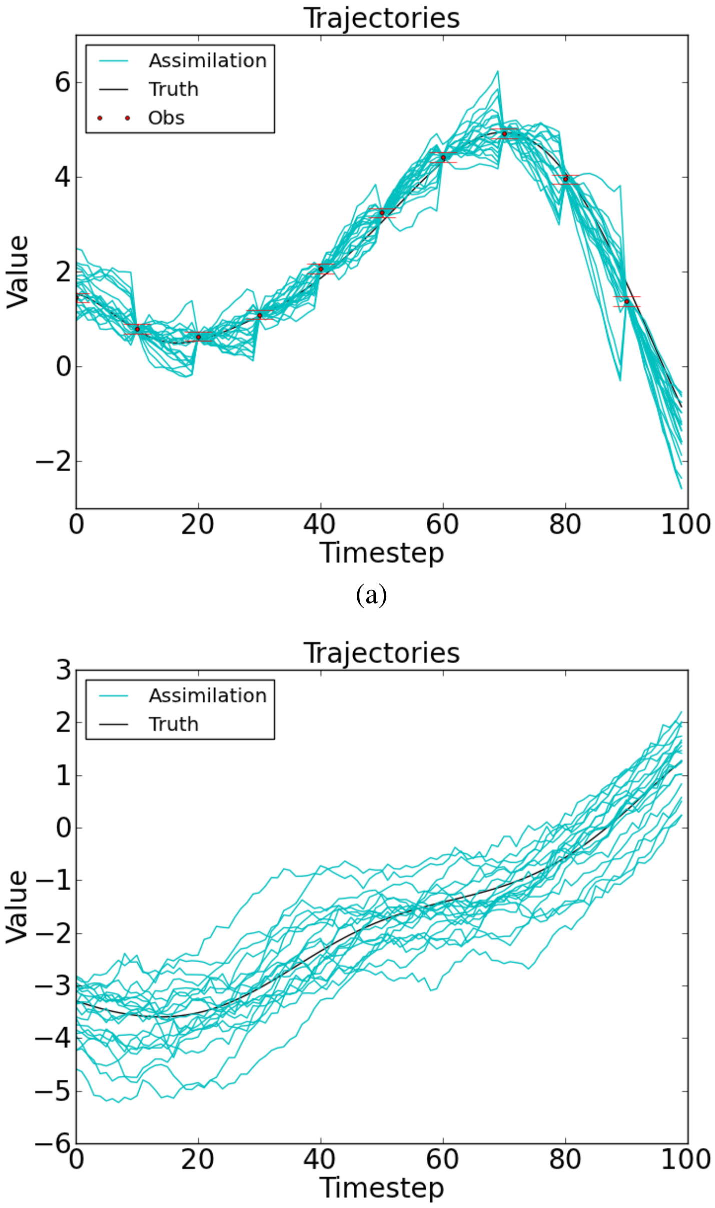


Figure 5.2: *Magnitude of Global RMSE for different combinations of parameters g and $gtau$ in the particle filter. The darkest green zones represent the combinations which lead to the lowest RMSEs. Yellow: higher RMSEs. Shaded gray: system overflow.*

Back to the experiments proposed in this thesis, the same configuration used for the previous ones in chapter 4 is basically kept. This chapter is focused to perform tests in the most complex environment presented in chapter 4, thus tests now are only performed in a 1000-variable system. The same proportion of it is observed (25%), i.e. every 4th variable is measured in the Lorenz ring. Also, the same standard deviation for the observation error is used: $\sigma_{obs} = 0.1$.

One first important point to note is that the stand-alone ensemble-based synchronisation scheme shown in the previous chapter deals with a leading trajectory, and the main goal is to keep its convergence towards the attractor. An ensemble is indeed generated in that case, mainly to make feasible a cheaper construction of the Jacobian. When we insert this scheme into a particle filter with its ensemble of trajectories, it means that the synchronisation coupling term has now different trajectories to work at, which does not necessarily mean that it will reach its optimal state as shown previously. Actually, we observe that in this case, the coupling term has now an impact on the majority of the ensemble members, but maybe not in all of them, as a few others may not feel much of its influence. This way, some of the experiment characteristics described in the previous chapter will vary for this more complex case.

Therefore, in this combination between the particle filter and the ensemble synchronisation, a noticeable difference is the use of the coupling constant g and the strength factor $gtau$. It is possible to note in figure 5.2 that, compared to figure 4.17, the optimal combinations now lie on the upper left edge of the graph (note the different scales between both figures). Indeed, in the particle filter, the coupling term which is generated from the mean of an evolving ensemble may have less strength, to guarantee the best global RMSEs. For stronger couplings, the system crashes.



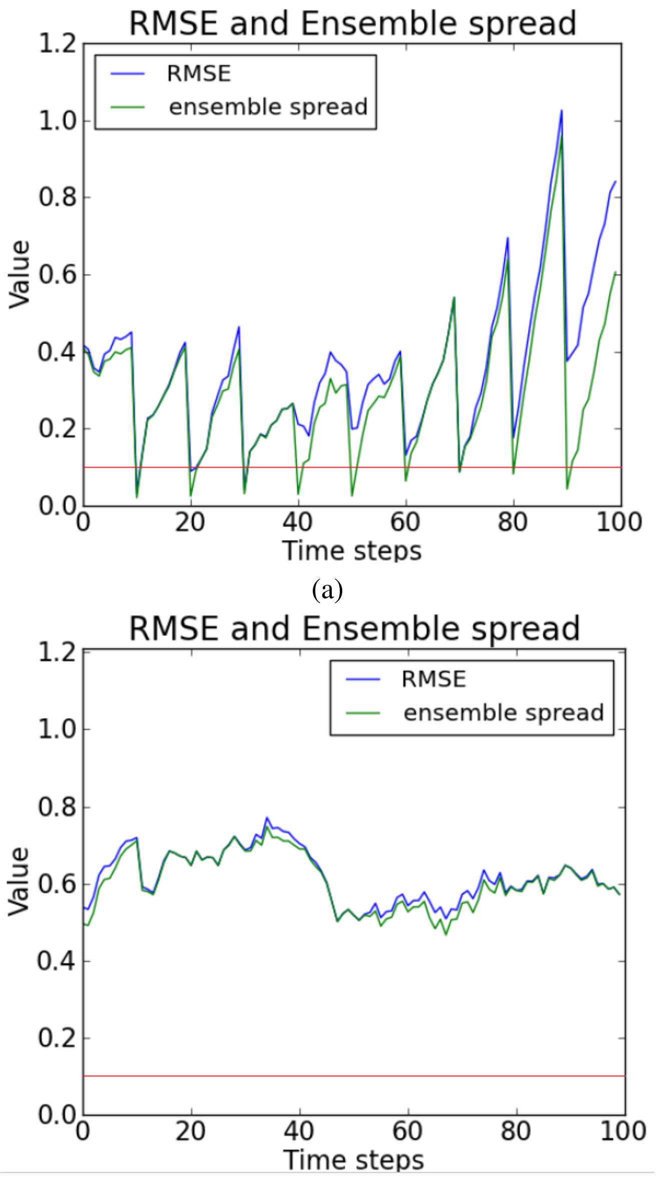
(a)

(b)

Figure 5.3: *Trajectories of the ensemble members for an observed variable (a) and an unobserved variable (b). Observations occur at every 10 time steps. ($Dx = 1000$, $Dy = 250$ and $Nens = 20$).*

In the experiments presented here, a coupling constant $g = 10$ and a strength factor $gtau = 0.2$ were used, although any other combination shown in figure 5.2 in the green zones does yield good results. Additionally, in these experiments a delay dimension $D_d = 6$ was used and further interesting discussions will appear later in this chapter about the relationship between this parameter and the number of particles in the ensemble. In these experiments, the number of ensemble members is $N_{ens} = 20$, each of them being perturbed by a random vector drawn from $N \sim (0, 0.5)$. The localisation radius of influence is the same which was used in the previous chapter, i.e. rad = 10 and a tridiagonal matrix **Q** is used, where the values of the main diagonal and the sub- and superdiagonal are:

$$\mathbf{Q} = \begin{pmatrix} 10^{-2} & 2.5 \times 10^{-3} & 0 & \dots & \dots & 0 \\ 2.5 \times 10^{-3} & 10^{-2} & 2.5 \times 10^{-3} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & \dots & 0 & 2.5 \times 10^{-3} & 10^{-2} & \end{pmatrix} \tag{5.15}$$



(a)

(b)

Figure 5.4: *RMSE and the ensemble spread for an observed variable (a) and an unobserved variable (b). Observations occur at every 10 time steps ($Dx = 1000$, $Dy = 250$ and $Nens = 20$). The horizontal red lines represent the standard deviation of the observation noise.*

Also, a percentage of 50$\%$ of positive ε_i in the IEWPF is used, i.e. ε_i will be chosen randomly to be negative or positive, with the same probability.

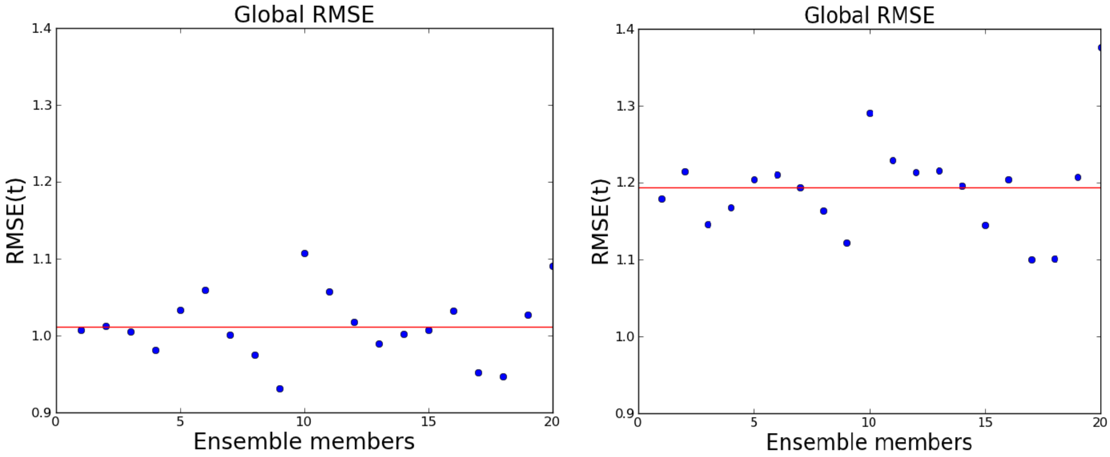


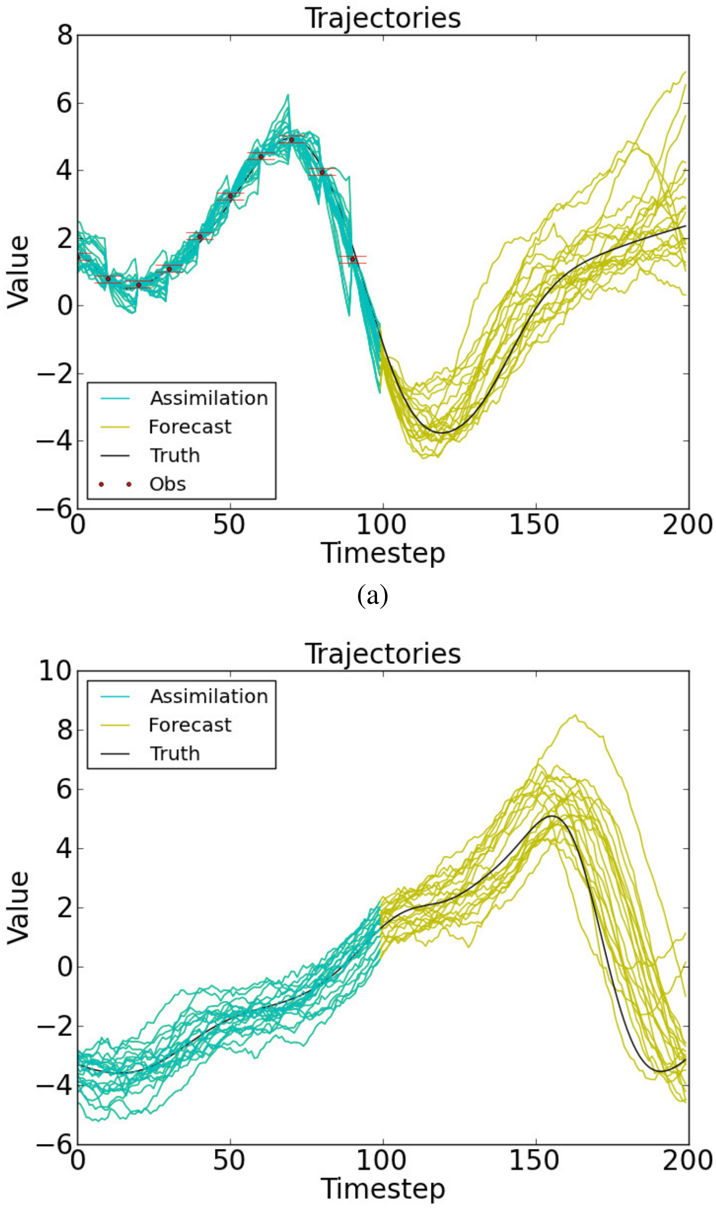
Figure 5.5: *Global RMSEs averaged over the estimation period (100 time steps) for each ensemble member for the IEWPF with different relaxation terms: EnSynch as a proposal (left) and proposal used in Zhu et al. (2016) (right). The red line is the ensemble mean.*

Different observation periods were tested, with $f = 2$, $f = 5$ and $f = 10$, but this chapter will show results only for the last more challenging case. To update the variables at unobserved time steps, the same methodology described in the previous chapter was used: the coupling term is progressively increased by $n = 1, 2, ...$(observation interval -1) in $g * (n * gtau)\frac{\partial S(x^j)}{\partial x^j}^{\dagger}(Y^j - S(x^j))$. This procedure makes the scheme less computationally intense, as the effort to find the coupling term is performed less frequently.

Results for the trajectories of the particles for a period of 100 time steps are shown in figure 5.3, both for an observed and an unobserved variable. The observed variable has its trajectory well tracked by the ensemble of particles between the observations, thanks to the ensemble synchronisation. It is possible to note that, between $t = 80$ and $t = 90$, four of the members did not follow the main trajectory of the rest of the ensemble, as the latter keeps tracking the truth. This is probably related to the different effectiveness of the coupling term in each member, as mentioned before. The main job of synchronisation, though, is done for the unobserved variable. The Jacobian spreads information from the observed to the unobserved variables, leading to an ensemble that nicely follows the true unobserved state.

Figure 5.4 shows the RMSEs and the ensemble spread for the same observed and unobserved variables. Low values for the observed variable are obtained and the ensemble spread is consistent with the behaviour of those errors. Results for the unobserved variable also follow this behaviour, showing low RMSEs and a good characterisation of the uncertainty with the ensemble spread.

Figure 5.5 shows the global RMSEs for each particle, averaged over the whole estimation period of 100 time steps for the IEWPF using different relaxation terms: the one presented in

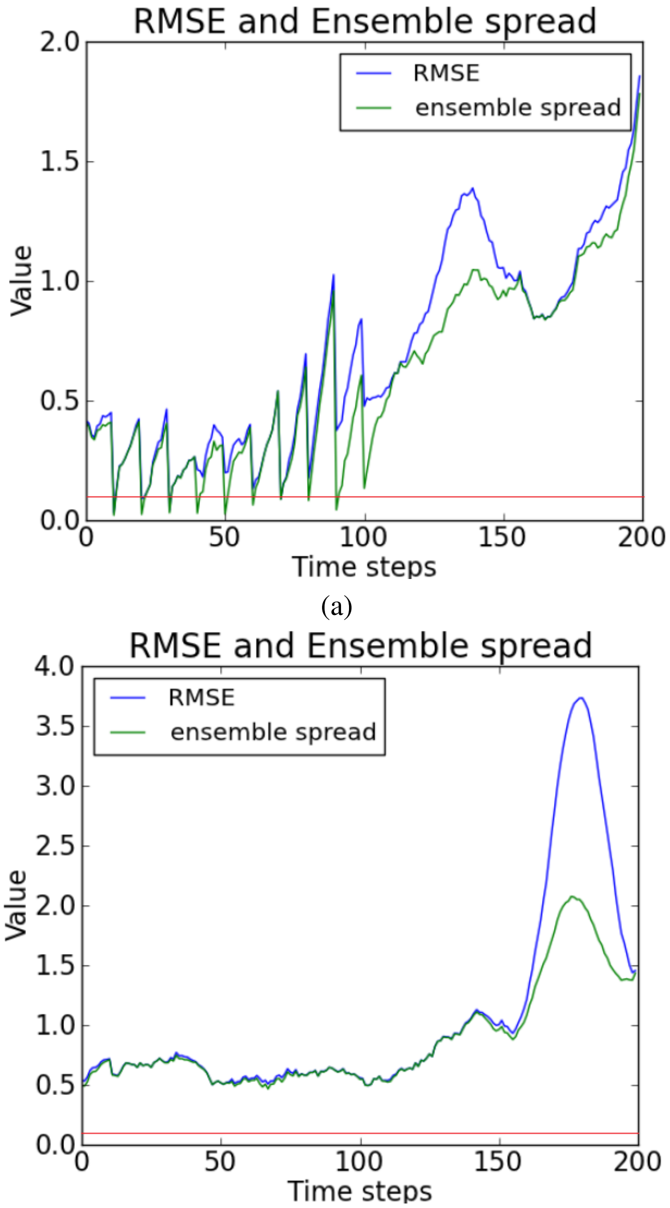


(a)

(b)

Figure 5.6: *Trajectories of the ensemble members for the assimilation and prediction (after t=100) periods for an observed variable (a) and unobserved variable (b). Observations occur at every 10 time steps (*$Dx = 1000$*,* $Dy = 250$ *and* $Nens = 20$*).*

this work, with the ensemble synchronisation as a proposal and the proposal density used in Zhu et al. (2016) (details in Section 2.4.4). The red line represents the mean of all global RMSEs. Note that these are RMSEs for all observed and unobserved variables. The framework using the ensemble-based synchronisation as a relaxation term generates RMSEs 15% lower than the relaxation term used in the previous work on the IEWPF. It is interesting to mention that in all experiments performed, testing different run periods, amount and period of observations, the use of EnSynch as a proposal density have generated lower RMSEs in the IEWPF scheme, if compared with the relaxation term used by Zhu et al. (2016).



(a)

(b)

Figure 5.7: *RMSE and the ensemble spread for the assimilation and prediction (after t=100) periods for an observed variable (a) and unobserved variable (b). Observations occur at every 10 time steps. ($Dx = 1000$, $Dy = 250$ and $Nens = 20$). The horizontal red lines represent the standard deviation of the observation noise.*

Let us now consider a forecast period of 100 time steps after the estimation period, in which the data assimilation module is turned off and the system runs freely. Figure 5.6 shows how the particles keep following the right path along the truth, even without counting on the observation information. Additionally, the unobserved trajectories do follow the pattern of the truth during the forecast period. The RMSEs shown in figure 5.7 naturally increase, considering it is a forecast. Also note that the ensemble spreads keep representing the uncertainty in a consistent way.

Figure 5.8 is a panel with trajectories of particles for 6 unobserved variables during the estimation period, located in different positions on the Lorenz ring and different distances from

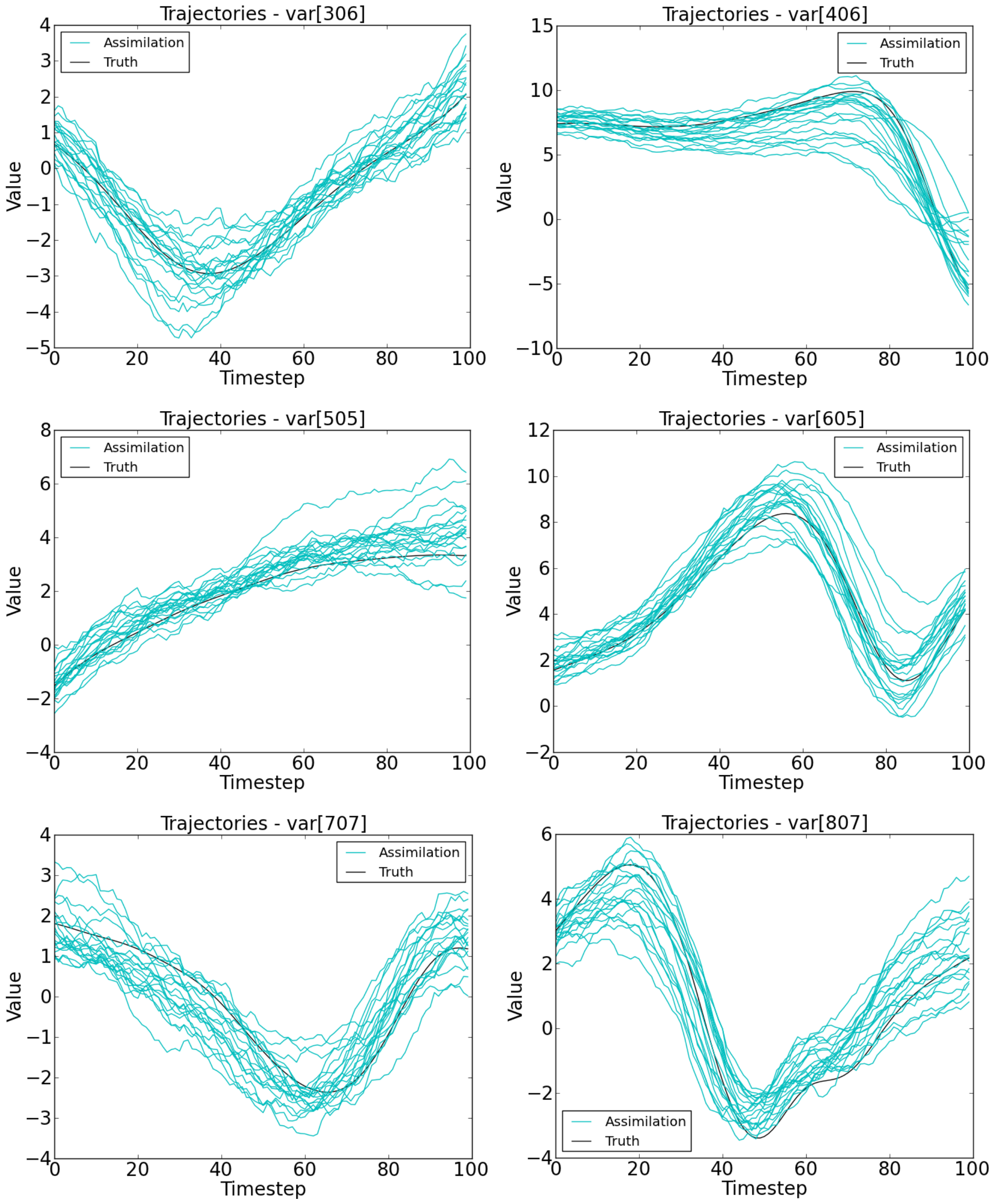


Figure 5.8: *Trajectories of the ensemble members for 6 unobserved variables located in different positions on the Lorenz ring (grid points 306, 406, 505, 605, 707 and 807). (Dx = 1000, Dy = 250 and $Nens$ = 20).*

the closest observations. They all show a good tracking of the truth, regardless the position of the variable in state space.

For a longer run of 1,000 time steps, estimations start to degrade, but trajectories keep following the behaviour of the truth, as can be seen in figure 5.9. The figure shows the mean of the ensemble of trajectories and the standard deviation in the shaded gray area for an observed and unobserved variable. The overall mean trajectories still fairly follow the truth along the period, specially the unobserved variable, which keeps a reasonable spread around the true state.

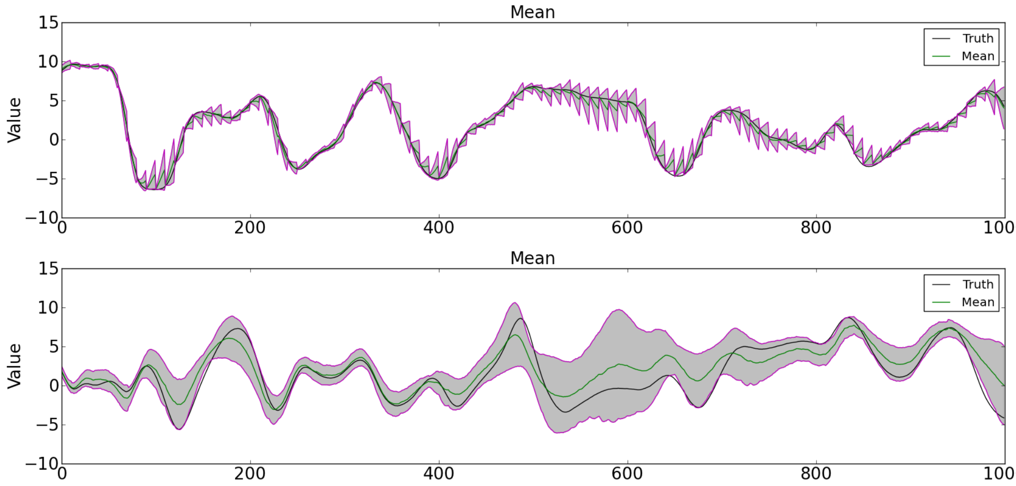


Figure 5.9: *Ensemble means for an observed (top) and unobserved (bottom) variables. The standard deviation is delimited by the gray shaded area.*

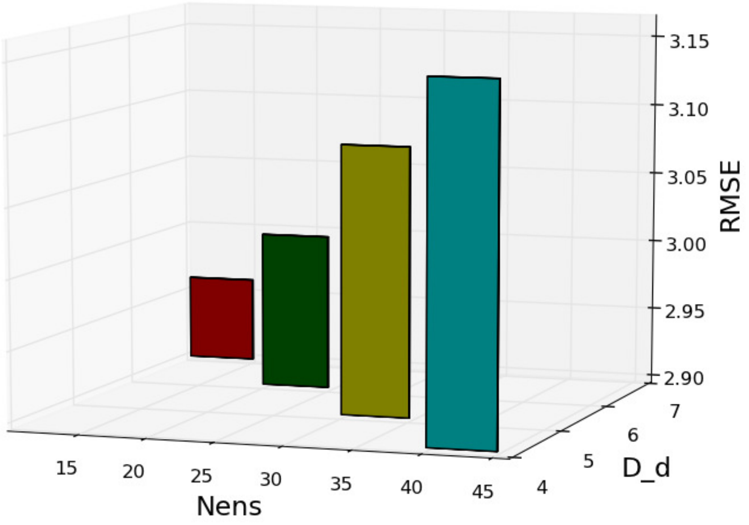


Figure 5.10: *Relationship between the number of ensemble members (N_{ens}), the size of the delay dimension (D_d) and the RMSEs produced (same constants g and $gtau$ were used).*

Another interesting point to be mentioned is shown in figure 5.10, in which results from one of the experiments is plotted (the same behavior applied to different initial conditions tested). While performing the experiments it was possible to observe that there is an inverse relationship between the number of ensemble members and the size of the delay dimension. The smaller the number of particles, the larger the time embedding to be used. We also note that we actually obtain lower RMSEs while using a smaller number of members in this system. Apparently, as we increase the number of particles, we also increase the chances of having more members misusing the information from the coupling term and derailing from the truth, generating more of those peaks as described before. Another possible prompter for this behaviour could be the IEWPF step used in the scheme. This, of course, needs further investigation and may yield interesting clues in the future on how these combined parameters actually work in this system.

5.5 Summary

This chapter presented results obtained in a nonlinear system, using a non-degenerate particle filter, the IEWPF, with a proposal density between observations composed of the ensemble-based synchronisation scheme. The main idea was to apply the coupling term derived from the ensemble synchronisation 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.

Results show reasonable RMSE errors for observed and unobserved variables and also that the ensemble spread in the system is consistent to the RMSEs found. Additionally, good forecasts can be obtained from this system, both for observed and unobserved variables, with a consistent ensemble spread. Although the uncertainties slightly increase for longer runs, the behavior of the true states are still followed by the ensemble trajectories. It should be noted that the scheme was only tested with a linear observation operator, Gaussian observation error and Gaussian errors in the model equations.

Looking deeper into this implementation, the analysis that can be made is that, although the method gives qualitatively reasonable results, the combined system still does not use the full potential of the ensemble synchronisation scheme in the proposal, as further work on understanding the efficiency of the synchronisation coupling term in different trajectories of the particles is still needed. One factor in this is that the synchronisation process is ”interrupted” at every 10 time steps to equalise the weights of the particles through the IEWPF part of the filter. This may also be a reason why the RMSEs in the particle filter do not reach the same order of magnitude shown in figure 4.14. While performing experiments where a larger part of the state space was observed, interesting relations between the time delay and the coupling constant g could be observed. For instance, for a system with $1/3$ of the variables observed, it was possible to reduce the size of the time embedding by just increasing 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. This way, many improvements can be made to the system, and these are the first results combining synchronisation and a particle filter to tackle this complex, nonlinear problem.

Chapter 6:

SYNCHRONISATION AND THE PARTICLE FILTER IN A HIGH DIMENSIONAL SYSTEM

6.1 Introduction

The previous chapter presented results for the IEWPF using as a relaxation the coupling term obtained from the ensemble synchronisation scheme for the nonlinear model Lorenz96. While preventing the use of tangent-linear models and adjoints, results have shown that synchronisation can be a useful proposal density, capable of pulling the particles around the true state, yielding converging trajectories.

The idea now is to investigate how does this scheme behave in a bigger, more realistic nonlinear model, such as the barotropic vorticity model.

In this chapter, another interesting discussion will be presented: two different non-degenerate particle filters will be compared - the IEWPF and the EWPF - and their advantages and issues will be exposed, while using synchronisation as their proposal density. This is the first comparison ever made between these two specific particle filters.

The formulation used to implement the ensemble synchronisation as a proposal density is the same for both particle filters and was already described in detail in section 5.2.

6.2 Barotropic vorticity model

In order to investigate further the applicability of synchronisation in a particle filter, twin experiments are now performed using the barotropic vorticity model. It is a fluid dynamics model representing a barotropic flow on a torus and it was chosen, as it is relatively simple to run and can show a very chaotic behavior. This section will present the derivation of the barotropic vorticity equation and introduce its non-dimensional variables. More details about the model and the twin experiments configuration will also be described.

6.2.1 The barotropic vorticity equation

The barotropic vorticity equation considers the assumption of a nearly barotropic atmosphere, i.e. there is no vertical wind shear of the geostrophic wind. This equation is derived from the Navier-Stokes momentum equation:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}.\nabla \boldsymbol{u} + \frac{1}{\rho}\nabla p + g = \nu \nabla^2 \boldsymbol{u} \tag{6.1}$$

where $\boldsymbol{u}$ is a vector representing the flow velocity, ρ is the density, p is the pressure, g represents a body acceleration such as gravity and ν is called the kinematic viscosity. In this equation, the second term represents advection, the third and fourth terms represent internal and external forces or momentum sources, respectively, and in the right-hand side of the equation there is the diffusion term.

Considering that the vorticity q is given by:

$$q = \nabla \times \boldsymbol{u} \tag{6.2}$$

and that the advection term can be rewritten as:

$$\boldsymbol{u}.\nabla \boldsymbol{u} = \nabla\left(\frac{\boldsymbol{u}.\boldsymbol{u}}{2}\right) + \boldsymbol{u} \times (\nabla \times \boldsymbol{u}) = \nabla\left(\frac{\boldsymbol{u}.\boldsymbol{u}}{2}\right) + \boldsymbol{u} \times q \tag{6.3}$$

then, equation 6.1 becomes:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla\left(\frac{\boldsymbol{u}.\boldsymbol{u}}{2}\right) + \boldsymbol{u} \times q + \frac{1}{\rho}\nabla p + g = \nu \nabla^2 \boldsymbol{u} \tag{6.4}$$

The vorticity equation is obtained by taking the curl of equation 6.4. As we know, the curl of a gradient is equal to zero, cancelling the second term in that equation. Also, the curl of a constant is zero and so the gravity term disappears from this derivation. If we also assume that viscosity is negligible compared to the other terms in this system and take the following identity into consideration:

$$\nabla \times (\boldsymbol{u} \times q) = q\,(\nabla.\boldsymbol{u}) - (q.\nabla)\,\boldsymbol{u} - \boldsymbol{u}\,(\nabla.q) + (\boldsymbol{u}.\nabla)\,q \tag{6.5}$$

we obtain the following:

$$\frac{\partial q}{\partial t} + q\left(\nabla.\boldsymbol{u}\right) - \left(q.\nabla\right)\boldsymbol{u} - \boldsymbol{u}\left(\nabla.q\right) + \left(\boldsymbol{u}.\nabla\right)q + \nabla \times \left(\frac{1}{\rho}\nabla p\right) = 0 \tag{6.6}$$

To obtain the barotropic vorticity equation, some further assumptions are made:

1. the fluid is considered incompressible, i.e $\nabla.\boldsymbol{u} = 0$
2. the equation is considered to be solved in a two-dimensional field, i.e. horizontal flows do not vary vertically and no vertical flows are expected either. This means that $(q.\nabla)\boldsymbol{u} = 0$
3. considering the identity $\nabla.(\nabla \times A) = 0$, where A is a vector, we observe that $\nabla.q = 0$
4. taking into account another vector identity which states that $\nabla \times \left(\frac{1}{\rho}\nabla p\right) = \nabla\left(\frac{1}{\rho}\right) \times \nabla p = -\left(\frac{1}{\rho^2}\right)\nabla\rho \times \nabla p$, and considering that the flow is barotropic, which means that ρ and p surfaces are parallel to each other, we can also find that $\nabla\rho \times \nabla p = 0$.

This way, we end up with the following formulation for the barotropic vorticity equation:

$$\frac{\mathrm{D}q}{\mathrm{D}t} = \frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} + v\frac{\partial q}{\partial y} = 0 \tag{6.7}$$

If we decide to use a non-dimensional barotropic vorticity equation in order to make its solution simpler, we can use the dimensionless versions of its variables, here represented by tildes: $q = V\tilde{q}, u = U\tilde{u}, v = U\tilde{v}, x = L\tilde{x}, y = L\tilde{y}$ and $t = T\tilde{t}$, where V, U, L and T are typical values for a realistic system. While simulating e.g. mesoscale systems, typical values that can be considered are: $U = 1ms^{-1}, L = 10^4m, V = 10^{-4}s^{-1}$ and $T = 10^4s$. Equation 6.7 then becomes:

$$\frac{V}{T}\frac{\partial\tilde{q}}{\partial\tilde{t}} + \frac{UV}{L}\tilde{u}\frac{\partial\tilde{q}}{\partial\tilde{x}} + \frac{UV}{L}\tilde{v}\frac{\partial\tilde{q}}{\partial\tilde{y}} = 0 \tag{6.8}$$

Considering that the temporal variable which appears in the vorticity term is related to advection, we note that $T = \frac{L}{U}$. So, by dividing all terms of equation 6.8 by $\frac{VU}{L}$, we obtain a non-dimensional version of equation 6.7:

$$\frac{\partial\tilde{q}}{\partial\tilde{t}} + \tilde{u}\frac{\partial\tilde{q}}{\partial\tilde{x}} + \tilde{v}\frac{\partial\tilde{q}}{\partial\tilde{y}} = 0 \tag{6.9}$$

in which the tildes can be removed from now on, for the sake of simplicity.

The barotropic vorticity model used for the experiments in this thesis actually considers a version of the previous equation with the addition of white noise. This way, we now obtain a stochastic differential equation with q as a random variable:

$$dq + (u\frac{\partial q}{\partial x} + v\frac{\partial q}{\partial y})dt = d\beta \tag{6.10}$$

and where $d\beta = Q^{1/2}dW$, i.e a random noise added by a space correlated Wiener process (Kloeden and Platen, 1992).

Equation 6.10 can be solved by considering the velocity field defined as the stream function (remember that the flow is assumed to be incompressible):

$$u = -\frac{\partial \psi}{\partial y} \tag{6.11}$$

and

$$v = \frac{\partial \psi}{\partial x} \tag{6.12}$$

The two-dimensional vorticity field can be described as:

$$q = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = \frac{\partial}{\partial x}\left(\frac{\partial \psi}{\partial x}\right) - \frac{\partial}{\partial y}\left(-\frac{\partial \psi}{\partial y}\right) \tag{6.13}$$

following equations 6.2, 6.11 and 6.12.

This way, by considering the equation above for the vorticity and reformulating equation 6.10, we obtain the main set of equations for the barotropic vorticity model, to be solved at every time step:

$$dq - (\frac{\partial \psi}{\partial y}\frac{\partial q}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial q}{\partial y})dt = d\beta \tag{6.14}$$

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = q \tag{6.15}$$

6.2.2 Numerical schemes

The numerical schemes used to update the previous set of equations are described as follows:

- Equation 6.14: a **semi-lagrangian** scheme is used to update the vorticity and an **Euler-Maruyama** scheme is used to integrate $d\beta$
- Equation 6.15: FFTs are used to compute the **inversion** of vorticity, in order to obtain the stream function to be used in equation 6.14 at the next time step.

- **Semi-Lagrangian scheme**: The main idea of using this scheme is to find the vorticity at time step $n+1$ by tracking the flow backward over a time interval Δt, considering that the field advected to time step $n+1$ contains the characteristics of the fluid parcels at time step n. The idea, then, is to know the departure points of these parcels, i.e. where those fluid parcels were at the previous time step.

The departure points are usually located in between grid cells, as shown in figure 6.1, where the departure point lies between grid points x_i and x_{i+1}, i.e. not corresponding to the formulated grid. This way, the vorticity at the departure point, at time step n, is determined by spatial interpolation. Considering that advection is the only process occurring in the system, then the values of the vorticity at the departure point at time n will be identical to those at the grid points at time $n+1$. A small enough value of $\Delta t = 0.04$ is used in this barotropic vorticity model, to ensure that the backwards neighbouring grid point trajectories do not cross.

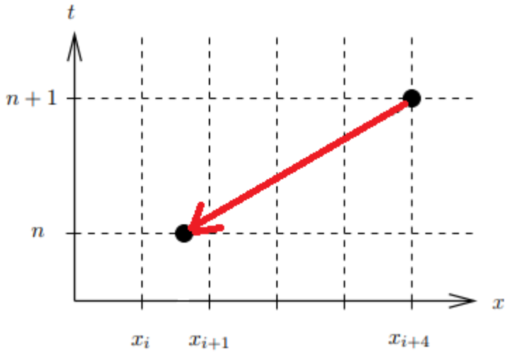


Figure 6.1: *Backward tracking of the flow, from current time $n+1$ to time n.*

For the backward tracking of the vorticity to its departure point at time n, we average the velocity field **u** over the time interval $[n, n+1]$ as:

$$\boldsymbol{u}^{n+\frac{1}{2}} = \frac{3}{2}\boldsymbol{u}^n - \frac{1}{2}\boldsymbol{u}^{n-1} \tag{6.16}$$

so obtaining a new velocity field for time step $n+\frac{1}{2}$. Note that in the first time step in which this procedure is to be performed, the velocity field for a previous time step is not known. In this case, we consider $\boldsymbol{u}^{n+\frac{1}{2}} = \boldsymbol{u}^n$.

The velocity $\boldsymbol{u}^{n+\frac{1}{2}}$ is then used to compute an estimate of the midpoint of the trajectory that ended at grid point x_{actual}, at time step $n+1$, such as:

$$x_{DP} = x_{actual} - \boldsymbol{u}^{n+\frac{1}{2}}(x_{actual})\frac{\Delta t}{2} \tag{6.17}$$

We repeat this computation twice more, after updating the velocity fields at grid point x_{DP} by cubic interpolation, although we must still start from the original grid point x_{actual}. As the final step, we repeat the procedure above, but not considering the factor of a half in the last term. We then obtain an estimate of the departure point which will end up at the actual point, at time $n+1$ and the vorticity at this computed point is calculated by cubic interpolation.

- **Euler-Maruyama scheme** This is a discretisation scheme used for stochastic differential equations. It considers that

$$\beta^{n+1} = \beta^n + Q^{1/2}\Delta W^n \tag{6.18}$$

where

$$\Delta W^n = W^{n+1} - W^n \tag{6.19}$$

and the value of the Wiener process at time step n is W^n. The random variables ΔW^n are independent and sampled from a normal distribution $N(0, \Delta t)$ at every time step.

For the barotropic vorticity model, the stochastic term in equation 6.18 is sampled from the distribution $N(0, I)$ and scaled by the matrix $Q^{1/2} = (V_\beta \tilde{Q})^{1/2}$, where V_β includes the value for Δt, a requirement by the Wiener process. These parameters will be described in more details later.

- **Inversion** The main idea is to invert the vorticity q to obtain ψ. Then, we next differentiate it, in order to get the velocities, following equations 6.11 and 6.12. To make this inversion feasible in this system, a Fourier transform $\tilde{\psi}(\omega_1, \omega_2)$ is used as follows:

$$\tilde{\psi}(\omega_1, \omega_2) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} e^{2\pi i k \omega_1 / N} e^{2\pi i l \omega_2 / N} \psi(k, l) \tag{6.20}$$

Considering that

$$F\left(\frac{\partial}{\partial x}\psi(x, y)\right) = 2\pi i N \frac{\omega_1}{N} \tilde{\psi}(\omega_1, \omega_2) \tag{6.21}$$

we can conclude that

$$F\left(\nabla^2 \psi(x, y)\right) = \left[\left(2\pi i N \frac{\omega_1}{N}\right)^2 + \left(2\pi i N \frac{\omega_2}{N}\right)^2\right] \tilde{\psi}(\omega_1, \omega_2) \tag{6.22}$$

back to equation 6.15, we can find that

$$\tilde{q}(\omega_1, \omega_2) = i^2 (2\pi N)^2 \left[\left(\frac{\omega_1}{N}\right)^2 + \left(\frac{\omega_2}{N}\right)^2\right] \tilde{\psi}(\omega_1, \omega_2) \tag{6.23}$$

or, if we want to relate it to the stream function

$$\tilde{\psi}(\omega_1, \omega_2) = \left(-(2\pi N)^2 \left[\left(\frac{\omega_1}{N}\right)^2 + \left(\frac{\omega_2}{N}\right)^2\right]\right)^{-1} \tilde{q}(\omega_1, \omega_2) \tag{6.24}$$

The last step is then to differentiate this equation, in order to obtain the Fourier transforms of the velocities, as in equations 6.11 and 6.12:

$$\tilde{u}(\omega_1, \omega_2) = -2\pi i N \frac{\omega_2}{N} \tilde{\psi}(\omega_1, \omega_2) \tag{6.25}$$

$$\tilde{v}(\omega_1, \omega_2) = -2\pi i N \frac{\omega_1}{N} \tilde{\psi}(\omega_1, \omega_2) \tag{6.26}$$

The velocities are found after inverting these transforms back to the physical domain.

6.2.3 Model configuration

As in the previous chapters, twin experiments are performed to investigate the use of synchronisation in the barotropic vorticity model, i.e. the truth is artificially generated by a stochastic run of the model, starting from a random vorticity field. A nondimensional amplitude $\lambda = 1$ and a spatial decorrelation length of 10 grid points were used, so the model could exhibit a chaotic behavior along the run period.

The barotropic vorticity model is tested with double periodic boundary conditions for 3 different grids: 32×32, 64×64 and 128×128, corresponding to $1,024$, $4,096$ and $16,384$ variables respectively. Again, only 25% of these systems are observed, i.e. every 2nd grid point in both meridional and zonal directions. Vorticity observations were generated by perturbing the truth run at every 10 time steps with an independent random additive observation Gaussian noise, with a standard deviation $\sigma_{obs} = 0.05$. Note that a typical size for the state variables in the barotropic vorticity model is between -3 and 3.

In the deterministic version of the barotropic vorticity equation, the flow tends to become more well organised as time progresses. The presence of a stochastic term $d\beta$ takes the role of injecting energy to the system, so it can keep a chaotic and turbulent behaviour throughout the run period.

The model error β is sampled from $N(0, Q)$, where the covariance matrix $Q = V_\beta \tilde{Q}$. The correlation matrix $\tilde{Q}$ represents the relation between the vorticity values at different grid points. These correlations depend only on a distance d between grid points, not involving their positions. The correlation matrix $\tilde{Q}$ was then generated from a second-order auto-regressive (SOAR) correlation function:

$$\rho(d) = \left(1 + \frac{|d|}{L}\right) exp\left(-\frac{|d|}{L}\right) \tag{6.27}$$

where L is the correlation lengthscale and for these experiments $L = 5$ was used. The distance d was related to the number of grid points between the variables, which allows for the periodic domain. Regarding the scaling V_β, it must include the timestep Δt, in accordance with the Wiener process used. In order to obtain a random error which was approximately 10% of a deterministic move, an l^2 norm of the vorticity difference between the state at time steps $n-1$ and n without including the random error was used. In comparison to the l^2 norm of the random error using different standard deviations, it could be found that a value of $V_\beta = (0.025)^2 \Delta t$ would be suitable for a tenth of the deterministic move.

Back to the sampling of β from $N(0, Q)$, in practice a random vector η is converted from a normal distribution $N(0, I)$ to samples from $N(0, Q)$. As described in chapter 5, this process is done by performing the multiplication: $Q^{1/2}\eta$. Different methods can be used to do so, but as this model uses double periodic boundary conditions, that means that we are dealing with a circulant matrix, which is an advantage on the computation of a product of a matrix and a vector. Circulant matrices have a special property of requiring only few Fast Fourier Transforms (FFTs) and a vector to compute the product. So, for a circulant matrix C we can obtain (Golub and Van Loan, 1996):

$$C(z) = F_n^{-1} diag(F_n z) F_n \tag{6.28}$$

where z is the first column of C, F_n is the FFT and F_n^{-1} represents its inverse. Considering the product $y = C(z)x$, we have:

$$\tilde{x} = F_n x \tag{6.29}$$

$$\tilde{z} = F_n z \tag{6.30}$$

$$w = \tilde{z} * \tilde{x} \tag{6.31}$$

$$y = F_n^{-1} w \tag{6.32}$$

The asterisk in equation 6.31 represents a component wise multiplication and to compute $Q^{1/2}\eta$ we use

$$w = \sqrt{\tilde{z}} * \tilde{\eta} \tag{6.33}$$

An additional computation that is needed in the particle filter framework is the computation of Q^{-1} for the calculation of the transition density $p(x^{j+1} \mid x^j)$, as described in equation 5.6. This probability is required to compute the weights in equation 5.12. Similarly to the computation of $Q^{1/2}\eta$, it is also possible to compute $Q^{-1}(x_i^{j+1} - f(x_i^j))$ in equation 5.6 through the use of FFTs. The difference is that now we should consider the inverse of the Fourier transform of z, i.e

$$w = \tilde{z}^{-1} * \tilde{\eta} \tag{6.34}$$

Another computation that is needed for both IEWPF and EWPF weight steps that occur immediately before the observation time step is the term $(HQH^T + R)^{-1}$. This matrix inversion is required to compute the particle weights. As we are dealing with a non-fully observed system, this matrix is not a circulant one and so we cannot use the FFT solver as described before. However, since the lengthscale of the correlation matrix $\tilde{Q}$ can be controlled in this system, it is possible to produce a sparse covariance matrix Q. This way, a numerical routine called HSL_MA87 that uses a direct method to solve large sparse positive-definite symmetric linear systems is used for this task.

The initial condition for the ensemble was generated by perturbing the random vorticity field with a multi-variate Gaussian noise. The covariance matrix used was $V_x\tilde{Q}$, where $V_x = 0.025^2$.

6.3 Implementation

The barotropic vorticity model was run with the particle filter on the High Performance Computing (HPC) machine ARCHER in a parallel fashion. ARCHER is the UK National Supercomputing Service which provides resources for researchers to run simulations that require parallelized schemes using large numbers of processing cores. It uses a Cray XC30 supercomputer with a total of 118,080 processing cores. To run each experiment in this work, though, only 3 cores were required.

All codes were written in Fortran90, running in parallel and using an MPI environment, through EMPIRE.

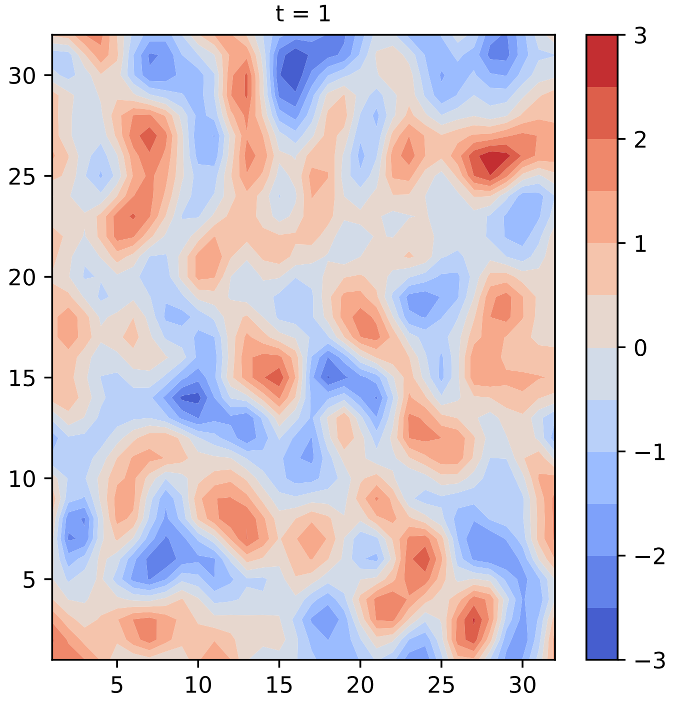
6.4 Experiments and results

As described before, the barotropic vorticity model used in this thesis is based in a stochastic version of the barotropic vorticity equation, yielding a turbulent behavior and chaotic flow structures. Figure 6.2 shows the evolution of the truth in this model, for a domain of 32 by 32 grid points with grid spacing $\Delta x = \Delta y = 1/32$, from time step $t = 1$ to $t = 300$, the end of the run period in the experiments presented here. It is possible to observe that different filaments and vortices are generated after 300 time steps.

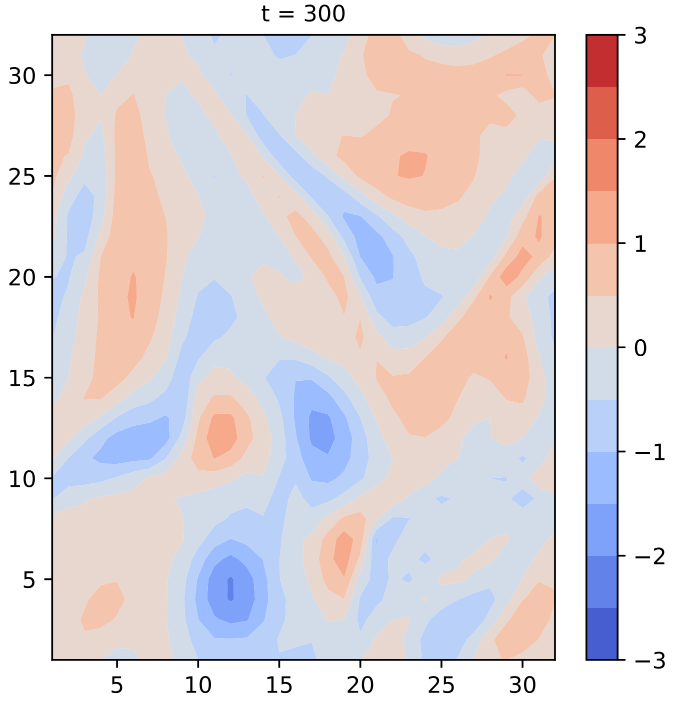
In the experiments presented next, a coupling constant $g = 10$ and a strength factor $gtau = 0.4$ were used. Still using figure 5.2 as a guidance on the possible combinations between g and $gtau$, the ones which lie in the middle green of the figure were tested for the barotropic vorticity model yielding the typical results presented here. Not all combinations were tested, though, due to the lack of time. Additionally, in these experiments a delay dimension $D_d = 3$ was used, showing that a smaller time embedding is needed in this 2-D domain, if compared to the Lorenz ring of variables. This is probably due to the differences on how the neighbouring variables communicate to each other in these different model grids.

The number of ensemble members is $N_{ens} = 25$. The ensemble is generated from the background error $N(0, B)$ in which $B = 100 * Q * \xi$ and ξ is sampled from $N(0, 1)$. The localisation radius in the synchronisation term is $loc = 10\Delta x$. Again, observations are available at every 10 time steps for every 2nd grid point in both meridional and zonal directions, i.e. only 25% of the system is observed.

The first part of this chapter will show results for the combination between the IEWPF and the ensemble synchronisation, so we can investigate how efficient this scheme can be in this type of system. The percentage of 50% of positive ε_i in the IEWPF is used, i.e. ε_i will be chosen randomly to be negative or positive, with the same probability.



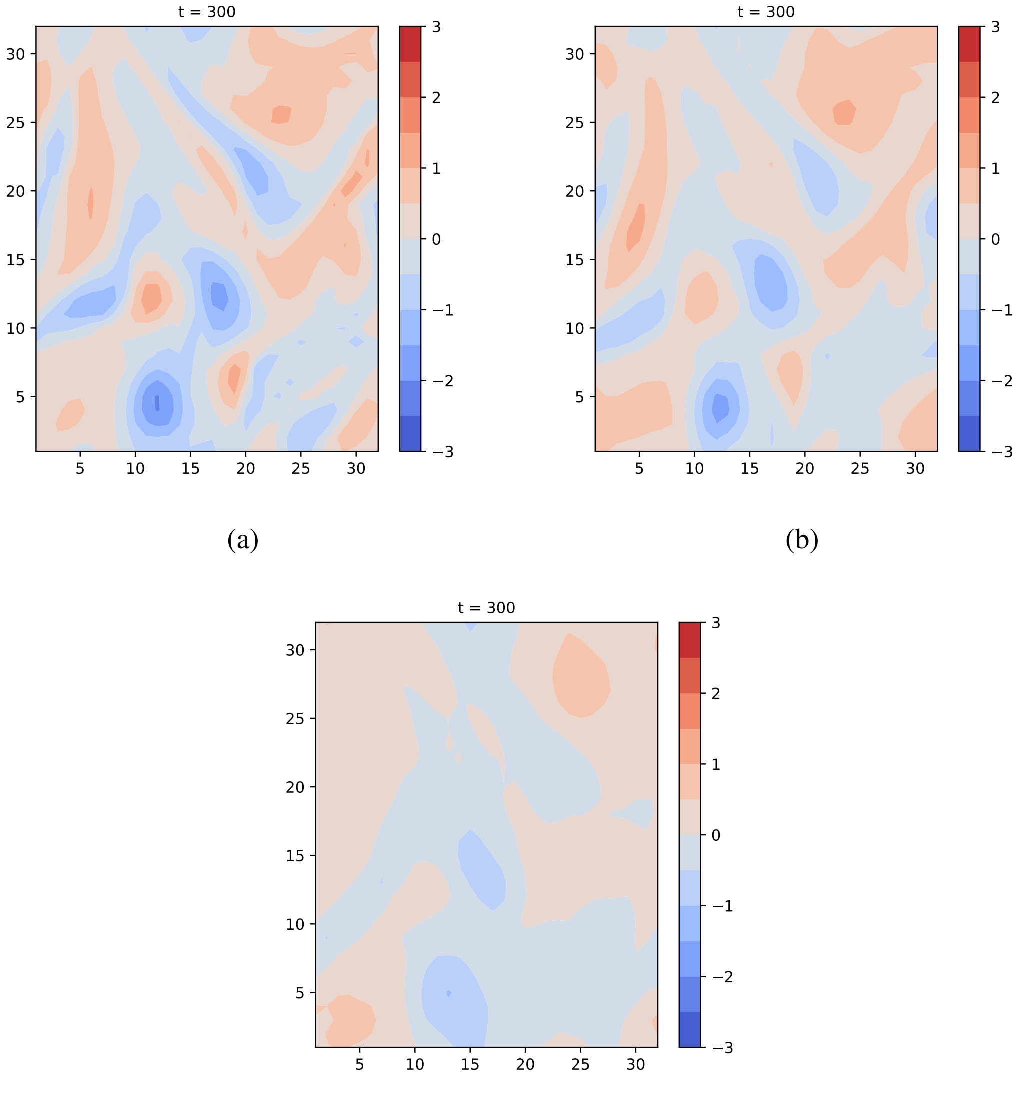
(a)



(b)

Figure 6.2: *Evolution of the true state of the barotropic vorticity model with a 32x32 dimension, from time step (a) t=1 to (b) t=300.*

Figure 6.3 shows a comparison between the true state, the mean of the 25-particle ensemble using the IEWPF+EnSynch and a free stochastic run in this identical twin experiment. It is noticeable that the particle filter represents the filaments and vortices in more details if compared to a simple stochastic run. And although some grid points have values slightly below the true ones, the particles are still representing most of the features of the true state, meaning that it is being sampled from a suitable region of the posterior.



(a)

(b)

(c)

Figure 6.3: *Comparison between: (a) the truth, (b) mean of the 25-particle ensemble for the IEWPF+EnSynch and (c) stochastic run. All states at time step t=300.*

In figure 6.4 we can visualize the global RMSE (for observed and unobserved variables) and the spread of the ensemble during the run period. The RMSE until $t = 300$ shows a tendency of increase. We note in figure 6.5 (a), which exhibits RMSEs for 1,000 time steps, that this tendency is kept until around $t = 500$, after which the RMSE starts to exhibit a behavior towards stabilisation. This is clearer in figure 6.5 (b), where less members were used in the ensemble

$(N_{ens} = 15)$. As mentioned in the last chapter, by decreasing the number of particles we obtain a smoother behavior in the RMSE values and also in the spread of the ensemble. The spread curves show that the ensemble exhibits a good variance. Figure 6.6 gives us an idea of how varied the results are for the 25 members at time step $t = 300$. It shows that the particles are exploring a desirable region of the state, so the mean obtained is fairly close to the truth.

Figure 6.7 exhibits the states for the truth in the first column and the ensemble mean in the second at each 50 time steps from $t = 50$ to $t = 300$. We note that during the evolution of the model, the particle filter follows the true features closely.

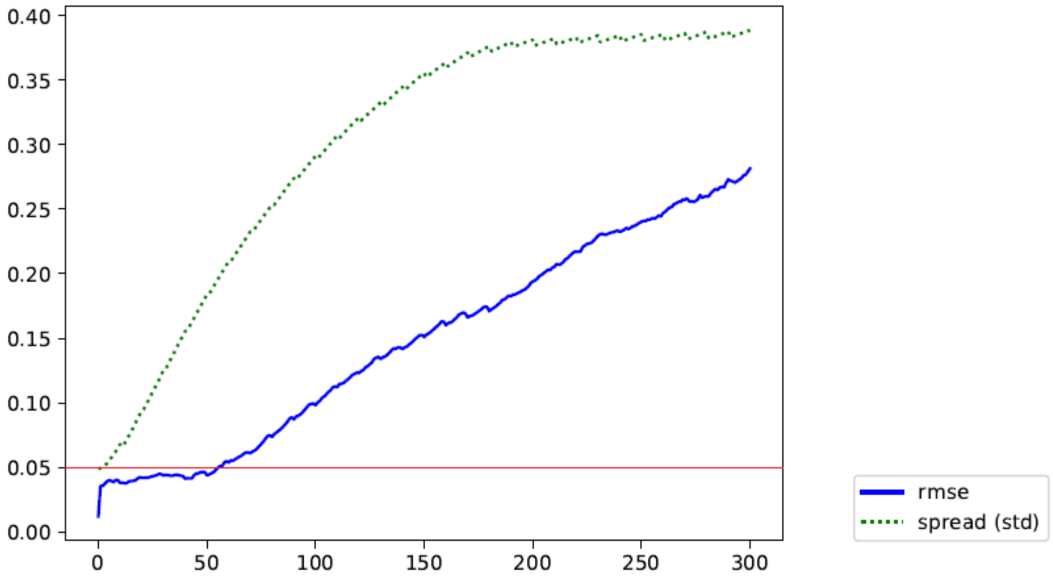


Figure 6.4: *Global RMSE and spread of the ensemble for the IEWPF+EnSynch during 300 time steps. The horizontal red line represents the standard deviation of the observation noise.*

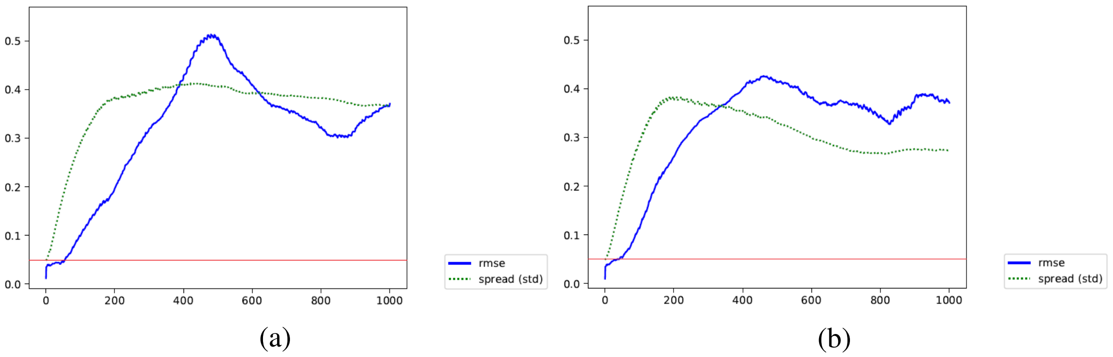


Figure 6.5: *Global RMSE and spread of the ensemble for the IEWPF+EnSynch during 1,000 time steps for (a) $Nens = 25$ and (b) $Nens = 15$. The horizontal red lines represent the standard deviation of the observation noise.*

The previous results exposed so far show that the IEWPF can produce reasonable mean states, if compared to a pure stochastic run, as the vortices and filaments present in the turbulent barotropic vorticity model can be described in the ensemble mean states during the run.

Aiming to investigate if these results could be improved, new experiments were performed using the ensemble synchronisation scheme as a proposal density in another particle filter: the EWPF. As described in chapter 2, the equivalent-weights particle filter makes sure that similar relative weights are attributed for the majority of the particles, so avoiding filter degeneracy. For these EWPF experiments 70% of the particles were retained and the effect of increasing this percentage will be discussed further.

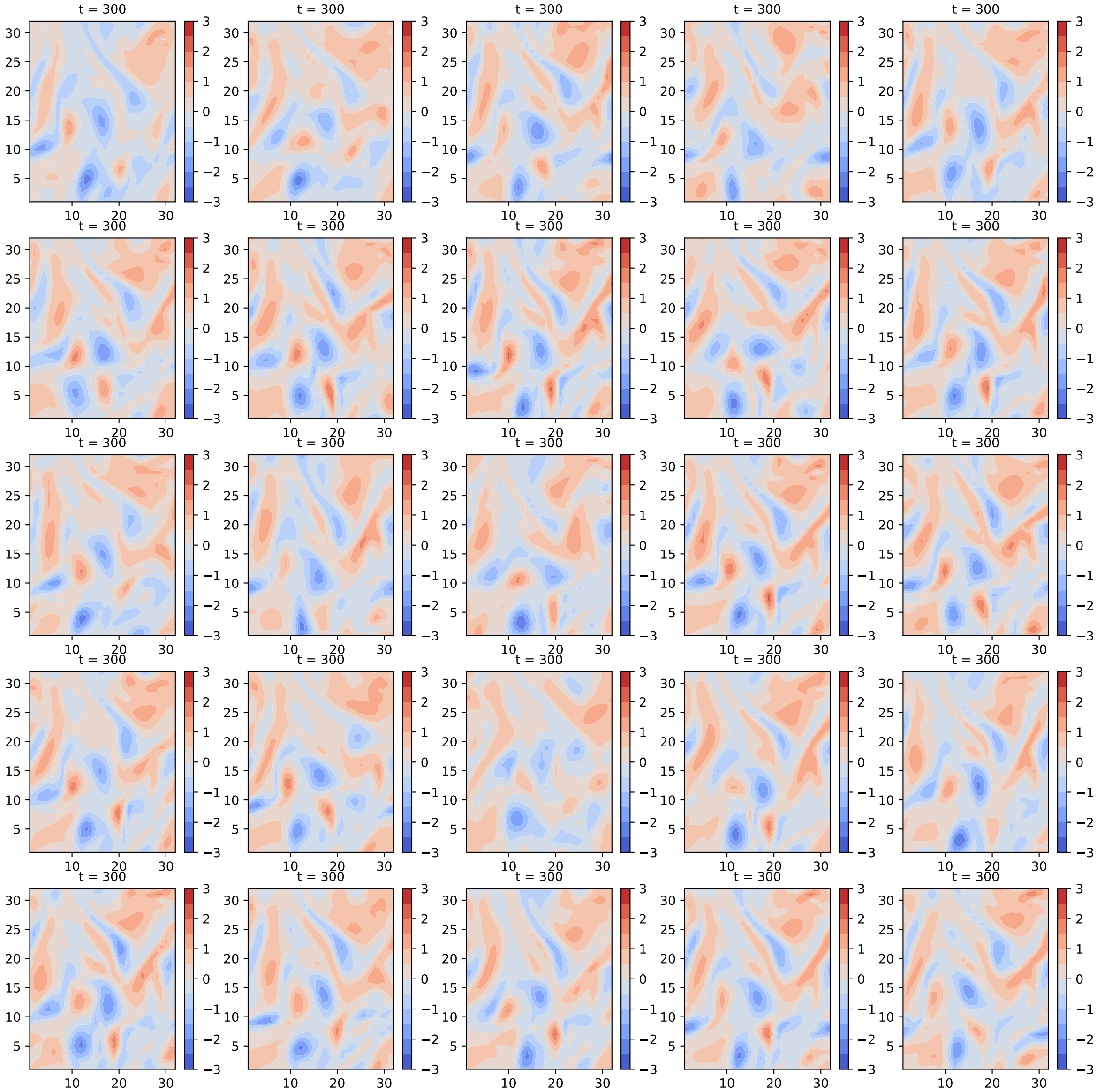
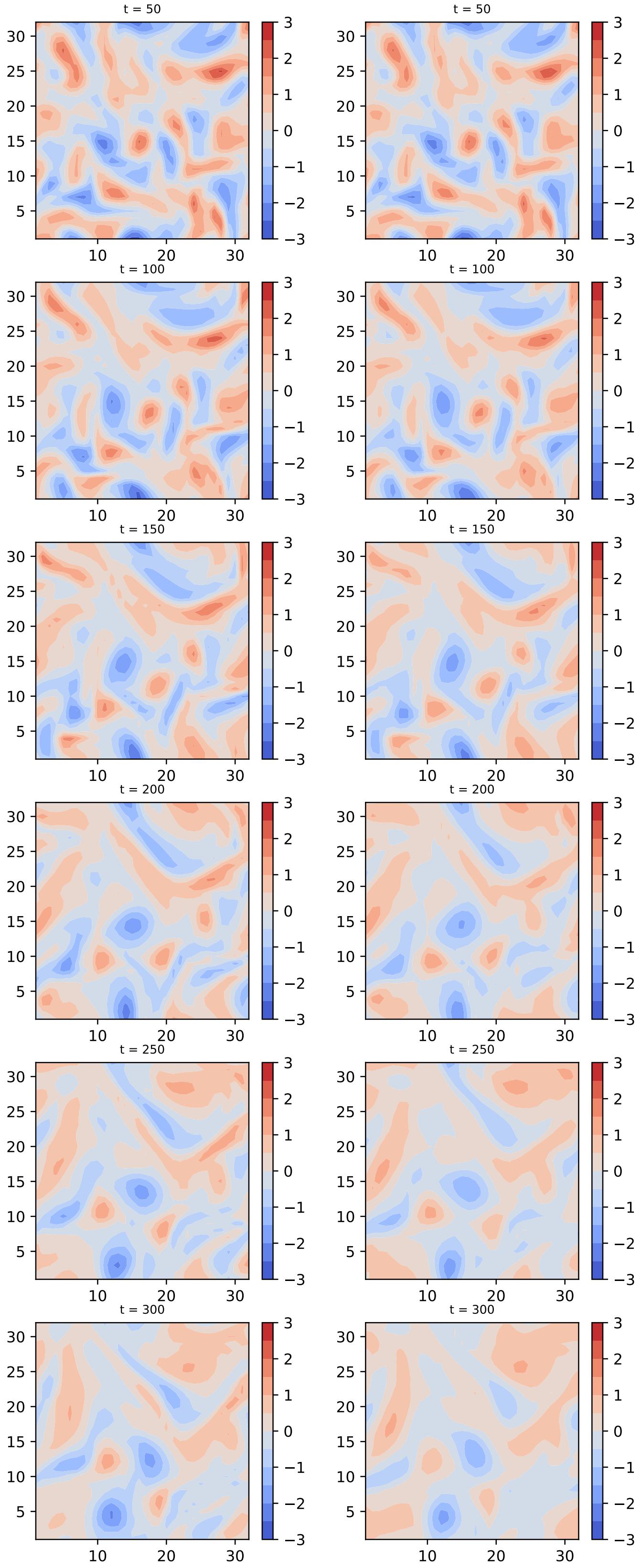


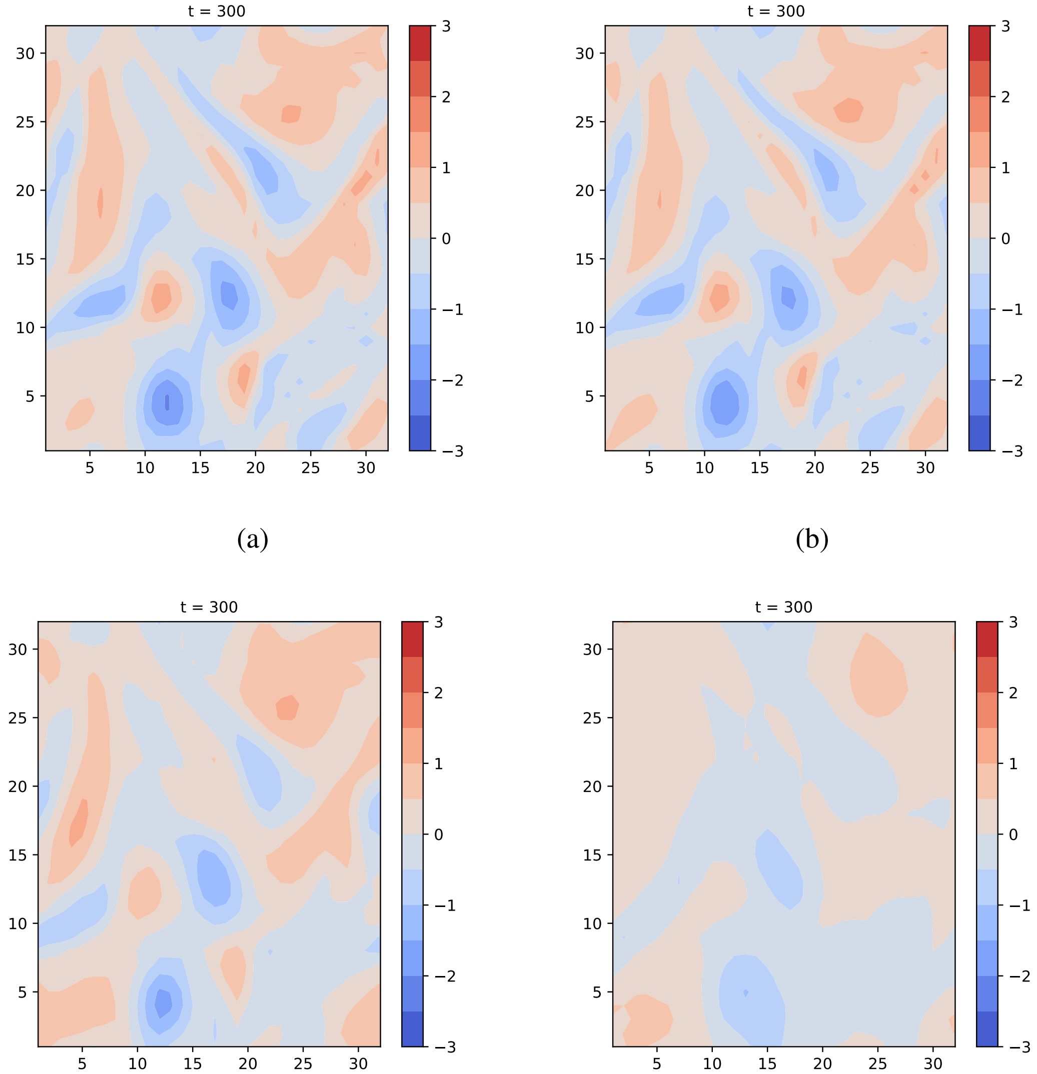
Figure 6.6: *States for each of the* 25 *members of the ensemble for the IEWPF+EnSynch at time step* $t = 300$.



(a)

(b)

Figure 6.7: *States for the truth (a) and the ensemble mean for the IEWPF+EnSynch (b) at each* 50 *time steps from* $t = 50$ *to* $t = 300$.



(a)

(b)

(c)

(d)

Figure 6.8: *Comparison between: (a) the truth, (b) mean of the 25-particle ensemble for the* ***EWPF+EnSynch****, (c) mean of the 25-particle ensemble for the* ***IEWPF+EnSynch*** *and (d) stochastic run. All states at time step t=300.*

Figure 6.8 shows a comparison between the truth and the state means for the EWPF+EnSynch, IEWPF+EnSynch and a stochastic run, indicating how well the particles are able to sample from the high probability region of the posterior for the EWPF+EnSynch case (figure 6.8 (b)). It is noticeable that the result for this particle filter overperforms the estimations for the IEWPF in this model. Figure 6.9 shows the values for $(truth - mean)^2$, indicating that in the vast majority of the grid points we observe that this quantity is below 0.02. This is also represented in figure 6.10, which shows that, at time step $t = 300$, the EWPF reaches a global $RMSE = 0.07$, while the IEWPF reached a global $RMSE = 0.27$, as previously shown in the figure 6.4. We note, indeed, that if we look at a longer run of 1,000 time steps (figure 6.11), the EWPF+EnSynch combination maintains this stabilised behavior towards the end of the run, until it reaches a global $RMSE = 0.06$, practically reaching the same value of the standard deviation used for the observation error which is $\sigma_{obs} = 0.05$. This is a significant result, considering that only 25% of the system is observed. The few jumps that occur in the RMSE shown in figures 6.10 and 6.11 may probably come from the EWPF itself, as they seem to happen at observation times. It means that the target weight is probably quite low, so the particles are pushed away from the high probability areas in the posterior pdf.

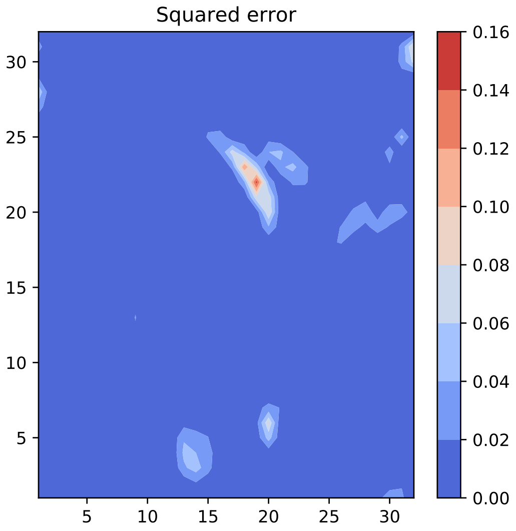


Figure 6.9: *Square of the truth minus the ensemble mean at time step $t = 300$ for the EWPF+EnSynch.*

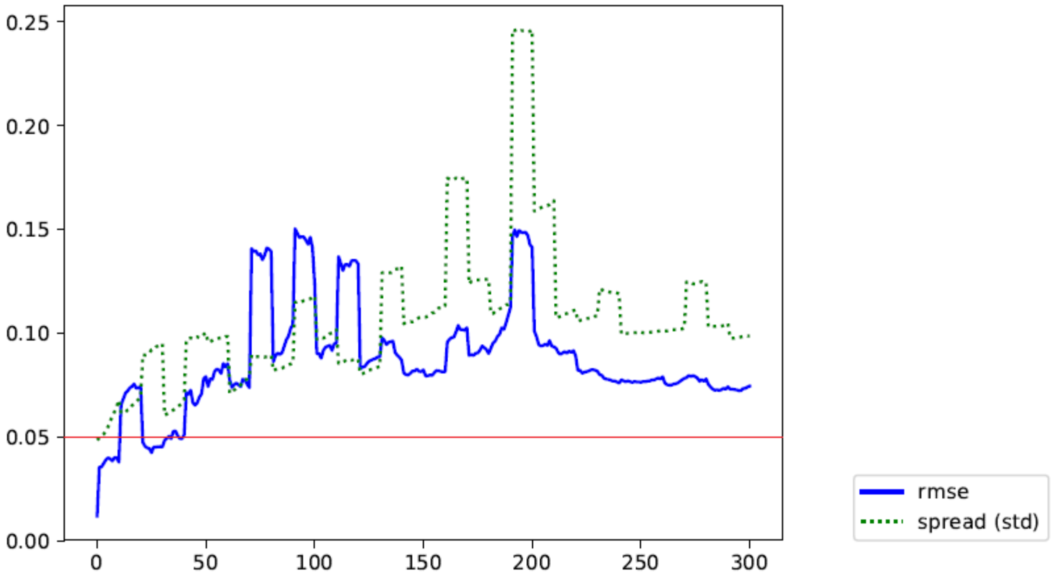


Figure 6.10: *Global RMSE and spread of the ensemble for the EWPF+EnSynch during 300 time steps. The horizontal red line represents the standard deviation of the observation noise.*

Regarding the spread, we observe a slightly overdispersive ensemble in figure 6.11, which actually seems to show that the ensemble is exploring different possible realisations and thus providing a good representation of the truth.

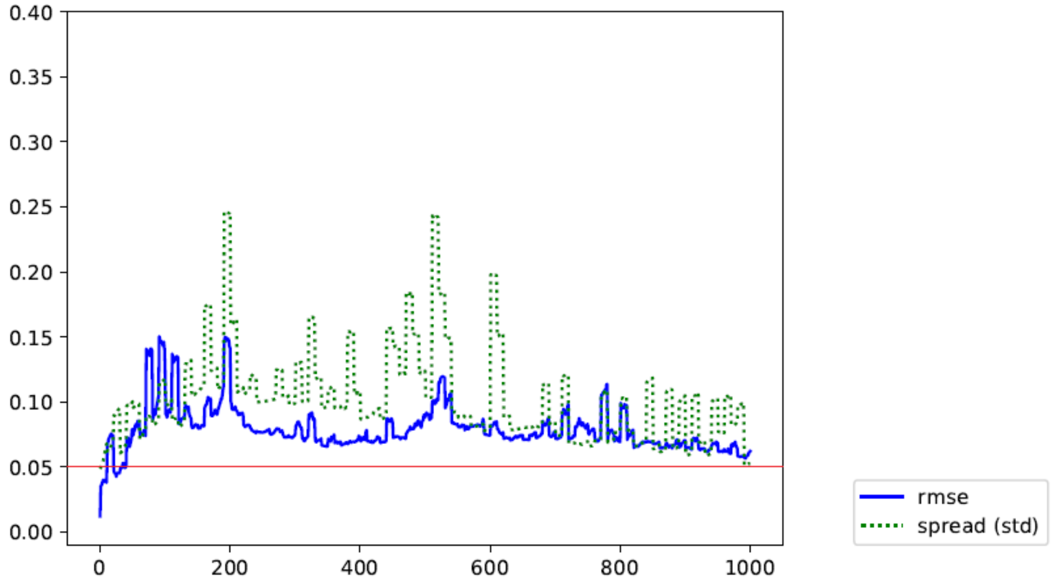


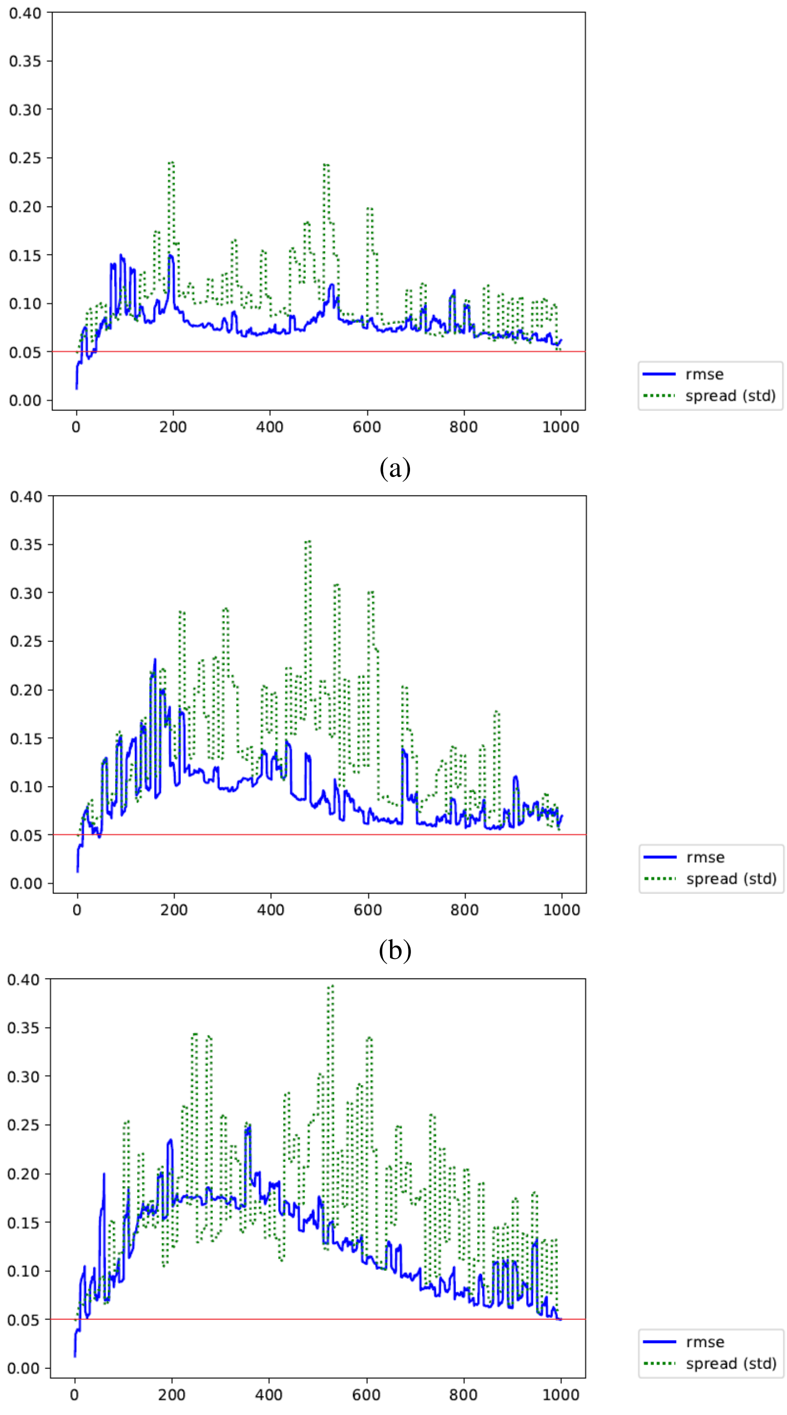
Figure 6.11: *Global RMSE and spread of the ensemble for the EWPF+EnSynch during 1,000 time steps. The horizontal red line represents the standard deviation of the observation noise.*

Aiming to investigate a bit further about those ”jumps” in the RMSEs, tests were performed to understand the sensitivity of the EWPF+EnSynch system to a change in the percentage of particles retained in the EWPF step. Figure 6.12 shows RMSEs for 1,000 time steps for different number of particles retained: 70%, 85% and 95%. It gets noticeable that when retaining more particles, the ”jumps”, the RMSE and the spread all increase, particularly during the first half of the period. As explained in Ades (2013), by increasing the percentage of particles, one tends to decrease the target weight, yielding a larger change in model state for the majority of particles. This larger movement tends to reduce the number of particles which are close to the observations at the EWPF time step, as the parameter α_i which appears in equation 2.64, will then increase. Also, Ades and van Leeuwen (2014) show that, while retaining more particles (e.g. 90%), the ensemble becomes more overdispersive. This can be noted in figure 6.12. This can justify the differences in the RMSEs in the figure and that is why the percentage of 70% of particles is used in the experiments presented in this work.

Figure 6.13 exposes the varied states produced by each of the 25 ensemble members at time step $t = 300$. The particles explore a region of high probability of the state, obtaining a mean which is pretty close to the true state.

To have an idea of the evolution of the mean states produced by the EWPF+EnSynch, figure 6.14 exhibits the states for the truth in the first column and the ensemble mean in the second at each 50 time steps until $t = 300$. The filaments and vortices produced by the truth are all reproduced by the particle filter, until the end of the period.

As this scheme was successful in representing the high probability region of the posterior



(a)

(b)

(c)

Figure 6.12: *Global RMSE and spread of the ensemble for the EWPF+EnSynch during 1,000 time steps for different percentage of particles in the EWPF: (a) 70%, (b) 85% and (c) 95%. The horizontal red lines represent the standard deviation of the observation noise.*

pdf so far, a further last test is to increase the dimension of the system from a grid of 32×32, i.e. 1,024 variables to a grid of 64×64, i.e. 4,096 variables and 128×128, i.e. 16,384 variables.

The configuration used for these higher-dimensional systems is the same as the one used before, apart from the fact that the time embedding used for the 64×64 system is $D_d = 4$ and for the grid of 128×128 the time embedding was decreased to $D_d = 3$. Also, as the grid spacings

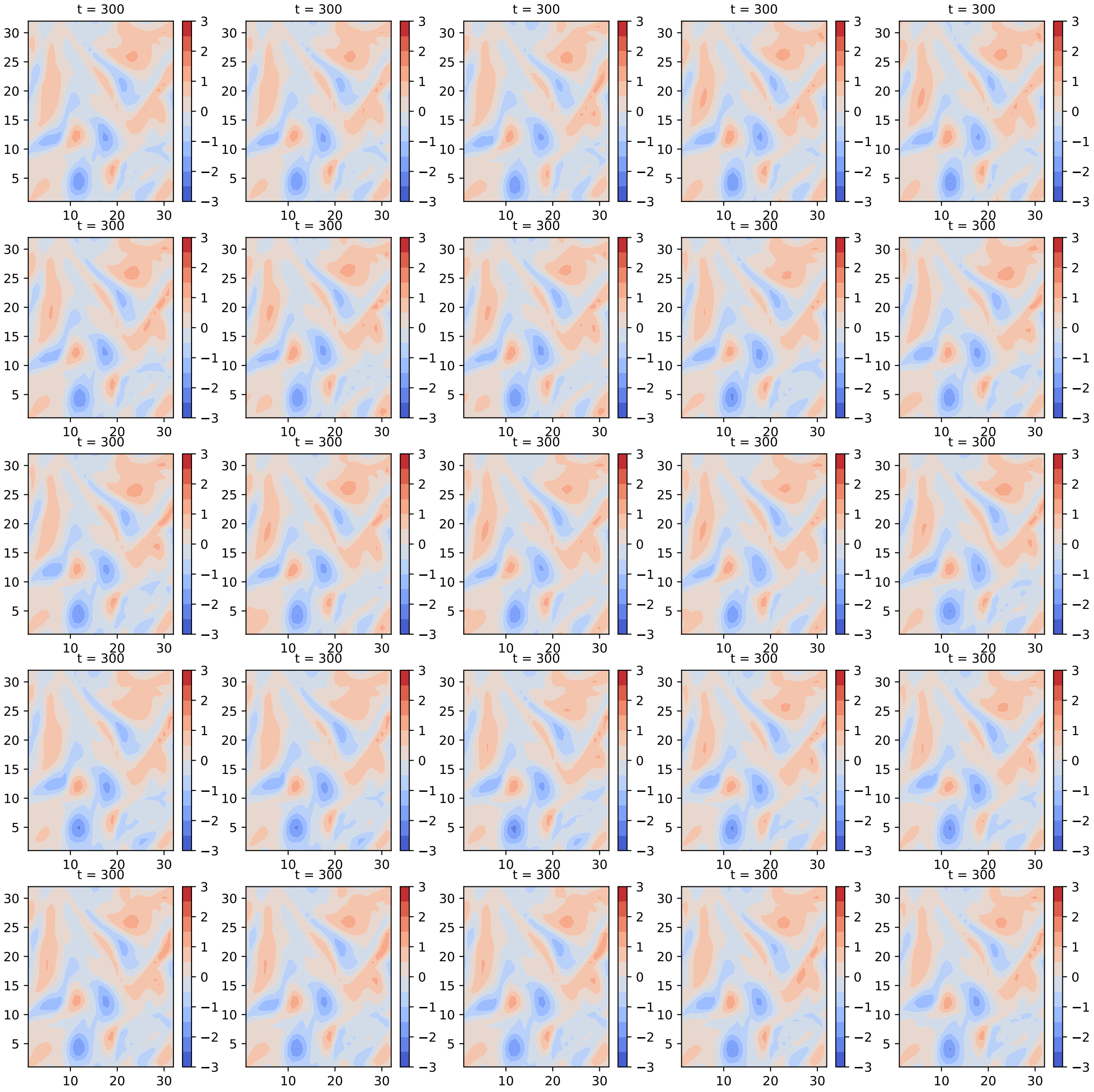
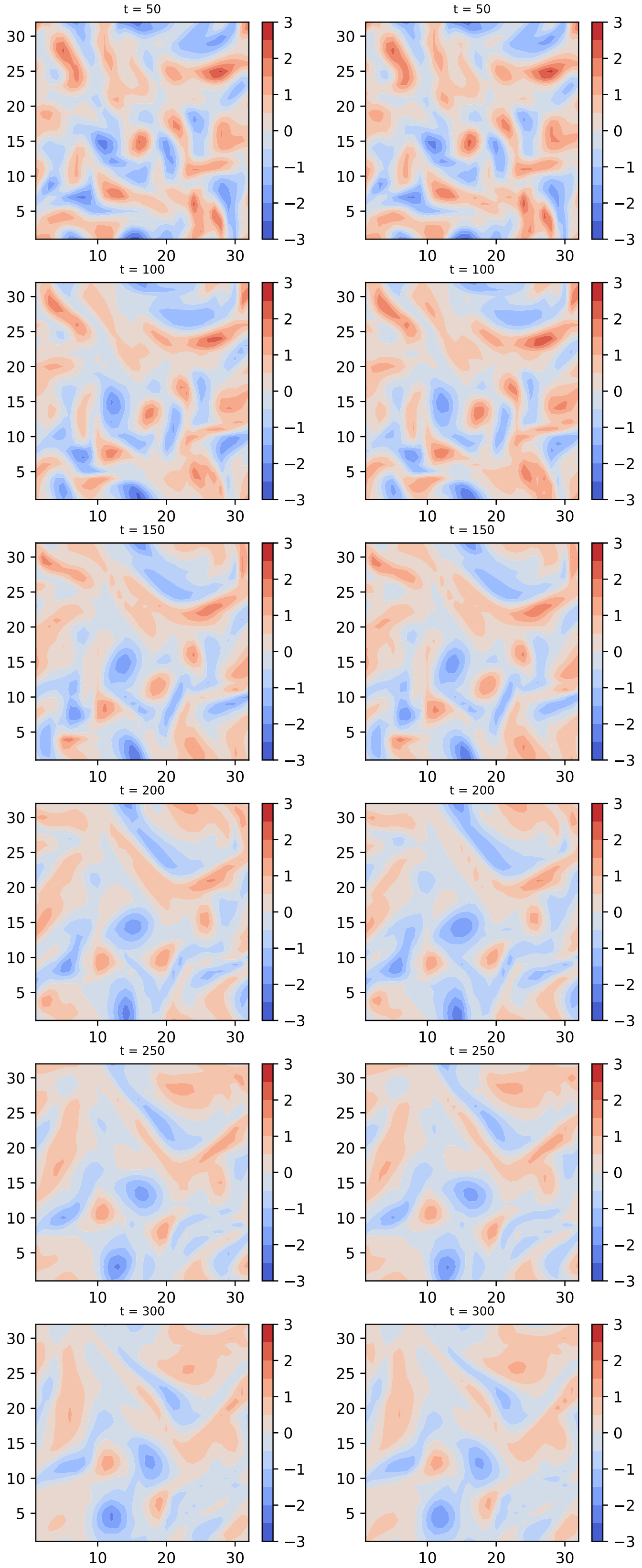


Figure 6.13: *States for each of the* 25 *members of the ensemble for the EWPF+EnSynch at time step* $t = 300$.

have changed to $\Delta x = \Delta y = 1/64$ and $1/128$, the values of the localisation radius do change as well, as they are related to the grid spacing by $loc = 10\Delta x$.

Figure 6.15 exhibits a comparison for the 64×64 grid between the truth, the mean of the 25-particle ensemble using the EWPF+EnSynch and a free stochastic run. The particle filter reproduces the filaments and vortices in more details if compared to the stochastic run, representing most of the features of the truth, meaning that it keeps sampling from a desirable

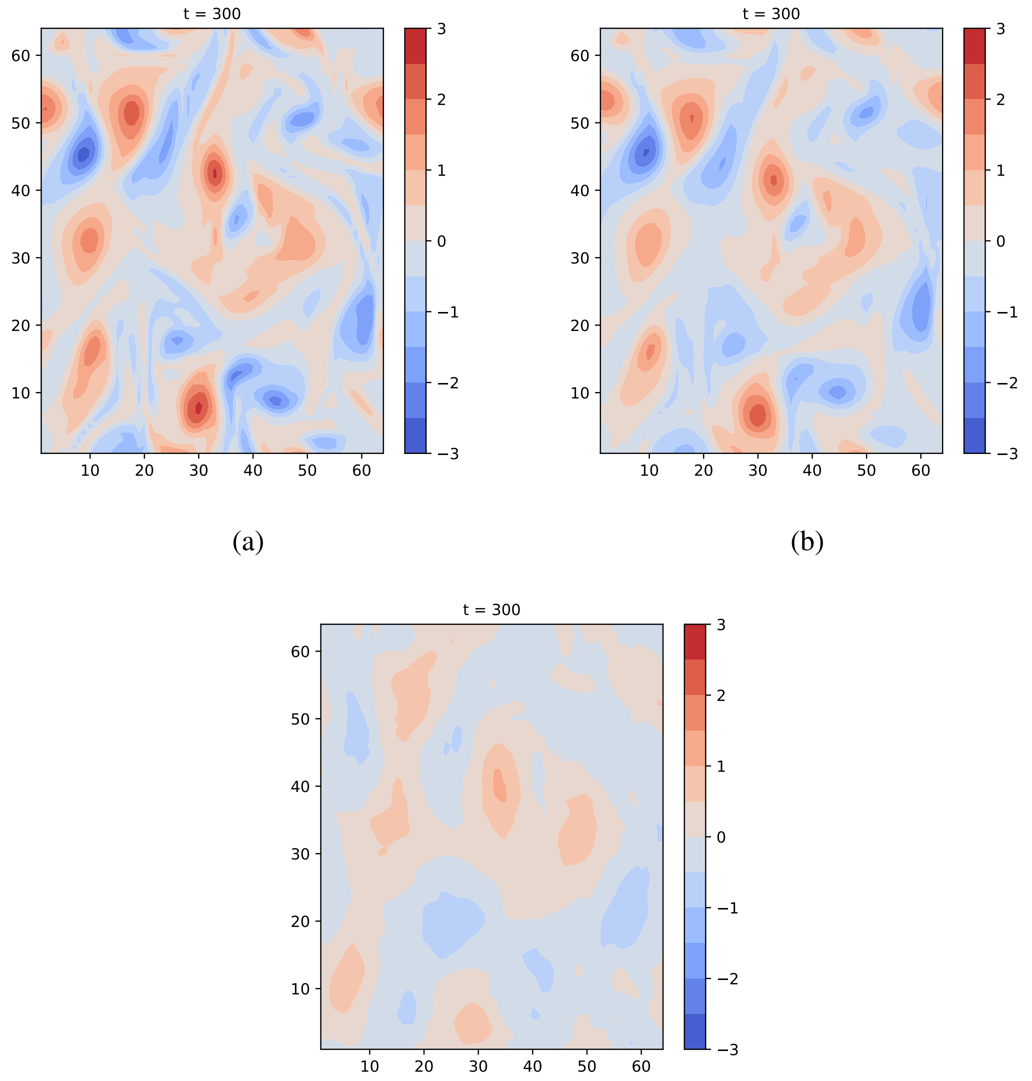


(a)

(b)

Figure 6.14: *States for the truth (a) and the ensemble mean for the EWPF+EnSynch (b) at each* 50 *time steps from* $t = 50$ *to* $t = 300$.

region of the posterior pdf. The global RMSE (not shown) does not reach the same order of magnitude of the observation error ($RMSE = 0.26$), but further improvements to this result can be done, if one performs more experiments with this configuration.



(a)

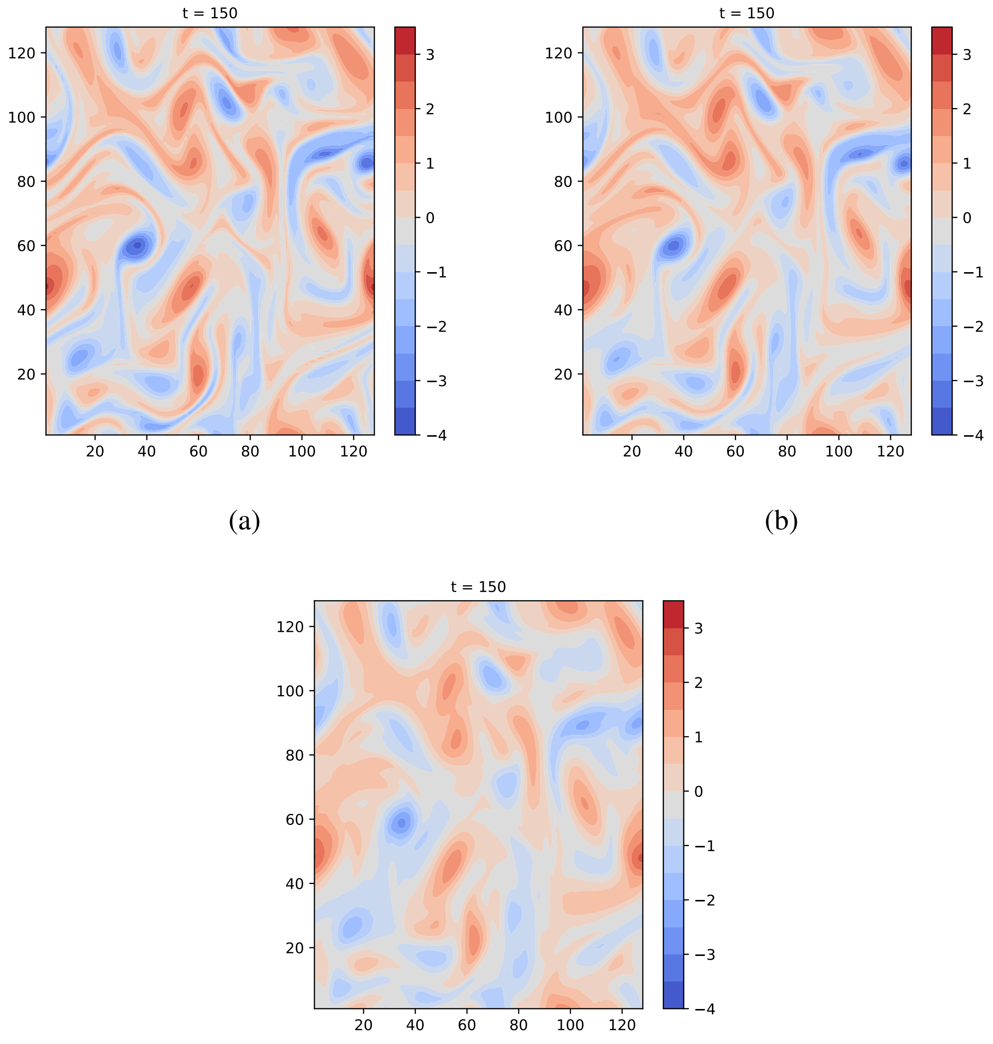
(b)

(c)

Figure 6.15: *Comparison between: (a) the truth, (b) mean of the 25-particle ensemble for the EWPF+EnSynch and (c) stochastic run for a system of 4,096 variables. All states at time step t=300.*

Figure 6.16 shows the same comparison presented in the previous figure, but for the 128×128 grid during 150 time steps. We note that, despite the decreased run period, the stochastic run is already missing the features located at the top right corner and in the middle of the left side of the grid. The particle filter, however, keeps reproducing the filaments and vortices that appear in the truth, in details.

These results suggest that the dimension of the underlying system does not influence the performance of the scheme, particularly when we remind ourselves that no additional tunings were performed while increasing the grid size.



(a)

(b)

(c)

Figure 6.16: *Comparison between: (a) the truth, (b) mean of the 25-particle ensemble for the EWPF+EnSynch and (c) stochastic run for a system of 16,384 variables. All states at time step t=150.*

We note, however, that the experiment with the bigger grid was indeed computationally expensive (which was the reason for the decrease in the time embedding to $D_d = 3$, still yielding a satisfactory result, anyway). The most probable reason for this issue is the computation of the truncated SVD for the pseudoinverse in the synchronisation coupling term.

On this computational issue, it is possible to implement more efficient solutions. Note that the SVD is actually thin, i.e. of size $(D_d * D_y) \times N_{ens}$. There are special routines for this type of matrix, in which a QR decomposition is first computed. Another solution could be the use of the Arnoldi algorithm to calculate an approximate SVD. There are other techniques and routines around, which could be implemented and make the process more efficient. Due to lack of time, though, this is not included in this thesis.

6.5 Summary

This chapter presented results obtained for experiments with the nonlinear barotropic vorticity model, using two non-degenerate particle filters, the IEWPF and the EWPF, both with a proposal density between observations composed of the ensemble-based synchronisation scheme. The main idea was to test the effectiveness of the synchronisation coupling term applied to each particle of a filter in a fluid dynamics high-dimensional model.

Quantitatively consistent results were obtained with the ensemble synchronisation scheme as a proposal density in the particle filters tested, but its effectiveness may vary depending on the particle filter which is being used. Also, it should be noted that the scheme was only tested with a linear observation operator, Gaussian observation error and Gaussian errors in the model equations.

Results for the combination with the IEWPF showed reasonable estimates for 300 time steps, as the ensemble mean was able to reproduce the filaments and vortices contained in the true state. The values of these features, however, were not so well represented, leading to global RMSE values which were still above the observational error introduced to the system, even when the run period was extended to 1,000 time steps.

The experiments with the EWPF using the EnSynch scheme as a proposal density have generated much better results. The comparison between the ensemble means and the truth along the 300-time step run has shown a very good representation of the high probability region of the posterior pdf, yielding global RMSEs of the same order of magnitude of the observation noise and the reproduction of all vortices and filaments apparent in the truth with more details than in the IEWPF experiments. These results may suggest an efficient performance of the combination between the EWPF and the ensemble synchronisation scheme. The ensemble spread for this filter is slightly overdispersive, but this fact does not seem to influence the means found. Actually, it is probably a good sign of how efficiently the ensemble explores the state space and so generating good estimates until the end of a run period of 1,000 time steps.

The EWPF has some specificities which affect directly its performance like, e.g. the percentage of particles kept in the equivalent-weights step, as their maximum weights are greater than the target weight chosen. In the experiments performed in this thesis, it was noticeable that keeping a greater number of particles tends to increase the ensemble spread and generate a larger movement of the particles towards the observations. Consequently, we observe more ”jumps” in the RMSE curves and a slight increase in their values. The ”jumps” are probably related to the equivalent-weight step and its target weight, which is probably low.

Other parameters like the root of α_i or the choice of γ_U, γ_N or ϵ, related to the mixture density, were not tested in these experiments and the values used were the ones described in chapter 2, following Ades (2013).

Note that better results than the ones presented here can still be found, as further experiments can be performed, testing different values for the constants g and $gtau$, time embeddings D_d or even the EWPF parameters. The number of particles retained and used in the ensemble could also be tested, so one can find an optimal configuration for this model.

Additionally, it is noticeable that computational improvements are still needed in the synchronisation scheme, as the calculation of the truncated SVD is still very computationally demanding. Several routines and/or techniques are available and can be used as an option to optimize the scheme. By improving the computational efficiency of the system and the synchronisation configuration, good results can be expected based on the simulations done so far, where RMSEs reached the same order of magnitude of the observation error included in the system.

Due to lack of time, it was impossible to consider and test every aspect of the behavior of the synchronisation scheme in these experiments. The results presented in this work, although not optimal, are indeed significant, considering the challenge of the problem.

Clearly, further investigations are still needed on the combination between both filters and the ensemble synchronisation as their proposal. Issues like the ones presented by the IEWPF may be related to the way some of its parameters are tuned (as this is the first time it is tested in the barotropic vorticity model) or even to the approximations made while deriving the IEWPF, which could be improved. While analysing the results obtained for the EWPF and focusing on the main purpose of this thesis, i.e. investigate the feasibility of using synchronisation concepts in a particle filter, the overall message is that the ensemble synchronisation is able to work as an efficient proposal density in this system, opening and connecting the synchronisation field to the data assimilation area in a direct way.

Chapter 7:

CONCLUSIONS

7.1 General remarks

The main goal of this thesis was to investigate the feasibility of combining the advances in the synchronisation theory with a methodology which has been proposed to be a promising fully nonlinear data assimilation method: the particle filters. To this end, the advantages of both were merged and explored.

To better understand the data assimilation field, a brief overview on the popular methods was described, along with the issues that the assumptions behind them can bring. Also, the particle filtering formulation was briefly explained, highlighting its well-known limitations, followed by an existing solution to tackle these issues: the proposal density freedom. Two non-degenerate particle filters, the IEWPF and the EWPF, were used in this thesis. The main idea was to include the synchronisation framework as a relaxation proposal density in these formulations.

On the synchronisation side, this work presented a summary on the most important topics related to the theory and different techniques that have been developed in the field, always trying to make a connection with data assimilation methodologies. We note that, although some similarities between these methods do exist, there are still major differences and the potential to merge their advantages gets tempting and intriguing.

To this end, two new ideas were explored in this work:

1. An ensemble-based synchronisation scheme was developed, making the technique proposed by Rey et al. (2014a) more applicable for high-dimensional systems. Some thoughts on the usefulness of the time embeddings and the ensemble smoothers were discussed, enriching the way we view some well-known data assimilation methodologies. It was also possible to learn a bit more about how the system works in order to stabilise the synchronisation manifold in a partially observed system and the role that the observation noise plays in it. The results obtained have proven the usefulness of the scheme in keeping

the model trajectory close to the truth, with state-averaged synchronisation errors smaller than the observation errors introduced, considering a partly observed system. This new synchronisation scheme is actually a big step forward in the synchronisation field, opening this wide area to high-dimensional applications. A direct interest for implementation of this new scheme has come from the physics field, in the cardiac dynamics control area.

2. The new scheme was then applied in a fully nonlinear particle filter, as a relaxation proposal density. The main idea was to use the synchronisation term on each particle, to efficiently bring the ensemble of trajectories closer to the truth, while controlling local instabilities in the dynamics of the system. The formulation used was relatively simple, avoiding the need for tangent-linear models and adjoint calculations, a clear advantage. Experiments in a 1-D and 2-D models were performed, testing two different particle filters that ensure equal weights for all particles, avoiding filter degeneracy by construction: an extension of the Implicit Equal-Weights Particle Filter and also the Equivalent-Weights Particle Filter. Note that the schemes used in this work were only tested with a linear observation operator and Gaussian errors in the model equations. Qualitatively good results were obtained in these nonlinear schemes, which also provided for the first time a glimpse on a comparison between both particle filters. The pros and cons of these new ideas were discussed, always including thoughts on how they can be improved.

7.2 Answers to key questions

There were two key questions posed in the first chapter and these have formed the motivation for each of the main work chapters. Reaching the end of this work, the answers can summarise the goals of this thesis:

1. *Is it possible to improve the existing synchronisation framework by developing a scheme which can be robust to high-dimensional systems and can also take into account realistic observation settings?*

   A new ensemble synchronisation scheme was developed, yielding successful results and a clear convergence between the model trajectory and the truth. The new scheme was tested in a nonlinear large-dimensional system and takes into account realistic observation errors. The development of this scheme helps to open the synchronisation field to high-dimensional systems.

2. *Can we use synchronisation into a particle filter and make the whole scheme more efficient in nonlinear systems?*

   While applying the synchronisation framework into a particle filter, the proposal density freedom was fully explored. This combination has presented some advantages over the main data assimilation methodologies, as it avoids tangent-linear models and adjoints and also because its formulation is not based on the Gaussian prior. Experiments with nonlinear models were used to investigate the efficiency of this combined methodology, producing satisfactory results. Note that although the method can handle full non-linearities, the experiments conducted here considered a linear observation operator, Gaussian observation errors and Gaussian model errors. Also, a further investigation was conducted: a first comparison between two non-degenerate particle filters using the same relaxation proposal density, i.e. the ensemble synchronisation. A whole field of future work arises from the interesting results obtained.

Thereby, an overall message from the results obtained in this thesis is that the use of the ensemble-based synchronisation scheme as a proposal density in a nonlinear, non-degenerate particle filter is feasible, as the synchronisation scheme has shown to be an efficient tool to steer model states to the truth, yielding satisfactory results in this first feasibility study ever made about merging synchronisation and particle filters.

7.3 Future Development

To conclude this work, it is essential to list all the thoughts that have arisen along the construction of this thesis, which could only be included in a future research section, due to the lack of time to complete a more comprehensive study.

Among many of the interesting topics that appeared during this work and worth further investigations, the following can be mentioned:

- After the development of the ensemble-based synchronisation scheme, results have shown that the ensemble size also played a role in the stabilisation of the unstable modes in the synchronisation manifold. Indeed, it was found that there is some kind of relationship between the time embedding dimension, the number of observations and the ensemble size, so one can better characterise the observational requirements to reach synchronisation. This is an intriguing characteristic of the system which still needs further investigation.
- Test the influence of a non-uniform observation network in the systems used in this work.
- Test the influence of different localisation radii in the formulation.
- Results suggested that the embedding dimension $D_e = D_d * D_y$ needs to be of the order of the system dimension, so one can obtain RMSEs below the observation noise. Further research is needed to confirm this remark, though.
- Understand the action of the synchronisation coupling term in different trajectories of the particles in a particle filter, differently from what happens in the stand-alone ensemble synchronisation, where this term acts and controls only one trajectory.
- Investigate the effects that the particle filter steps to equalise the weights of the particles play

on synchronisation, as it can be seen as an ”interruption” to the synchronisation process. Additionally, an interesting point is if this effect changes from a 1-D to a 2-D grid model.

- When observing a larger part of the state space, interesting relations between the time delay and the coupling constant g could be noticed: it was possible to reduce the size of the time embedding by just increasing g. As it was explained in the text, the time embeddings play an important role in increasing the observability of the system. This way, it would be interesting to investigate how observability could be controlled by the coupling factor g. This could give an additional importance to this factor and perhaps lead to improvements in the method.
- Further tests changing parameters in the EWPF like the root of α_i or the choice of γ_U , γ_N or ϵ, related to the mixture density, and the number of particles retained would be useful to understand how these changes would affect the performance of the system.
- Further experiments with the barotropic vorticity model can be performed to test different values for the constants g, $gtau$, time embeddings D_d or the ensemble size used. Due to lack of time, the experiments did not focus on an optimal configuration and so it is expected that better results can still be found for this model.
- Computational improvements are still needed in the ensemble synchronisation scheme, as the calculation of the truncated SVD for the pseudoinverse in the synchronisation coupling term is probably making the scheme computationally expensive. Alternative routines and/or techniques are available and can be used to optimize the scheme.
- Results for the barotropic vorticity model have pointed out some deficiencies of the IEWPF scheme. These may be related to the way some of its parameters are tuned, as these are the first experiments of this particle filter using the barotropic vorticity model. Another reason for these issues could be that the approximations made while deriving the IEWPF still need improvements. Investigation of both arguments would actually be an interesting endeavour towards the improvement of this non-degenerate particle filter.
- Finally, a future work to be included in this system is the implementation of a backward-forward time embedding in the ensemble synchronisation scheme, in order to compare its performance with the forward one, used here. Additionally, in the backward-forward version of the time embedding, the disruptions generated by the particle filter steps in the synchronisation process are expected to be less severe, which could potentially improve the results found in this work.

This is just a summarized list of the interesting research that the combination between synchronisation and a particle filter could generate. It is clear that it represents a considerable amount of work, but is indeed a very intriguing and exciting area for further research.

To conclude this section on a look towards the future and bringing our attention to the operational data assimilation applied to the numerical weather prediction, it is noticeable that the hybrid methods are increasingly popular in the major operational centres. These methodologies aim to combine variational methods and the EnKF. The idea is to take the ensemble information on the error in the background model state and incorporate it in a variational analysis or even use an ensemble of 4D-Var trajectories to compute the background error covariance. The probabilistic meaning of the results produced by these hybrid methods still needs further exploration. Particle filters like the ones used in this work can be a future alternative as a hybrid fully nonlinear data assimilation scheme, while providing promising results to overcome the challenges of predicting the behavior of our chaotic, phenomenal nature.

Appendix A:

SYNCHRONISATION

This appendix contains the pseudocode of the synchronisation scheme, described in Chapter 3.

Algorithm 1 Synchronisation scheme

```
$S \leftarrow H * x$
$Y \leftarrow y$
$dsdx \leftarrow H$
for $m \leftarrow 2$ to $D_d$ do
    for $i \leftarrow 1$ to $tau$ do
        Propagate $x$ using model
    end for
    $S \leftarrow H * x$ {append to the existing $S$}
    $Y \leftarrow y^i$ {append to the existing $Y$}
    $dsdx \leftarrow H * x$ {append to the existing dsdx}
end for
$dxds \leftarrow$ inverse of truncated SVD of dsdx
$x^{j+1} \leftarrow f(x^j) + g * dxds * (Y - S)$
```

Appendix B:

ENSEMBLE-BASED SYNCHRONISATION

This appendix contains the pseudocode for the ensemble-based synchronisation scheme, described in Chapter 4.

Algorithm 2 Ensemble-based Synchronisation scheme

```
$Ens \leftarrow$ initial ensemble
$E_A \leftarrow$ ensemble mean
$A \leftarrow Ens - E_A$
$S \leftarrow H * E_A$
$Y \leftarrow y$
$dsdx \leftarrow H$
for $m \leftarrow 2$ to $D_d$ do
  for $i \leftarrow 1$ to $tau$ do
    $Ens \leftarrow$ ensemble propagated
  end for
  $E_B \leftarrow$ ensemble mean
  $B \leftarrow Ens - E_B$
  $S \leftarrow H * E_B$ {append to the existing $S$}
  $Y \leftarrow y^i$ {append to the existing $Y$}
  $dsdx \leftarrow H * B$ {append to the existing dsdx}
end for
$dxds \leftarrow$ inverse of truncated SVD of dsdx
$x^{j+1} \leftarrow f(x^j) + g * (A * dxds) * (Y - S)$
```

Appendix C:

IEWPF USING ENSYNCH AS PROPOSAL

This appendix contains the pseudocode for the IEWPF using EnSynch as a proposal density, described in Chapter 5.

Algorithm 3 EnSynch (observed time steps)

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

Algorithm 4 EnSynch (unobserved time steps)

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

Algorithm 5 IEWPF time steps

```
for $i \leftarrow 1$ to $N_{ens}$ do
    $\boldsymbol{d_i} \leftarrow \boldsymbol{y} - \boldsymbol{H}(\boldsymbol{f}(\boldsymbol{x^t}))$
    $\boldsymbol{b_i} \leftarrow (\boldsymbol{HQH^T} + \boldsymbol{R})^{-1} * \boldsymbol{d_i}$
    $\boldsymbol{\Phi_i} \leftarrow \boldsymbol{d_i^T} * \boldsymbol{b_i}$
    $\boldsymbol{c_i} \leftarrow 2 * w_{particle} + \boldsymbol{\Phi_i}$
end for
$w_{target} \leftarrow max(\boldsymbol{c})$
$\boldsymbol{P} = (\boldsymbol{Q}^{-1} + \boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{H})^{-1}$
$\boldsymbol{\eta}_i \sim N(0, \boldsymbol{P})$
for $i \leftarrow 1$ to $N_{ens}$ do
    $\boldsymbol{K}_i \leftarrow \boldsymbol{QH}^T\boldsymbol{b}_i$
    $\boldsymbol{x}_i^a \leftarrow \boldsymbol{f}(\boldsymbol{x^t}) + \boldsymbol{K}_i$
    $\Gamma_i \leftarrow \boldsymbol{\eta_i^T}\boldsymbol{\eta_i}$
    $\boldsymbol{a_i} \leftarrow \boldsymbol{\Phi_i} - w_{particle} + w_{target}$
    Solve $(\alpha_i - 1)\Gamma_i - N_x log(\alpha_i) + \boldsymbol{a_i} = 0$
    $(\boldsymbol{Ens})_i \leftarrow \boldsymbol{x}_i^a + \alpha_i \eta_i$
end for
```

Appendix D:

EWPF PROPOSAL

This appendix contains the pseudocode for the EWPF step, described in Chapter 2.

Algorithm 6 EWPF time steps

```
Define percentage of particles $\rho$
$N_k \leftarrow \rho N_{ens}$
for $i \leftarrow 1$ to $N_{ens}$ do
    $\boldsymbol{d_i} \leftarrow \boldsymbol{y} - \boldsymbol{H}(\boldsymbol{f}(\boldsymbol{x_i^{n-1}}))$
    $\boldsymbol{b_i} \leftarrow (\boldsymbol{HQH^T} + \boldsymbol{R})^{-1} * \boldsymbol{d_i}$
    $\boldsymbol{\Phi_i} \leftarrow \boldsymbol{d_i^T} * \boldsymbol{b_i}$
    $\boldsymbol{c_i} \leftarrow w_{particle} + (1/2)\boldsymbol{\Phi_i}$
end for
Sort max weights with $i_c$ sorted indices:
$(\hat{\boldsymbol{c}}, \boldsymbol{i_c}) \leftarrow sort(\boldsymbol{c})$
$c_{max} \leftarrow \hat{\boldsymbol{c}}(\boldsymbol{N_k})$
for $j \leftarrow 1$ to $N_K$ do
    $i \leftarrow \boldsymbol{i_c}(j)$
    $\boldsymbol{K}_i \leftarrow \boldsymbol{QH}^T \boldsymbol{b}_i$
    $a_i \leftarrow \frac{1}{2}\boldsymbol{d}_i^T \boldsymbol{R}^{-1} \boldsymbol{HK}_i$
    $r_i \leftarrow \boldsymbol{d}_i^T \boldsymbol{R}^{-1} \boldsymbol{d}_i$
    $b_i \leftarrow \frac{1}{2}r_i - c_{max} + w_{particle}(i)$
    $\alpha_i \leftarrow 1 + \sqrt{1 - b_i/a_i}$
    $\boldsymbol{\beta} \sim (1-\epsilon)\boldsymbol{Q}^{1/2}\mathcal{U}(-\gamma_U \boldsymbol{I}, +\gamma_U \boldsymbol{I}) + \epsilon\mathcal{N}\left(\gamma_N^2 \boldsymbol{Q}\right)$
    $(\boldsymbol{Ens})_i \leftarrow \boldsymbol{f}(\boldsymbol{x_i^{n-1}}) + \alpha_i \boldsymbol{K}_i + \boldsymbol{\beta}$
    if $\boldsymbol{\beta} \leftarrow \mathcal{U}$ distribution then
        $\boldsymbol{w}_j \leftarrow \boldsymbol{w}_{particle}(i) + (\alpha_i^2 - 2\alpha_i)a_i + \frac{1}{2}r_i$
    else
        $\boldsymbol{w}_1 \leftarrow \boldsymbol{w}_{particle}(i) + (\alpha_i^2 - 2\alpha_i)a_i$
        $\boldsymbol{w}_2 \leftarrow \boldsymbol{w}_1 + \frac{1}{2}r_i + \left(2^{-N_x/2}\right)\left(\pi^{N_x/2}\right)$
        $\boldsymbol{w}_3 \leftarrow \boldsymbol{w}_2 \gamma_N \gamma_U^{-N_x}(\frac{1-\epsilon}{\epsilon})$
        $\boldsymbol{w}_j \leftarrow \boldsymbol{w}_3 \exp\left(\frac{1}{2}\boldsymbol{\beta}_i^2\right)$
    end if
end for
Resampling (Stochastic Universal Resampling used here:)
$\hat{w}_1 \leftarrow w_1$
for $i \leftarrow 2$ to $N_{ens}$ do
    $\hat{w}_i = \sum_{j=1}^{i} w_j$
end for
$u \sim \mathcal{U}[0, 1/N_{ens}]$
$a \leftarrow 1$
$N_{Gens} \leftarrow$ number of particles to be generated
for $i \leftarrow 2$ to $N_{Gens}$ do
    while u $> \hat{w}_a$ do
        $a \leftarrow a + 1$
    end while
    $idx_i \leftarrow a$
    $u \leftarrow u + 1/N_{ens}$
    $a \leftarrow 1$
end for
return idx
$w \leftarrow 1/N_{ens}$
```

REFERENCES

Abarbanel, H. D. I., 2013: *Predicting the Future - Completing Models of Observed Complex Systems*. Springer, New York, USA, 238 pp.

Abarbanel, H. D. I., T. A. Carroll, L. M. Pecora, J. J. Sidorowich, and L. S. Tsimring, 1994: Predicting physical variables in time-delay embedding. *Phys Rev E*, **49**, 1840–1853, doi:10.1103/PhysRevE.49.1840.

Ades, M., 2013: *Data Assimilation in highly nonlinear systems*. Ph.D. thesis, University of Reading.

Ades, M. and P. J. van Leeuwen, 2012: An exploration of the equivalent weights particle filter. *Quart. J. Roy. Meteorol. Soc.*, **139**, 820–840, doi:10.1002/qj.1995.

— 2014: The Equivalent-weights Particle Filter in a high-dimensional system. *Quart. J. Roy. Meteorol. Soc.*, **141**, 484–503, doi:10.1002/qj.2370.

Anderson, B. D. O. and J. B. Moore, 1979: *Optimal Filtering*. Prentice-Hall, Inc., New Jersey, 357 pp.

Asch, M., M. Bocquet, and M. Nodet, 2016: *Data Assimilation: Methods, Algorithms, and Applications*. Fundamental of Algorithms, SIAM, Philadelphia, USA, 306 pp.

Berliner, L. M. and C. K. Wikle, 2007: Approximate importance sampling Monte Carlo for data assimilation. *Physica D*, **230**, 37–49.

Bocquet, M. and P. Sakov, 2014: An iterative ensemble Kalman smoother. *Quart. J. Roy. Meteorol. Soc.*, **140**, 1521–1535, doi:10.1002/qj.2236.

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, doi:10.3402/tellusa.v68.30466.

Browne, P. A. and P. J. van Leeuwen, 2015: Twin experiments with the equivalent weights particle filter and HadCM3. *Quart. J. Roy. Meteorol. Soc.*, **141**, 3399–3414, doi:10.1002/qj.2621.

Browne, P. A. and S. Wilson, 2015: A simple method for integrating a complex model into an ensemble data assimilation system using MPI. *Environ. Model. Softw.*, **68**, 122–128, doi:10.1016/j.envsoft.2015.02.003.

Chen, M. and J. Kurths, 2007: Synchronization of time-delayed systems. *pr*, **76**, 036212, doi:10.1103/PhysRevE.76.036212.

Chorin, A. J., M. Morzfeld, and X. Tu, 2010: Implicit particle filters for data assimilation. *Communications in Applied Mathematics and Computational Science*, **5**, 221–240.

Chorin, A. J. and X. Tu, 2009: Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, volume 106, 17249–17254.

Cohn, S. E., N. S. Sivakumaran, and R. Todling, 1994: A fixed-lag Kalman smoother for retrospective data assimilation. *Mon. Wea. Rev.*, **122**, 2838–2867.

Cosme, E., J. Brankart, J. Verron, P. Brasseur, and M. Krysta, 2010: Implementation of a reduced-rank, square-root smoother for ocean data assimilation. *Ocean Modell.*, **33**, 87100.

Cosme, E., J. Verron, P. Brasseur, J. Blum, and D. Auroux, 2012: Smoothing problems in a Bayesian framework and their linear Gaussian solutions. *Mon. Wea. Rev.*, **140**, 683–695, doi:10.1175/MWR-D-10-05025.1.

Courtier, P. and O. Talagrand, 1987: Variational assimilation of meteorological observations with the adjoint vorticity equation. ii: Numerical results. *Quart. J. Roy. Meteorol. Soc.*, **113**, 1329–1347.

Daley, R., 1991: *Atmospheric Data Analysis*. Cambridge University Press, Cambridge, UK.

Dhamala, M., V. K. Jirsa, and M. Ding, 2004: Enhancement of neural synchrony by time delay. *Phys. Rev. Lett.*, **92**, 074–104, doi:10.1103/PhysRevLett.92.074104.

Doucet, A., N. De Freitas, and N. J. Gordon, 2001: *Sequential Monte Carlo Methods in Practice*. Springer, New York, USA, 1st edition.

Doucet, A., S. Godsill, and C. Andrieu, 2000: On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, **10**, 197–208.

Duane, G. S., 2015: Synchronicity from synchronized chaos. *Entropy*, **17**, 1701–1733, doi:10.3390/e17041701.

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

Evensen, G., 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.*, **99**, 10143–10162, doi:10.1029/94JC00572.

— 2003: The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, **53**, 343–367.

Evensen, G. and P. J. van Leeuwen, 2000: An ensemble kalman smoother for nonlinear dynamics. *Mon. Wea. Rev.*, **128**, 1852–1867, doi:10.1175/1520-0493.

Eyink, G. and S. Kim, 2006: A maximum entropy method for particle filtering. *Journal of statistical physics*, **123**, 1071–1128.

Fairbairn, D., S. Pring, A. C. Lorenc, and I. Roulstone, 2014: A comparison of 4DVar with ensemble data assimilation methods. *Quart. J. Roy. Meteorol. Soc.*, **140**, 281–294, doi:10.1002/qj.2135.

Feng, C., Y. Zhang, and Y.-H. Wang, 2007: Different types of synchronization in time-delayed systems. *Chinese Physics Letters*, **24**, 50.

Fletcher, S. J., 2017: *Data Assimilation for the Geosciences*. Elsevier, 976 pp.

Fujisaka, H. and T. Yamada, 1983: Stability Theory of Synchronized Motion in Coupled-Oscillator Systems. *Prog. Theor. Phys.*, **69**, 32–47, doi:10.1143/PTP.69.32.

Garthwaite, P., I. Jolliffe, and B. Jones, 2002: *Statistical Inference*. Oxford University Press, 328 pp.

Gauthier, P., 1992: Chaos and quadri-dimensional data assimilation: a study based on the Lorenz model. *Tellus A*, **44**, 217.

Gelb, A., 1974: *Applied Optimal Estimation*. M.I.T Press, Cambridge, UK.

Ghosh, D., I. Grosu, and S. K. Dana, 2012: Design of coupling for synchronization in time-delayed systems. *Chaos*, **22**, 033111, doi:10.1063/1.4731797.

Golub, G. H. and C. F. Van Loan, 1996: *Matrix Computations*. The Johns Hopkins University Press, 3rd edition.

Gordon, N. J., D. J. Salmond, and A. F. Smith, 1993: Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE proceedings*, volume 140, 107–113.

Guan, S., K. Li, and C. H. Lai, 2006: Chaotic synchronization through coupling strategies. *Chaos*, **16**, 023107, doi:10.1063/1.2193684.

Gustafsson, N. and J. Bojarova, 2014: Four-dimensional ensemble variational (4D-En-Var) data assimilation for the HIgh-Resolution Limited Area Model (HIRLAM). *Nonlinear Process Geophys.*, **21**, 745–762, doi:10.5194/npg-21-745-2014.

Hogg, R. V., J. McKean, and A. T. Craig, 2013: *Introduction to Mathematical Statistics*. Pearson, Boston, USA.

Houtekamer, P. and H. L. Mitchell, 1998: Data Assimilation Using an Ensemble Kalman Filter Technique. *Mon. Wea. Rev.*, **126**, 796–811, doi:10.1175/1520-0493.

Huang, T., G. Chen, and J. Kurths, 2011: Synchronization of chaotic systems with time-varying coupling delays. *Discrete and Continuous Dynamical Systems - B*, **16**, 1071–1082, doi:10.3934/dcdsb.2011.16.1071.

Johnson, G. A., D. J. Mar, T. L. Carroll, and L. M. Pecora, 1998: Synchronization and imposed bifurcations in the presence of large parameter mismatch. *Phys. Rev. Lett.*, **80**, 3956–3959, doi:10.1103/PhysRevLett.80.3956.

Junge, L. and U. Parlitz, 2001: Synchronization using dynamic coupling. *Phys Rev E*, **64**, 055204, doi:10.1103/PhysRevE.64.055204.

Kalman, R. E., 1960: A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering Series D*, volume 82, 35–45.

Kantz, H. and T. Schreiber, 1997: *Nonlinear time series analysis*. Cambridge University Press.

Kim, S., G. L. Eyink, J. M. Restrepo, F. J. Alexander, and G. Johnson, 2003: Ensemble filtering for nonlinear dynamics. *Mon. Wea. Rev.*, **131**, 2586–2594.

Kitagawa, G., 1996: Monte-Carlo filter and smoother for non-Gaussian non-linear state-space models. *Journal of Computational and Graphical Statistics*, **10**, 253–259.

Klaas, M., N. de Freitas, and A. Doucet, 2005: Towards practical n2 monte carlo: The marginal particle filter. *Proceedings of the 21st Annual conference on uncertainty in Artificial Intelligence (UAI-05)*, AUAI Press, Arlington, VA, 308–315.

Kloeden, P. and E. Platen, 1992: *Numerical solution of stochastic differential equations*. Springer, Berlin.

Le Dimet, F.-X. and O. Talagrand, 1986: Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, **38**, 97–110.

Lermusiaux, P. F. J., P. J. H. Haley, and W. G. Leslie, 2002: Advanced interdisciplinary data assimilation: Filtering and smoothing via Error Subspace Statistical Estimation. *OCEANS '02 MTS/IEEE*, volume 2, 795–802.

Lermusiaux, P. F. J. and A. R. Robinson, 1999: Data assimilation via error subspace statistical estimation. Part I: Theory and schemes. *Mon. Wea. Rev.*, **127**, 1385–1407.

Letellier, C., L. A. Aguirre, and J. Maquet, 2005: Relation between observability and differential embeddings for nonlinear dynamics. *Phys Rev E*, **71**, 066213, doi:10.1103/PhysRevE.71.066213.

Liu, C., Q. Xiao, and B. Wang, 2008: An ensemble-based four-dimensional variational data assimilation scheme. Part I: Technical formulation and preliminary test. *Mon. Wea. Rev.*, **136**, 3363–3373, doi:10.1175/2008MWR2312.1.

Lorenc, A., 1986: Analysis methods for numerical weather prediction. *Quart. J. Roy. Meteorol. Soc.*, **112**, 1177–1194.

Lorenz, E. N., 1963: Deterministic nonperiodic flow. *J. Atmos. Sci.*, **20**, 130–141.

— 1969: The predictability of a flow which possesses many scales of motion. *Tellus*, **21**, 289–307, doi:10.1111/j.2153-3490.1969.tb00444.x.

— 1995: Predictability - a problem partly solved. *Proc. of the Seminar on Predictability, vol.1*, ECMWF, Reading, UK, 1–18.

Lui, J. S. and R. Chen, 1998: Sequential Monte-Carlo methods for dynamical systems. *Journal of the American Statistical Association*, **90**, 567–576.

Mènard, R. and R. Daley, 1996: The application of Kalman smoother theory to the estimation of 4DVAR error statistics. *Tellus A*, **48**, 221–237.

Miller, R. N., M. Ghil, and F. Gauthiez, 1994: Advanced data assimilation in strongly nonlinear dynamical systems. *J. Atmos. Sci.*, **51**, 1037–1056.

Morzfeld, M., X. Tu, E. Atkins, and A. Chorin, 2012: A random map implementation of implicit filters. *Journal of Computational Physics*, **231**, 2049–2066.

Nakano, S., G. Ueno, and T. Higuchi, 2007: Merging particle filter for sequential data assimilation. *Nonlinear Process Geophys.*, **14**, 395–408.

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

Parlitz, U., J. Schumann-Bischoff, and S. Luther, 2014: Local observability of state variables and parameters in nonlinear modeling quantified by delay reconstruction. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **24**, 024411, doi:10.1063/1.4884344.

Pazó, D., A. Carrassi, and J. M. López, 2016: Data assimilation by delay-coordinate nudging. *Quart. J. Roy. Meteorol. Soc.*, **142**, 1290–1299, doi:10.1002/qj.2732.

Pecora, L. M. and T. L. Carroll, 1990: Synchronization in chaotic systems. *Phys. Rev. Lett.*, **64**, 821–825, doi:10.1103/PhysRevLett.64.821.

Peña-Ramirez, J., L. Olvera, and H. Nijmeijer, 2016: The sympathy of two pendulum clocks: beyond Huygens' observations. *Sci. Rep.*, **6**, doi:10.1038/srep23580.

Pikovsky, A., M. Rosenblum, and J. Kurths, 2001: *Synchronization:A universal concept in nonlinear sciences*. Cambridge University Press, Cambridge, UK, 1st edition.

Pikovsky, A. S., 1984: On the interaction of strange attractors. *Z. Phys. B.*, **55**, 149–154.

Pinheiro, F. R., P. J. van Leeuwen, and U. Parlitz, 2018: An ensemble framework for time delay synchronization. *Quart. J. Roy. Meteorol. Soc.*, **144**, 305–316, doi:10.1002/qj.3204.

Rey, D., M. Eldridge, M. Kostuk, H. D. I. Abarbanel, J. Schummann-Bischoff, and U. Parlitz, 2014a: Accurate state and parameter estimation in nonlinear systems with sparse observations. *Phys. Lett. A*, **378**, 869–873, doi:10.1016/j.physleta.2014.01.027.

Rey, D., M. Eldridge, U. Morone, H. D. I. Abarbanel, U. Parlitz, and J. Schummann-Bischoff, 2014b: Using waveform information in nonlinear data assimilation. *Phys. Rev. E*, **90**, 062916, doi:10.1103/PhysRevE.90.062916.

Ross, S. M., 2014: *Introduction to Probability and Statistics for Engineers and Scientists*. Elsevier Academic Press, 5th edition, 680 pp.

Snyder, C., T. Bengtsson, P. Bickel, and J. Anderson, 2008: Obstacles to highdimensional particle filtering. *Mon. Wea. Rev.*, **136**, 4629–4640, doi:10.1175/2008MWR2529.1.

Snyder, C., T. Bengtsson, and M. Morzfeld, 2015: Performance Bounds for Particle Filters Using the Optimal Proposal. *Mon. Wea. Rev.*, **143**, 4750–4761, doi:10.1175/MWR-D-15-0144.1.

Spiller, E. T., A. Budhiraja, K. Ide, and C. K. Jones, 2008: Modified particle filter methods for assimilating Lagrangian data into a point-vortex model. *Physica D*, **237**, 1498–1506.

Stojanovski, T., L. Kocarev, U. Parlitz, and R. Harris, 1996: How to achieve synchronised motion with a finite number of samples. *Proc. NOLTA'96*, Kochi, Japan, 83–86.

Talagrand, O., 2012: 4D-Var: Four-Dimensional variational assimilation. *Advanced Data Assimilation for Geosciences: Lecture Notes of the Les Houches School of Physics*, E. Blayo, M. Bocquet, E. Cosme, and F. L. Cugliandolo, eds., Oxford University Press, Oxford, chapter 1, 3–30.

Talagrand, O. and P. Courtier, 1987: Variational assimilation of meteorological observations with the adjoint vorticity equation. i: theory. *Quart. J. Roy. Meteorol. Soc.*, **113**, 1311–1328.

Tremolet, Y., 2006: Accounting for an imperfect model in 4D-var. *Quart. J. Roy. Meteorol. Soc.*, **132**, 2483–2504.

Tribbia, J. and D. Baumhefner, 2004: Scale interactions and atmospheric predictability: an updated perspective. *Mon. Wea. Rev.*, **132**, 703–713.

van Leeuwen, P. J., 1999: The time-mean circulation in the agulhas region determined with the ensemble smoother. *J. Geophys. Res.*, **104**, 1393–1404, doi:10.1029/1998JC900012.

— 2002: Ensemble kalman filters: Sequential importance resampling and beyond. *ECMWF Workshop on the Role of the Upper Ocean in Medium and Extended Range Forecasting*, ECMWF, Reading, UK, 46–56.

— 2009: Particle filtering in geophysical systems. *Mon. Wea. Rev.*, **137**, 4089–4114, doi:10.1175/2009MWR2835.1.

— 2010: Nonlinear data assimilation in Geosciences: an extremely efficient particle filter. *Quart. J. Roy. Meteorol. Soc.*, **136**, 1991–1999, doi:10.1002/qj.699.

van Leeuwen, P. J., Y. Cheng, and S. Reich, 2015: *Nonlinear data assimilation*. Frontiers in Applied Dynamical Systems: Reviews and Tutorials, Springer International Publishing, Switzerland, 118 pp.

van Leeuwen, P. J. and G. Evensen, 1996: Data assimilation and inverse methods in terms of a probabilistic formulation. *Mon. Wea. Rev.*, **124**, 2898–2913, doi:10.1175/1520-0493.

Wikle, C. K. and L. M. Berliner, 2007: A bayesian tutorial for data assimilation. *Physica D*, **230**, 116.

Xiong, X., I. M. Navon, and B. Uzunoglu, 2006: A note on the particle filter with posterior Gaussian resampling. *Tellus A*, **58**, 456–460, doi:10.1111/j.1600-0870.2006.00185.x.

Yang, S. C., D. Baker, H. Li, K. Cordes, M. Huff, G. Nagpal, E. Okereke, J. Villafane, E. Kalnay, and G. S. Duane, 2006: Data Assimilation as Synchronization of Truth and Model: Experiments with the Three-Variable Lorenz System. *J. Atmos. Sci.*, **63**, 2340–2354, doi:10.1175/JAS3739.1.

Zhan, M., X. Wang, X. Gong, G. W. Wei, and C. H. Lai, 2003: Complete synchronization and generalized synchronization of one-way coupled time-delay systems. *Phys Rev E*, **68**, 036208, doi:10.1103/PhysRevE.68.036208.

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.