University of Reading

School of Mathematics, Meteorology and Physics

Comparison of the Ensemble Transform Kalman Filter with the Ensemble Transform Kalman Smoother

David Fairbairn

August 2009

This dissertation is submitted to the Department of Mathematics in partial fulfilment of the requirements for the degree of Master of Science

Abstract

Data assimilation aims to produce initial conditions for a weather forecast that are as close as possible to reality. The Ensemble Kalman Filter (EnKF) is a data assimilation method that forecasts a statistical sample of state estimates using a linear or nonlinear model. This forecast state is then updated with observations according to given errors in the observations and the forecast state. This update is the best estimate of the state of the system, called the analysis. Various formulations of the EnKF are discussed, which differ in the analysis step. The ETKF has been chosen for implementation.

Smoothing differs from filtering in that future observations are assimilated in the analysis step. The smoother should therefore have smaller analysis errors than the filter inside the time-window. The Ensemble Transform Kalman Smoother (ETKS) is derived and two different variations are given, the ETKS and the ETKS2. The ETKS2 is an approximation of the ETKS.

The ETKF and ETKS2 are implemented using a swinging spring model and experimental results are compared. The experiments differ in the ensemble size, the ETKS2 lag and whether the observations are perfect (noise-free) or imperfect. It is discovered that the ETKS2 is always more accurate than the ETKF and both solutions are most accurate with perfect observations. It is also discovered that higher lags generally improve the ETKS2 accuracy, although this is only tested for the imperfect case. Increasing the ensemble size up to the model dimension greatly improves the accuracy of the solution for the ETKF and ETKS2, although this is also only tested for the imperfect case. An obvious reason is there are smaller sampling errors for larger ensemble sizes. However, increasing the ensemble size above the model dimension generally worsens the solution for both. This may have been causing a poor selection of vectors from the basis for the space in which the state estimate lies to form the new analysis state estimate, thus increasing the ensemble mean analysis error.

Acknowledgements

Firstly, I would like to thank my supervisor Dr Sarah Dance for her great suggestions and enthusiasm on the project. I am also grateful to Professor Nancy Nichols for her help.

I would like to thank the students on the MSc course for helping me to enjoy this year, which was my best at Reading university. The staff were also supportive and I would like to thank Dr Pete Sweby in particular, for letting me on the course and his hard work in organising the course. Finally, I would like to thank EPSRC for their financial support.

Contents

List of Symbols

Matrices and vectors

h	Observation operator
H	Linear Observation operator
I	Identity matrix
K	Kalman gain matrix
M	Linear dynamical model
m	Non-linear dynamical model
P	State error covariance matrix
Q	Model error covariance matrix
R	Observation error covariance matrix
R_e	Observation error covariance matrix for perturbed observations
S	Ensemble version of innovations covariance matrix
T	Post-multiplier in deterministic formulations of the EnKF
U, V, W	Orthogonal matrices
$\underline{x}$	State vector
X	Ensemble perturbation matrix
$\underline{y}$	Observation vector
Y	Observation ensemble perturbation matrix
$\underline{d}$	Observation vector for perturbed observations (ETKF)
D	Observation ensemble perturbation matrix for the ETKF
$\underline{z}$	Arbitrary vector
Z	Arbitrary matrix
ϵ	Random observation error vector
η	Random model error vector
Λ	Diagonal matrix of eigenvalues
Σ	Matrix of singular values

Sub and superscripts

f	Forecast value
a	Analysis value
e	Ensemble value (unless otherwise stated)
i	Ensemble Member
j	Time index
t	True value
$\overline{100}$	Average over 100 runs

Numbers of components

n	Number of state components
p	Number of observation components
N	Number of ensemble members
I	Number of time-steps in time-interval

Operators

σ	Standard deviation
$< z >$	Expectation of value
$\overline{z}$	Mean value
z'	Deviation from the mean
$\hat{z}$	Scaled value

Abbreviations

KF	Kalman Filter
EnKF	Ensemble Kalman Filter
SRF	Square Root Filter
EnSRF	Ensemble Square Root Filter
ETKF	Ensemble Transform Kalman Filter
KS	Kalman Smoother
SRS	Square Root Smoother
EnSRS	Ensemble Square Root Smoother
ETKS	Ensemble Transform Kalman Smoother
ETKS2	Approximation of Ensemble Transform Kalman Smoother
NWP	Numerical Weather Prediction
NMI	Normal Mode Initialization
LNMI	Linear Normal Mode Initialization
NNMI	Non-linear Normal Mode Initialization
SVD	Singular Value Decomposition

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is data assimilation?

Data assimilation incorporates observational data into a numerical model to produce a model state that aims to be as close as possible to reality. In order to produce a weather forecast initial conditions are required in the numerical model. Numerical weather prediction (NWP) would not function properly by just inserting measured observations into a model. One main reason is that there are too few observations to determine the state of the system everywhere. Some regions are data rich such as Eurasia and North America, but others are poorly observed (Kalnay, 2003). In modern NWP the number of degrees of freedom of a model is of $O(10^7)$. But the number of conventional observations that can be inserted directly into a model is $O(10^4)$ (Kalnay, 2003). Many types of data such as satellite and radar observations are indirect, meaning they cannot be simply inserted into a model. In NWP it is necessary to have a complete first guess estimate of the state of the atmosphere at all the grid points in order to generate the initial conditions for the forecasts (Bergthorsson and Doos, 1955). This is done by updating a prior estimate of the state of the atmosphere (Background state) with current observations to produce a best estimate of the state of the system (Analysis state) (Kalnay, 2003).

Data assimilation methods are either sequential, variational or both. Sequential refers to the time behaviour of the scheme. This means that the analysis is solved by updating a forecast by assimilating available observations at each time-step in a time-window (Bannister, 2007). An example is the Kalman filter (Kalman, 1960). Variational methods mean that iterations are used to minimize a cost function. The cost function is a function of the model state and is a measure of the difference between the background state and the observations. The model state that minimizes the cost function is used as the analysis state

(Bannister, 2007). An example of a variational method is 4D-Var (Dimet and Talagrand, 1986). Variational methods are most commonly used in operational NWP but are not the subject of this report.

1.2 Background

This report focuses on a sequential form of data assimilation called the Ensemble Transform Kalman smoother (Evensen and van Leeuwen, 2000). This is one of the more recent of a series of modifications of the Kalman filter (KF), introduced by Kalman [1960]. The KF uses an initial single analysis state estimate. This is forecasted to the next time-step using a linear model. Then the forecast is updated by assimilating the observations at that time, to produce the analysis state. A weighting is given to the observations by a Kalman gain matrix, according to a ratio between estimated errors in the forecast and estimated errors in the observations. The estimate of the errors in the forecast and analysis states are also updated at each time-step using the forecast model and the Kalman gain matrix. This step by step process is repeated until the latest analysis step reaches the end of the time window. The KF is not used in operational NWP since it cannot process non-linear models and is computationally very expensive (Burgers et al., 1997). More detail on the KF and the KF equations are given in section 2.2. The Ensemble Kalman Filter (EnKF, section 2.3), Ensemble Square Root filter (EnSRF, section 2.4) and the Ensemble Transform Kalman Filter (ETKF, section 2.5) all use an ensemble of state estimates. Each ensemble member is forecast individually. A ratio between the ensemble forecast covariances and the observation covariances are used in the Kalman gain to assimilate the observations in the analysis stage. The ensemble mean analysis state and the analysis ensemble perturbation matrix are calculated from this single gain matrix. The ensemble analysis can then be calculated from the mean and the perturbations. Although there is added expense in updating an ensemble of state estimates rather than just one, overall they are more computationally efficient than the KF since you only have to store each ensemble member, not the full error covariance matrix. They can also use non-linear forecast models and are more suited to parallel processing. These features make them more desirable for operational NWP.

Smoothing differs from filtering in that future observations are assimilated in the analysis step (Ravela and McLaughlin, 2007). This would imply that the smoothed analysis would be more accurate than the filtered analysis since the smoothed analysis is assimilating extra observations (Ravela and McLaughlin, 2007). Smoothing can be divided into two types, fixed-interval and fixed-lag. Fixed-interval smoothing uses all the observations in a time interval to update all the analysis states. Fixed-lag smoothing only updates a fixed number of states prior to the current observation time, which saves computational cost (Cohn et al., 1994). The fixed-lag Ensemble Transform Kalman Smoother 2 (ETKS2, see section 2.9.2) introduced in this report is based on the linear smoothing method of Jazwinski [1970] and the fixed-lag method of Cohn et al. [1994]. The implementation is similar to the implementation of the ETKF by Livings [2005].

A schematic showing the comparison between a filter and a fixed-interval smoother is shown in Figure 1.1. In the schematic the smoothed solution (black curve on smoother diagram) is closer to the truth than for the filter (black curve on filter graph). A specific analysis time t_k is highlighted to show that the filter analysis uses only past and present observations while the fixed-interval smoother analysis uses all the observations in the time window at each step. Also the filter analysis curve tends towards the true solution as more observations are assimilated. The fixed-interval smoother analysis curve oscillates at approximately equal distance from the true solution over time, since all the observations are used in the time window at each step. The fixed-lag smoother diagram (Figure 1.2) shows that some of the future observations are used at analysis time t_k. In this case the $lag = 1$ meaning the first closest future observation is assimilated. Thus the solution is less accurate than the fixed-interval smoother but more accurate than the filter. Experiments using smoothing methods such as the fixed-lag Kalman Smoother of Cohn et al. [1994] and the fixed-lag Ensemble Square Root Smoother (see section 2.8) of Whitaker and Compo [2002] have been compared with the KF and EnSRF respectively and show an improvement in accuracy.

The ETKF forecast model implemented by Livings [2005] uses the swinging spring equations with a pseudo-random initial ensemble (see the implementation Chapter 4 for more details). The NNMI solution is used as the true solution. Livings [2005] compared the results for perfect and imperfect observations as well as for different ensemble sizes. The experimental part of this project aims to implement the ETKS2 and experiment with perfect and imperfect observations, as well as different ensemble sizes. The results can then also be compared with the ETKF implementation of Livings [2005].

1.3 Goals for the project

- Implement the fixed-lag ETKS2
- Compare results of the fixed-lag ETKS2 with the ETKF for
  - Perfect/Imperfect observations
  - Different lag lengths

- Different ensemble sizes

1.4 Outline of the report

Chapter 2 describes the Kalman Filter (KF) and different stochastic and deterministic formulations of the KF. The Ensemble Square Root Filter is discussed and the Ensemble Transform Kalman Filter (ETKF) is given as an alternative Square Root Filter. The ETKF is the method used in the experiments. An introduction to smoothing and different smoothing types is given. The Kalman Smoother (KS) is derived, using the linear smoothing method of Jazwinski [1970] and a Bayesian Maximum likelihood approach. The fixed-lag Ensemble Square Root Smoother is given and the ETKS follows on from the KS equations. An approximation to the ETKS called the ETKS2 is introduced as the method to be used in the experiments.

Chapter 3 describes the swinging spring model to be used in the experiments, as well as introducing the concept of initialization.

Chapter 4 gives the implementation of the ETKF and ETKS2 to be used in the experiments, including algorithms. An ETKS implementation is suggested although not used. Results from ETKF and ETKS2 experiments are discussed in chapter 5 and comparisons made. The conclusion in Chapter 6 summarises the report, as well as suggesting some limitations in the experiments and possible future work.

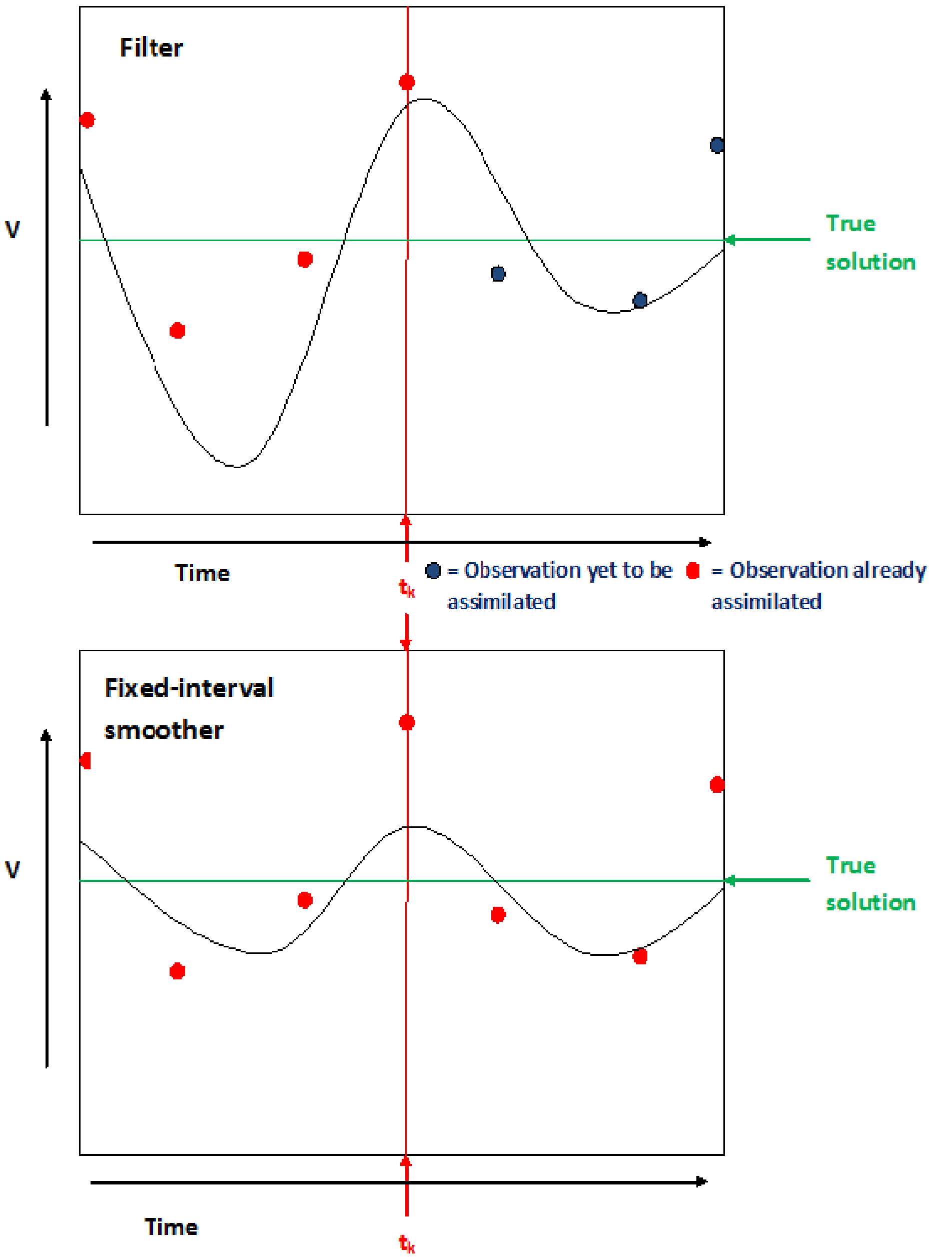
Filter
V
True
solution
Time
t_k
● = Observation yet to be assimilated
● = Observation already assimilated
Fixed-interval
smoother
V
True
solution
Time
t_k

Figure 1.1: Diagram of an analysis curve of a filter and a fixed-interval smoother for the variable to be measured V in a time window. Analysis at time t_k, shown on the red line shows use of just past observations for the filter but all available observations for the fixed-interval smoother.

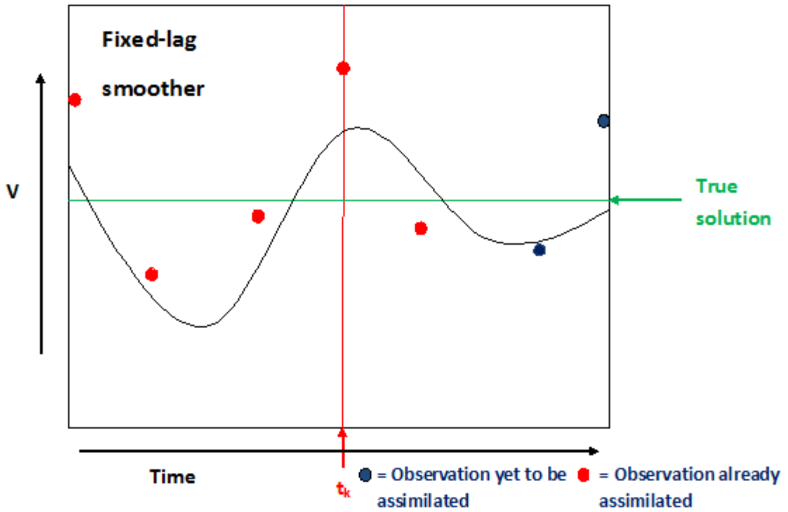
Fixed-lag
smoother
V
True
solution
Time
t_k
● = Observation yet to be assimilated
● = Observation already assimilated

Figure 1.2: Diagram of an analysis curve of a fixed-lag smoother for the variable to be measured V in a time window. Analysis at time t_k, shown on the red line shows use of the past and one future observation

Chapter 2

Formulations of the Kalman Filter

This chapter describes the necessary background on filtering and smoothing methods that is important in order to understand the Ensemble Transform Kalman Filter (ETKF) and Ensemble Transform Kalman Smoother 2 (ETKS2) that are used in the experimental work for this report. The meaning of the notation introduced in each section will remain in effect unless specified otherwise.

2.1 Background information

2.1.1 The forecast model

The forecast model of the system that is being estimated is described here and applies to all formulations of the Kalman Filter (KF). Let n be equal to the state space dimension and p be equal to the observation space dimension. The state vector at time t_k is represented by $\underline{x}_k$. The observation vector at the same time is represented by $\underline{y}_k$. Let the true state of the system $\underline{x}_k^t$ be related to the dynamical model operator m (such that $m : \Re^n \to \Re^n$) by

$$\underline{x}_k^t = m_k(\underline{x}_{k-1}^t) + \eta_{k-1}.$$

Here m_k can be linear or non-linear and η_{k-1} (a vector of dimension n) is the random model error, which is unbiased.

2.1.2 Statistics

Estimating the model errors and the forecast, analysis and observational error covariance matrices are essential to all the formulations of the KF. The random model error η_{k-1} is normally distributed as $N(0, Q_{k-1})$, where Q_{k-1} is the covariance matrix. Q_{k-1} is a measure of the error correlation between different variables in the modelled state ($m_k(\underline{x}^t_{k-1})$) and can be written in the form

$$Q_{k-1} = \left\langle \eta_{k-1}\eta^T_{k-1} \right\rangle , \tag{2.1}$$

where the angle brackets represent the expected value. The normal distibution (or Gaussian distribution) of the error (η) should be symmetric about zero. This means that the error is expected to average to zero and $\langle\eta\rangle = 0$, which means that η is unbiased. The variance is the measure of the spread around the expected value. A narrower distribution of errors around the expected value means a smaller variance. The covariance measures the error correlation between two variables:

$$cov(\eta_1, \eta_2) = \langle(\eta_1 - \langle\eta_1\rangle)(\eta_2 - \langle\eta_2\rangle)\rangle . \tag{2.2}$$

An error covariance matrix relates errors in the state vector components. The diagonal represents the variances and the covariances are the off-diagonal elements of the matrix.

In deriving the Kalman Smoother in section 2.7 it is important to know the following definition of conditional probability. According to Jazwinski [1970] the probability of an event B given that A has occurred is given by

$$P(A|B) = \frac{P(A,B)}{P(B)} \tag{2.3}$$

where $P(A,B)$ is the probability of A and B occurring, or the joint probability.

2.2 The Kalman Filter

The Kalman Filter (KF) is a sequential method of data assimilation introduced by Kalman [1960]. In the Kalman Filter, the analysis $\underline{x}^a_k$ (best estimate of the state of the system) is a single state estimate. The full KF algorithm is reviewed in detail in Jazwinski [1970]. It is important to understand the KF equations in order to understand the ensemble formulations based on the KF, such as the Ensemble Transform Kalman Smoother (ETKS) experiments used in this report.

2.2.1 The observation operator

The state space $\underline{x}_k$ is mapped to observation space $\underline{y}_k$ by the observation operator h_k (such that $h_k : \Re^n \rightarrow \Re^p$). For example, this can be done by interpolating from the model grid to the observation point. In the Kalman filter the observation operator is linear and will be expressed as H. It will be assumed that H does not vary with time. Thus we have,

$$y_k = H\underline{x}_k + \epsilon_k,$$

where ϵ_k (of length p) is the random observational error and is assumed to be unbiased. The observations have an error covariance matrix R_k (of dimension $(p \times p)$). Assuming the observation error does not vary with time then

$$\left\langle \epsilon_k \epsilon_k^T \right\rangle = R. \tag{2.4}$$

in the KF.

2.2.2 Forecast equations

In the Kalman filter the current forecast state $\underline{x}_k^f$ (of length n), is a forecast of the previous analysis $\underline{x}_k^a$ (of n) using the linear model M. For simplicity, it will be assumed that the model is perfect. Firstly an initial analysis state $\underline{x}_0^a$ and an initial analysis error covariance matrix P_0^a (of dimension $n \times n$) are estimated. Then the linear forecast model M is used to produce the forecast at the next time step. The forecast state $\underline{x}_k^f$ is defined as

$$\underline{x}_k^f = M\underline{x}_{k-1}^a. \tag{2.5}$$

The forecast error covariance P_k^f (of dimension $n \times n$) is defined as

$$P_k^f = \left\langle (\underline{x}_k^f - \underline{x}_k^t)(\underline{x}_k^f - \underline{x}_k^t)^T \right\rangle, \tag{2.6}$$

where x_k^t is the true solution. However, the true solution is unknown so P_k^f is given as

$$P_k^f = M_{k-1}P_{k-1}^a M_{k-1}^T + Q_{k-1} \tag{2.7}$$

where Q_{k-1} (of dimension $n \times n$) is the model error covariance matrix and P_{k-1}^a is the error covariance matrix of the previous analysis state.

2.2.3 Kalman gain matrix

The Kalman gain matrix K (of dimension $n \times p$) gives a weighting to the observations according to a ratio between forecast and observational error covariances. This is essential for the analysis step. It is expressed as

$$K_k = P_k^f H^T [HP_k^f H^T + R]^{-1} \tag{2.8}$$

where R is the observational error covariance matrix. This equation for K comes from minimising the following cost function to find the analysis:

$$J(\underline{x}) = (\underline{x} - \underline{x}^f)^T (P^f)^{-1} (\underline{x} - \underline{x}^f) + (\underline{y} - H\underline{x})^T (R)^{-1} (\underline{y} - H\underline{x}). \tag{2.9}$$

The first term gives the distance between the model state $\underline{x}$ and the forecast state $(\underline{x}^f)$ weighted according to the inverse of the forecast error covariance (P^f). The second term gives the distance between the model state and the observations $\underline{y}$ weighted according to the inverse of the observational covariance matrix R. The minimiser of J is the analysis state and satisfies the analysis equation 2.10 (Kalnay, 2003).

2.2.4 Analysis equations

In the analysis step, the Kalman gain is used to assimilate the observations to update the forecast. The analysis $\underline{x}_k^a$ (of length n) is defined as

$$\underline{x}_k^a = \underline{x}_k^f + K_k(\underline{y}_k - H\underline{x}_k^f), \tag{2.10}$$

where y_k of dimension p is the vector of observations. The Kalman gain is also used to update the forecast error covariance to produce the analysis error covariance P_k^a. This is discussed in Burgers et al. [1997].

P_k^a is defined as

$$P_k^a = \left\langle (\underline{x}_k^a - \underline{x}_k^t)(\underline{x}_k^a - \underline{x}_k^t)^T \right\rangle \tag{2.11}$$

Firstly, the analysis (equation 2.10) is substituted into equation 2.11 and $K(H\underline{x}_k^t - \underline{y}_k^t) = 0$ is added to give

$$P_k^a = \left\langle (\underline{x}_k^f - \underline{x}_k^t + K_k(\underline{y}_k - \underline{y}_k^t - H\underline{x}_k^f + H\underline{x}_k^t))(\underline{x}_k^f - \underline{x}_k^t + K_k(\underline{y}_k - \underline{y}_k^t - H\underline{x}_k^f + H\underline{x}_k^t))^T \right\rangle . \tag{2.12}$$

Factorising out $(\underline{x}_k^f - \underline{x}_k^t)(\underline{x}_k^f - \underline{x}_k^t)^T$ and $(\underline{y}_k - \underline{y}_k^t)(\underline{y}_k - \underline{y}_k^t)^T$ gives

$$P_k^a = (I - K_k H) \left\langle (\underline{x}_k^f - \underline{x}_k^t)(\underline{x}_k^f - \underline{x}_k^t)^T \right\rangle (I - K_k H)^T + K_k \left\langle (\underline{y}_k - \underline{y}_k^t)(\underline{y}_k - \underline{y}_k^t)^T \right\rangle K_k^T$$

Note that there are no cross-product terms since it is assumed that $(\underline{y}_k - \underline{y}_k^t)(\underline{x}_k^f - \underline{x}_k^t) = 0$. Substituting in P_k^f and R respectively gives

$$P_k^a = (I - K_k H) P^f (I - H^T K_k^T) + K_k R K_k^T. \tag{2.13}$$

Expanding out the brackets and re-arranging then gives

$$\begin{aligned} P_k^a &= P^f - K_k H P^f - P^f H^T K_k^T + K_k (H P^f H^T + R) K_k^T \\ &= (I - K_k H) P_k^f. \end{aligned} \tag{2.14}$$

2.2.5 Advantages and Disadvantages of the KF

The Kalman filter is a good method in that it produces unbiased analysis and forecast state updates and hence their covariances are exact. Also the filter is optimal since the Kalman gain is produced from minimizing a cost function to find the optimal solution. However, the Kalman filter is not currently used in operational weather forecasting. One reason is that the Kalman filter can only be used for linear systems but NWP models are non-linear. Also, it is very computationally expensive to implement (Ehrendorfer, 1992), mostly due to its propagation of large forecast error covariance matrices using the model. In NWP models, the number of state variables is $O(10^7)$ (UKMO, 2009). Hence the corresponding error covariances will have $O(10^7 \times 10^7)$. Observations of $O(10^6)$ (UKMO, 2009) must then be assimilated at each analysis step. This requires too much storage to be practical. On top of this the Kalman gain requires the inversion of a large matrix, which is expensive and inefficient.

2.2.6 Summary of the KF

To summarise, the Kalman filter uses a linear forecast model to produce a forecast (equation 2.5), which is updated by giving a weighting to the observations according to a ratio between forecast and observational error covariances. This is done using the Kalman gain (equation 2.8). This updated forecast is the analysis (equation 2.10). The forecast covariance matrix is also updated using the linear forecast model (see equation 2.7). The analysis covariance matrix (equation 2.14) comes from updating the forecast covariance matrix using the Kalman gain.

The KF is not currently used in NWP as it is too expensive to implement and can only be used for linear systems.

2.3 The Ensemble Kalman filter (EnKF)

The Ensemble Kalman Filter was introduced by Evensen [1994]. Much of the following developments of the EnKF are discussed in Evensen [2003]. The EnKF uses many of the ideas of the KF but can can be applied to non-linear systems. While the KF uses a single state estimate, the EnKF uses a statistical sample of state estimates, called an ensemble. Like with the KF there is a forecast step and an analysis step. In the forecast step the sample mean and error covariances are calculated from an ensemble of forecast states. These are used to calculate a single Kalman gain matrix. This Kalman gain matrix is used to assimilate the observations in to the forecast ensemble state and produce the analysis ensemble state.

2.3.1 EnKF terms and notation

The following equations 2.15 to 2.19 come from Evensen [2003], but using a new notation. Firstly, assume that an ensemble of state estimates of size N will be used. Therefore $\underline{x}_i$ (of length n) represents the state vector for each ensemble member ($i = 1, ..., N$). The ensemble mean (of length n) is:

$$\overline{\underline{x}} = \frac{1}{N}\sum_{i=1}^{N}\underline{x}_i. \tag{2.15}$$

Define the ensemble error covariance P_e (of dimension $n \times n$) as

$$P_e = \frac{1}{N-1}\sum_{i=1}^{N}(\underline{x}_i - \overline{\underline{x}})(\underline{x}_i - \overline{\underline{x}})^T. \tag{2.16}$$

Here the errors for each member are measured as a deviation from the ensemble mean. The ($n \times N$) state ensemble matrix X_e is defined as

$$X_e = \frac{1}{\sqrt{N-1}}\begin{pmatrix} \underline{x}_1 & \underline{x}_2 & \dots & \underline{x}_N \end{pmatrix}. \tag{2.17}$$

The $n \times N$ ensemble perturbation matrix (X) is defined as:

$$X = \frac{1}{\sqrt{N-1}}\begin{pmatrix} \underline{x}_1 - \overline{\underline{x}} & \underline{x}_2 - \overline{\underline{x}} & \dots & \underline{x}_N - \overline{\underline{x}} \end{pmatrix}. \tag{2.18}$$

Thus P_e can be written as

$$P_e = XX^T \tag{2.19}$$

and X represents a matrix square root of P_e.

2.3.2 Forecast equations

The forecast step for the EnKF ensemble members is identical to the KF except the EnKF can use a non-linear model m. Assume for simplicity that the there are no model errors.

The forecast state $\underline{x}_i^f$ at time t_k is defined as

$$\underline{x}_{i,k}^f = m(\underline{x}_{i,k-1}^a). \tag{2.20}$$

Here the subscript notation i represents the ensemble members $i = 1, 2, 3...N$ and k represents the time-step t_k. There is added expense in the EnKF in evolving an ensemble of states rather than just the one in the KF. However, savings are made in that the forecast error covariance matrix is evolved implicitly. It is calculated from

$$P_e^f = X^f (X^f)^T, \tag{2.21}$$

where the forecast perturbation matrix is defined in equation 2.18.

2.3.3 Analysis equations

Assume for simplicity that the observation operator H is linear and does not depend on time. A detailed description of how a non-linear observation operator could be applied is shown in section 4.5 of Evensen [2003]. Since all the variables occur at the same time for simplicity the time subscript k is not included. Suppose the observations $\underline{y}$ used in the KF are used in the EnKF. An ensemble Kalman gain matrix K_e (of dimension $n \times p$) can be written as

$$K_e = P_e^f H^T (H P_e^f H^T + R)^{-1}. \tag{2.22}$$

In the early implementations of the EnKF the ensemble analysis error covariance is calculated using the same method as for the KF (Burgers et al., 1997). Using the notation above the analysis ensemble perturbation matrix was updates as

$$X^a = (I - K_e H) X^f. \tag{2.23}$$

However, this implied (from equation 2.19) that the analysis error covariance is

$$P_e^a = (I - K_e H) P_e^f (I - K_e H)^T. \tag{2.24}$$

This is a factor $(I - K_e H)$ too large compared with equation 2.14. In order to solve this problem an observation ensemble is defined as

$$\underline{d}_i = \underline{y} + \epsilon_i. \tag{2.25}$$

The observation ensemble $\underline{d}_i$ (of length p) has ensemble members $i = 1, ..., N$. Here ϵ_i is pseudo-random observation errors normally distributed as $N(0, R)$ (with mean zero and covariance R). An observation state matrix D_e can be defined with columns $\underline{d}_i$, $i = 1, ..., N$. An observation ensemble perturbation matrix D analogous to equation 2.18 (of dimension $p \times N$) has columns $\underline{d}'_i$, $i = 1, ..., N$. Also the observation ensemble covariance matrix is now defined as

$$R_e = DD^T. \tag{2.26}$$

The new Kalman gain has R_e in place of R:

$$K_e = P_e^f H^T (HP_e^f H^T + R_e)^{-1}. \tag{2.27}$$

The EnKF analysis step is similar to the KF, but uses the ensemble Kalman gain and the observation ensemble matrix as follows

$$\underline{x}_i^a = \underline{x}_i^f + K_e(\underline{d}_i - H\underline{x}_i^f). \tag{2.28}$$

for $i = 1, ..., N$. The ensemble mean can then be calculated as

$$\overline{\underline{x}^a} = \overline{\underline{x}^f} + K_e(\overline{\underline{d}} - H\overline{\underline{x}_e^f}). \tag{2.29}$$

Here $\overline{\underline{d}}$ is replacing the actual observation $\underline{d}$. This can be accepted as it is, considering that $\overline{\underline{d}}$ tends to $\underline{d}$ as the ensemble size increases. However, imposing the constraint $\overline{\epsilon} = 0$ on the vectors will ensure $\overline{\underline{d}} = \underline{d}$ (Evensen, 2003). The ensemble perturbation matrix is

$$X^a = (I - K_e H)X^f + K_e D. \tag{2.30}$$

Thus

$$P_e^a = (I - K_e H)P_e^f + (I - K_e H)X^f D^T K_e^T + K_e D(X^f)^T (I - K_e H)^T \tag{2.31}$$

The first term in this expression is the desired result. Evensen [2003] discusses how this desired result can be achieved, but uses a different notation. As long as the distributions used to generate the model state ensemble and observation state ensemble are independent then the random vectors ϵ_i are independent of $\underline{x}_i^f - \overline{\underline{x}^f}$. This implies that $X^f D^{fT}$ and hence the second and third terms tend to zero as the ensemble size increases. Equation 2.31 can be left as it is, but in order to make the 2nd and 3rd terms vanish altogether then the constraint $\overline{(\underline{x}^f - \overline{\underline{x}^f})\epsilon^T} = 0$ can be imposed. Information on how to minimize the calculation of large covariance matrices in the analysis step when implementing the scheme is also discussed in Evensen [2003].

2.3.4 Advantages and Disadvantages of the EnKF

The EnKF is able to use non-linear models, which is necessary in NWP. Using an ensemble of state estimates should improve the quality of the forecast provided that the members have been sampled properly. For example, care must be taken to ensure that the size of the sample is statistically representative of the model (Kalnay, 2003). Many problems can develop from undersampling, some of which have been investigated by Petrie [2008]. Inbreeding is one such problem. This means that the analysis error covariances are systematically underestimated after each of the observation assimilations. The obvious expense from the EnKF comes from maintaining the ensemble of state estimates. However, the covariance matrices are no longer evolved using the forecast model like they are in the KF, which is cheaper to implement. Also, a single Kalman gain is applied to each state estimate, which is less expensive than if separate Kalman gains were used for each member.

2.3.5 Summary of the EnKF

To summarise, The EnKF uses a statistical sample of state estimates that are forecasted forewards in time using a non-linear forecast model (equation 2.20). Unlike the KF, the forecast ensemble covariance matrix is not updated using the forecast model, but is updated using the matrix square of the forecast ensemble perturbation matrices (equation 2.21). In order that the analysis error covariance matrix can be calculated correctly observation perturbations are introduced (equation 2.25). Filters such as the EnKF that use perturbed observations are called stochastic filters. The forecast ensemble can then be updated using a single Kalman gain matrix (equation 2.27) to produce the analysis (equation 2.28). The desired result for the analysis error covariance matrix can be achieved by imposing the constraint $\overline{(\underline{x}^f - \overline{\underline{x}^f})\epsilon^T} = 0$. This means that the last term on the right hand side of equation 2.31 vanishes.

Unlike the KF the EnKF can use non-linear models, necessary for NWP. Also an ensemble of state estimates should improve the quality of the forecast over one state estimate provided that the members have been sampled properly. The EnKF has added expense in maintaining an ensemble of state estimates instead of one for the KF. However, calculating the Kalman gain matrix and error covariance matrices are less expensive in the EnKF.

2.4 The Ensemble Square Root Filter

The Ensemble Kalman Square Root Filter (EnSRF) presented here uses the formulation in Tippett et al. [1999]. The EnKF is an example of a filter that uses perturbed observations. This implies that the observation ensemble is created by adding random vectors to the actual observations for each member. Therefore the EnKF is called a stochastic filter. Filters that do not use perturbed observations are called deterministic filters (Hamill, 2006). The EnSRF is a deterministic filter and differs in the analysis step to the EnKF (section 2.3).

2.4.1 The observational ensemble perturbation matrix

Since the EnSRF is deterministic, the observational ensemble perturbation matrix Y is different than for the EnKF (section 2.3). The state members ($\underline{x}_i$) can be mapped to observation space by

$$\underline{y}_i = H\underline{x}_i.$$

It follows that the ensemble perturbations are

$$\begin{aligned} \underline{y}'_i &= H\underline{x}_i - \overline{H\underline{x}} \\ &= H(\underline{x}_i - \overline{\underline{x}}). \end{aligned} \tag{2.32}$$

Thus Y (of dimension $p \times N$) with columns $\underline{y}'_i, i = 1, ..., N$ can be defined as

$$Y = HX. \tag{2.33}$$

The forecast observation ensemble perturbations are given by

$$Y^f = HX^f. \tag{2.34}$$

2.4.2 The Kalman gain matrix

The Kalman gain can now be written in terms of the ensemble perturbation matrices X and Y. Note that the observational error covariance is a single matrix R, as for the KF (section 2.2).

$$\begin{aligned} K_e &= P_e^f H^T (HP_e^f H^T + R)^{-1} \\ &= X^f (X^f)^T H^T (HX^f (X^f)^T H^T + R)^{-1} \\ &= X^f (Y^f)^T S^{-1}, \end{aligned} \tag{2.35}$$

where

$$S = Y^f(Y^f)^T + R. \tag{2.36}$$

S (of dimension $p \times p$) is positive definite and hence invertible.

2.4.3 Analysis step

The analysis ensemble mean is given by the following step.

$$\begin{aligned} \underline{\overline{x}^a} &= \underline{\overline{x}^f} + K_e(\underline{y} - \overline{\underline{y}^f}) & (2.37)\\ &= \underline{\overline{x}^f} + K_e(\underline{y} - \overline{H\underline{x}^f}). & (2.38) \end{aligned}$$

using the K_e defined in equation 2.35. The analysis covariance matrix (P_e^a) can now be written as

$$\begin{aligned} P_e^a &= X^a(X^a)^T \\ &= (I - K_eH)P_e^f \\ &= (I - X^f(Y^f)^TS^{-1}H)X^f(X^f)^T \\ &= X^f(I - (Y^f)^TS^{-1}Y^f)(X^f)^T. & (2.39) \end{aligned}$$

The analysis ensemble perturbation matrix (X^a) is updated by

$$X^a = X^fT, \tag{2.40}$$

where T (of dimension $N \times N$) is a matrix square root of $I - (Y^f)^TS^{-1}Y^f$ such that

$$TT^T = I - (Y^f)^TS^{-1}Y^f. \tag{2.41}$$

Thus the analysis states can now be updated as

$$\underline{x}_i^a = \underline{\overline{x}^a} + \underline{x}_i'^a, \tag{2.42}$$

where $\underline{x}_i'^a$ are the perturbation vectors for $i = 1, 2, .., N$.

2.4.4 Advantages and Disadvantages of the EnSRF

The EnSRF is a deterministic filter, which gives it some advantages over the EnKF. Sampling errors are common due to the perturbed observations in the EnKF (Evensen, 2004). This is particularly common when the number of ensemble members is small. However these problems are eliminated when the square root analysis algorithm is used as it does not use perturbed observations (Evensen, 2004). Square root filters (SRFs) have also been shown to be computationally more efficient and less expensive (Hamill, 2006). This is partly because they do not calculate the forecast error covariance matrix, only the ensemble state perturbations (Hamill, 2006).

2.4.5 Summary of the EnSRF

To summarise, the EnSRF is a deterministic filter, which means it does not use perturbed observations. This implies the observational ensemble perturbation matrix Y can be written in terms of the ensemble state perturbations in equation 2.33. Like for the EnKF, a statistical sample of state estimates are forecasted forwards in time using a non-linear forecast model. The forecast error covariance matrix is updated using the forecast ensemble perturbation matrices (equation 2.21). The Kalman gain used in the analysis stage now contains the EnSRF version of Y and the observational error covariance R is the same R used for the KF (equation 2.4). The analysis error covariance (see equation 2.39) can now be updated using the analysis perturbation matrices. The analysis perturbation matrix (equation 2.40) is updated using the square root matrix T (equation 2.41).

Unlike the EnKF, the EnSRF does not use pertubed observations, which can introduce sampling errors. SRFs such as the EnSRF have also been shown to be more efficient and less expensive than the EnKF (Hamill, 2006).

2.5 The Ensemble Transform Kalman Filter (ETKF)

2.5.1 The ETKF equations

The Ensemble Transform Kalman Filter was introduced by Bishop et al. [2001]. It is a form of square root filter. It is the chosen filtering method to be used in experimental work in this report (see the implementation section 4). The ETKF uses the identity

$$I - (Y^f)^T S^{-1} Y^f = (I + (Y^f)^T R^{-1} Y^f)^{-1}. \tag{2.43}$$

This can be verified as follows. Firstly multiply $I - (Y^f)^T S^{-1} Y^f$ by $(I + (Y^f)^T R^{-1} Y^f)$

$$\begin{aligned} & I - (Y^f)^T S^{-1} Y^f \times (I + (Y^f)^T R^{-1} Y^f) \\ = \; & I - (Y^f)^T S^{-1} Y^f + (Y^f)^T R^{-1} Y^f - ((Y^f)^T S^{-1} Y^f)((Y^f)^T R^{-1} Y^f) \end{aligned}$$

and then replacing S by equation 2.36 to give

$$\begin{aligned}
& I + (Y^f)^T (Y^f (Y^f)^T + R)^{-1} \left[-Y^f + (Y^f (Y^f)^T + R) R^{-1} Y^f - Y^f (Y^f)^T R^{-1} Y^f \right] \\
= \; & I + (Y^f)^T (Y^f (Y^f)^T + R)^{-1} \left[-Y^f + Y^f (Y^{fT}) R^{-1} Y^f + Y^f - Y^f (Y^f)^T R^{-1} Y^f \right] \\
= \; & I + (Y^f)^T (Y^f (Y^f)^T + R)^{-1} \times 0 \\
= \; & I.
\end{aligned}$$

This is important since calculating R^{-1} is usually simpler than calculating S^{-1}, since R is usually diagonal while S has a more complicated structure. An eigenvalue decomposition can then be calculated by

$$Y^{fT} R^{-1} Y^f = U \Lambda U^T, \tag{2.44}$$

where U is orthogonal and Λ is diagonal. The identity in equation 2.43 can now be written as

$$I - (Y^f)^T S^{-1} Y^f = U (I + \Lambda)^{-1} U^T. \tag{2.45}$$

The final term U^T is required here to make the filter unbiased (Livings et al., 2008). Hence the square root is given by

$$T = U (I + \Lambda)^{-\frac{1}{2}}. \tag{2.46}$$

This T is the square root that gives the ETKF. This is more efficient to compute than the square root in the EnSRF (matrix square root of 2.41) since $(I + \Lambda)$ is diagonal, making the calculation of $(I + \Lambda)^{-1}$ simple. According to Hamill [2006] the ETKF was verified to be faster to compute than the EnSRF of Whitaker and Hamill [2002].

2.5.2 Summary of the ETKF

To summarise, the ETKF is a form of square root filter that uses the identity $I - (Y^f)^T S^{-1} Y^f = (I + (Y^f)^T R^{-1} Y^f)^{-1}$. Then taking an eigenvalue decomposition of $Y^{fT} R^{-1} Y^f$ (equation 2.44) forms a square root matrix 2.46, which is more efficient to compute than the EnSRF square root matrix (matrix square root of 2.41).

2.6 Smoothing - What is it and why is it important in data assimilation?

Smoothing differs from filtering in that future observations are assimilated in the analysis step (Ravela and McLaughlin, 2007). If you consider a time window with all available observations in use, then theoretically the smoothed analysis at the end of the window is the same as the filtered analysis at the end of the window since in both cases all the observations have been assimilated. However, the analysis in the middle of the window will not be the same since the filter will only use the observations in the first half of the window while the smoother will use observations in both the first and second half of the window. Thus at all points inside the window you would expect to get more accurate analyses for the smoother than for the filter. However, a great deal more work will be required to get this since past, present and future observations in the window will have to be assimilated at each step for the smoother, as opposed to just past and present observations for the filter (Ravela and McLaughlin, 2007).

The following descriptions of different types of smoothing methods come from Ravela and McLaughlin [2007]. There are two types of smoothing, fixed interval and fixed lag smoothing. Fixed interval smoothing updates all desired states within a time interval [0,T] using all the available observations. Fixed lag smoothing only updates a fixed number of states prior to the current observation time. Therefore it only requires updates in a lag window W (where $W < T$) before the most recent measurement. Fixed lag smoothing and their applications have generally proven to be computationally faster, since fewer prior model states are updated using the current observations. Fixed lag smoothing can be a good approximation for long interval smoothing problems. For example, fixed lag smoothing methods have been tested for the Goddard Earth Observing System (GEOS). It has been tested both for retrospective analysis of climate and for short term forecasting (Zhu et al., 1999).

A schematic of a filter analysis (black curve in filter diagram) and a smoother analysis (black curve on smoother diagram) is shown in Chapter 1 in Figure 1.1. The analysis time t_k is highlighted to show that the filter uses just past and present observations while the fixed-interval smoother uses all available observations. The fixed-lag smoother analysis is shown in Figure 1.2 and only assimilates one ($lag = 1$) future observation. As would be expected, the filter analysis tends towards the true solution as more observations are assimilated, while the fixed-interval analysis oscillates at approximately the same magnitude in the time window. The fixed-interval smoother is using all the observations at each analysis step so the accuracy should be constant. The fixed-lag analysis improves over the time window and is more accurate than the filter but less accurate than the fixed-interval smoother. This is also expected since the fixed-lag smoother uses more observations than the filter inside the time-window. The accuracy of the fixed-lag analysis depends on the lag. A higher lag means that more observations will be used by the analysis at the beginning of the time window. However, at least at the beginning of the window, the fixed-lag smoother will use less observations than the fixed-interval smoother, so will be less accurate.

Both fixed lag and fixed interval smoothing can be solved in two different ways. They can be solved in a batch form where all model updates are derived simultaneously from all the measurements. They can also be solved in sequential form, where model updates are derived recursively through time (Ravela and McLaughlin, 2007). Variational algorithms such as 4D-Var use batch methods for intervals of fixed length (Menard and Daley, 1996). Sequential smoothing methods such as the EnKS can be more useful than batch form methods due to the flow dependent error covariances. They can accomodate model errors without an increase in computational effort. However, sequential smoothing methods are generally more computationally demanding as a function of the interval or lag length than the batch form methods (Ravela and McLaughlin, 2007).

2.7 The Kalman Smoother (KS)

The description of the Kalman Smoother given here is based on the linear smoothing method of Jazwinski [1970] (pages 215-217), who uses the Bayesian Maximum likelihood approach of Rauch [1967]. The fixed-lag Ensemble Transfrom Kalman Smoother used for experimental work in this report follows on from the KS. The desired outcome of the KS is the probability of the state vector ($\underline{x}_k$) at time t_k given observations up to time t_{k+l} ($l > 0$), which is expressed as $p(\underline{x}_k|\underline{y}_{1:k+l})$, where $\underline{y}_{1:k+l}$ is shorthand for $\underline{y}_1, \underline{y}_2, \underline{y}_3, ..., \underline{y}_{k+l}$. This can be expressed as

$$p(\underline{x}_k|\underline{y}_{1:k+l}) = \frac{p(\underline{x}_k, \underline{y}_{1:k+l})}{p(\underline{y}_{1:k+l})} \tag{2.47}$$

using the definition of conditional probability (equation 2.3). Now

$$p(\underline{x}_k|\underline{y}_{1:k+l}) = p(\underline{x}_k, \underline{y}_{k+1:k+l}|\underline{y}_{1:k})p(\underline{y}_{1:k}) \tag{2.48}$$

again using the definition of conditional probability. Using the independence properties of observations

$$\begin{aligned} p(\underline{x}_k, \underline{y}_{k+1:k+l}|\underline{y}_{1:k}) &= p(\underline{y}_{k+1:k+l}|\underline{x}_k, \underline{y}_{1:k})p(\underline{x}_k|\underline{y}_{1:k}) \\ &= p(\underline{y}_{k+1:k+l}|\underline{x}_k)p(\underline{x}_k|\underline{y}_{1:k}). \end{aligned} \tag{2.49}$$

Assuming Gaussian distributions, the Kalman filtering distribution is

$$p(\underline{x}_k|\underline{y}_{1:k}) = N(\underline{x}_k^a, P_k^a). \tag{2.50}$$

The observation distribution is

$$p(\underline{y}_{k+1:k+l}|\underline{x}_k) = \prod_{i=1}^{l} N(HM_{k\to k+i}\underline{x}_k, R). \quad (2.51)$$

Substituting equations 2.50 and 2.51 into 2.49 and taking logs gives the cost function

$$\frac{1}{2}(\underline{x}_k - \underline{x}_k^a)^T P_k^{a-1}(\underline{x}_k - \underline{x}_k^a) + \sum_{i=1}^{l}(\underline{y}_{k+i} - HM_{k\to k+i}\underline{x}_k)^T R^{-1}(\underline{y}_{k+i} - HM_{k\to k+i}\underline{x}_k). \quad (2.52)$$

Using one future observation time $l = 1$, the minimizer of equation 2.52 satisfies

$$\underline{x}_k^s = \underline{x}_k^a + K_{k|k+1}^s(\underline{y}_{k+1} - HM_{k\to k+1}\underline{x}_k^a), \quad (2.53)$$

where $\underline{x}_k^s$ is the smoothed analysis. Note that the filtered analysis must have already been calculated and stored before calculating 2.53. Here $K_{k|k+1}^s$ is the Kalman gain at time t_k using observations up to t_{k+1}.

$$K_{k|k+1}^s = P_k^a M_{k\to k+1}^T H^T(HM_{k\to k+1}P_k^a M_{k\to k+1}^T H^T + R)^{-1} \quad (2.54)$$

and

$$P_k^s = (I - K_{k|k+1}^s HM_{k\to k+1})P_k^a. \quad (2.55)$$

Now consider the smoothed solution using two future observations, $l = 1, 2$. The minimizer of equation 2.52 now satisfies

$$\begin{aligned} \underline{x}_k^s &= \underline{x}_k^a + K_{k|k+1}^s(\underline{y}_{k+1} - HM_{k\to k+1}\underline{x}_k^a) \\ &+ K_{k|k+2}^s(\underline{y}_{k+2} - HM_{k\to k+2}(\underline{x}_k^a + K_{k|k+1}^s(\underline{y}_{k+1} - HM_{k\to k+1}\underline{x}_k^a))) \end{aligned} \quad (2.56)$$

Now consider the solution using L future observations, $l = 1, 2, ...L$ for the general fixed-lag smoother. The same notation for the fixed-lag smoother of Cohn et al. [1994] is introduced. The smoothed analysis will now be represented by $\underline{x}_{k|k+l}^a$, where the subscript $k|k+l$ notation represents the value at time t_k using observations up to t_{k+l}. Thus $\underline{x}_{k|k+l}^a$ is calculated from

$$\underline{x}_{k|k+l}^a = \underline{x}_{k|k+l-1}^a + K_{k|k+l}(\underline{y}_{k+l} - HM_{k\to k+l}\underline{x}_{k|k+l-1}^a). \quad (2.57)$$

Here $K_{k|k+l}$ is the Kalman gain at time t_k using observations up to t_{k+l}.

$$K_{k|k+l} = P^a_{k|k+l-1} M^T_{k\to k+l} H^T (H M_{k\to k+l} P^a_{k|k+l-1} M^T_{k\to k+l} H^T + R)^{-1} \tag{2.58}$$

and

$$P^a_{k|k+l} = (I - K_{k|k+l} H M_{k\to k+l}) P^a_{k|k+l-1}. \tag{2.59}$$

2.7.1 Summary of the KS

The fixed-lag smoother has been defined using the linear smoothing method of Jazwinski [1970]. A Bayesian likelihood approach produces a cost function, which is minimised to produce the smoothed solution. Firstly, the analysis states over the time window must be calculated for the KF. Using the fixed-lag method of Cohn et al. [1994], the fixed-lag smoothed solution for $l \geq 1$ is the smoothed analysis state at time t_k using observations up to t_{k+l}. The smoothed solution $\underline{x}^a_{k|k+l}$ (see equation 2.57) updates the smoothed solutions for the previous lag $\underline{x}^a_{k|k+l-1}$ using the Kalman gain for the current lag time $K_{k|k+l}$. This Kalman gain gives a weighting to the observations at the current lag time $\underline{y}_{k+l}$ according to a ratio between the previous lag error covariances in $P^a_{k|k+l-1}$ and current observational error covariances in R (see equation 2.58). The smoothed analysis error covariance $P^a_{k|k+l}$ (see equation 2.59) updates the previous lag analysis error covariance $P^a_{k|k+l-1}$ using the Kalman gain.

2.8 The Ensemble Square Root Smoother (EnSRS)

The fixed-lag Ensemble Square Root Smoother was introduced by Whitaker and Compo [2002]. The mean ensemble fixed-lag smoother $\overline{\underline{x}^a}_{k|k+l}$ for $l \geq 1$ is the smoothed ensemble analysis state at time t_k using observations up to t_{k+l}. Using the general derivation in section 7.3 of Anderson and Moore [1979], a fixed-lag smoother can be written as where $G_{k|k+l}$ is the gain matrix. Different choices for the gain $G_{k|k+l}$ lead to different smoother types. Cohn et al. [1994] derived the optimum Kalman smoother gain in a least squares sense. This is defined as

$$K_{k|k+l} = (H P^{fa}_{k+l,k|k+l-1})^T [H P^f_{k+l|k+l-1} H^T + R_{k+l}]^{-1} \tag{2.60}$$

where $P^{fa}_{k+l,k|k+l-1}$ is the cross-covariance matrix between the forecast for the Kalman filter update equation for time $k+l$ and the lag $l-1$ Kalman smoother analysis for time k. Whitaker and Compo [2002] used a Monte-Carlo technique to show that the EnSRS cross-covariance matrix can be computed from

$$(HP^{fa}_{k+l,k|k+l-1})^T = X^a_{k|k+l-1}(HX^f_{k+l|k+l-1})^T. \tag{2.61}$$

Equation 2.61 substituted into equation 2.60 is equivalent to equation 2.58 but applied to the EnSRS. The EnSRS ensemble mean can then be calculated from

$$\overline{\underline{x}}^a_{k|k+l} = \overline{\underline{x}}^a_{k|k+l-1} + K_{k|k+l}(\underline{y}_{k+l} - H\overline{\underline{x}}^f_{k+l|k+l-1}). \tag{2.62}$$

The lag l smoothed analysis state ensemble deviation matrix is

$$X^a_{k|k+l} = X^a_{k|k+l-1} - \hat{K}_{k|k+l}HX^f_{k+l|k+l-1} \tag{2.63}$$

where $\hat{K}$ is equivalent to

$$\hat{K}_{k|k+l} = \left(1 + \sqrt{\frac{R}{HP^f_{k+l|k+l-1}H^T + R}}\right)^{-1}. \tag{2.64}$$

$\hat{K}_{k|k+l}$ is defined assuming R to be diagonal and that observations are assimilated serially, one at a time. Thus R and $HP^f_{k+l|k+l-1}H^T + R$ are scalars and hence the square root is possible. Also $\hat{K}_{k|k+l}$ is defined such that the ensemble analysis cross covariance is equal to the value predicted by the theory of Cohn et al. [1994], which implies

$$P^a_{k|k+l} = P^a_{k|k+l-1} - K_{k|k+l}HP^{fa}_{k+l,k|k+l-1}. \tag{2.65}$$

The analysis states are updated by

$$\underline{x}^a_{i,k|k+l} = \overline{\underline{x}}^a{}_{i,k|k+l} + \underline{x}'_{i,k|k+l}{}^a \tag{2.66}$$

2.8.1 Experimental results with the EnSRS

Whitaker and Compo [2002] used the 40-dimensional model of Lorenz and Emanuel [1998] to integrate observations that have uncorrleated errors with unit variance. The ensemble mean errors for the EnSRS were less than for the EnSRF and lower for more ensemble members and higher lags. This would be expected due to less sampling errors with the larger ensemble size (Ehrendorfer, 2007) and greater smoothing near the beginning of the time window for the higher lags. They also showed that the difference between the errors in a 10 and 50 member ensemble for the EnSRS increase with increasing lag l. This indicates that sampling error in the estimate of the forecast-analysis error cross covariance matrix ($P^{fa}_{k+l,k|k+l-1}$) increases with l. This means a larger ensemble is required to take advantage of observations further removed from the analysis time. This is consistent with the fact that the quality of the analysis starts to degrade as the lag is increased beyond a certain point. This point is higher for more ensemble members (Whitaker and Compo, 2002).

2.8.2 Summary of the EnSRS

The EnSRS of Whitaker and Compo [2002] uses the fixed-lag method of Cohn et al. [1994]. The fixed-lag ensemble smoothed solution for $l \geq 1$ is the ensemble smoothed analysis state at time t_k using observations up to t_{k+l}. The Kalman gain (see equation 2.60) uses a cross covariance matrix between the forecast for the Kalman filter update equation for time $k+l$ and the lag $l-1$ Kalman smoother analysis for time k. This is derived from a least squares sense (see Cohn et al. [1994]) and is used to produce the smoothed analysis ensemble. The cross-covariance and Kalman gain are also used to update the analysis covariance matrix $P^a_{k|k+l}$ (see equation 2.65). Experimental results by Whitaker and Compo [2002] from the 40-dimensional model of Lorenz and Emanuel [1998] have shown that as expected ensemble mean errors for the EnSRS are less than for the the EnSRF and lower for more ensemble members and higher lags. However, results suggested that sampling error in the estimate of the forecast-analysis error cross covariance matrix ($P^{fa}_{k+l,k|k+l-1}$) increases with l. Thus larger ensemble sizes are required to offset this increase in sampling error for higher lags.

2.9 The Ensemble Transform Kalman Smoother and an approximation

2.9.1 ETKS using a smoothed forecast state (ETKS)

The Kalman Smoother equations in section 2.7 can be modified to produce the Ensemble Transform Kalman Smoother. The ETKS method described here follows on from the ETKF of Bishop et al. [2001] using the fixed-lag smoother method of Cohn et al. [1994]. It is very similar to the smoothing method used in the experimental work for this report (section 2.9.2) and could be a useful method to investigate in the future.

Consider time t_k to be the current smoothing time and l to be the lag time. The solution for the smoother ensemble mean is equivalent to equation 2.62 in section 2.8. However, the Kalman gain is now written in a similar form to the KS. The only difference being that the forecast model is evolved in the same way as for the fixed-lag smoother of Cohn et al. [1994], from t_{k+l-1} to t_{k+l} instead of from t_k to t_{k+l} for the KS. M is now short for

$M_{k+l-1\rightarrow k+l}$. Thus the Kalman gain can be written as

$$\begin{aligned} K_{k|k+l} &= P^a_{k|k+l-1} M^T H^T (H M P^a_{k|k+l-1} M^T H^T + R)^{-1} \\ &= X^a_{k|k+l-1} X^a_{k|k+l-1}{}^T M^T H^T (H M X^a_{k|k+l-1} X^a_{k|k+l-1}{}^T M^T H^T + R)^{-1} \\ &= X^a_{k|k+l-1} X^f_{k+l|k+l-1}{}^T H^T (H X^f_{k+l|k+l-1} X^f_{k+l|k+l-1}{}^T H^T + R)^{-1}. \qquad (2.67) \end{aligned}$$

The analysis covariance matrix can be written as

$$\begin{aligned} P^a_{k|k+l} &= X^a_{k|k+l} X^a_{k|k+l}{}^T \\ &= (I - X^a_{k|k+l-1} X^f_{k+l|k+l-1}{}^T H^T [H X^f_{k+l|k+l-1} X^f_{k+l|k+l-1}{}^T H^T + R]^{-1} H M) X^a_{k|k+l-1} X^a_{k|k+l-1}{}^T \\ &= X^a_{k|k+l-1} (I - Y^f_{k+l|k+l-1}{}^T [Y^f_{k+l|k+l-1} Y^f_{k+l|k+l-1}{}^T + R]^{-1} Y^f_{k+l|k+l-1}) X^a_{k|k+l-1}{}^T \\ &= X^a_{k|k+l-1} T_{k+l|k+l-1} T^T_{k+l|k+l-1} X^a_{k|k+l-1}{}^T. \qquad (2.68) \end{aligned}$$

where T is defined in the same way for the EnSRF in equation 2.41. Note that the innovation vector used in the smoothing of the ETKS uses the forecast states from the smoothed analysis. In other words, the innovation vector in the ETKS is written as

$$(\underline{y}_{k+l} - H \underline{x}^f_{i,k+l|k+l-1}). \qquad (2.69)$$

The innovation vector is essential in calculating the ensemble mean analysis, given by equation 2.62. Note that T is also defined from the forecast perturbations of the smoothed analysis such that

$$T_{k+l|k+l-1} T^T_{k+l|k+l-1} = I - (Y^f)^T_{k+l|k+l-1} S^{-1}_{k+l|k+l-1} Y^f_{k+l|k+l-1} \qquad (2.70)$$

where

$$S = Y^f_{k+l|k+l-1} (Y^f)^T_{k+l|k+l-1} + R. \qquad (2.71)$$

Note that

$$Y^f_{k+l|k+l-1} = H X^f_{k+l|k+l-1}. \qquad (2.72)$$

The ensemble analysis perturbation matrix is updated as

$$X^a_{k|k+l} = X^a_{k|k+l-1} T_{k+l|k+l-1}. \qquad (2.73)$$

The mean is updated in the same way as the EnSRS in equation 2.62. The analysis equation can now be updated as

$$\underline{x}^a_{i,k|k+l} = \overline{\underline{x}}^a_{k|k+l} + \underline{x}'_{i,k|k+l}{}^a \qquad (2.74)$$

Thus equations 2.68, 2.73 and 2.74 give the equations for the l-step fixed lag ETKS.

2.9.2 ETKS using a fixed square root matrix (ETKS2)

The ETKS2 described here is an approximation of the ETKS method described in section 2.9.1. The ETKS2 is the method used in the experimental work for this report (see implementation in section 4.2.1). The innovation vector used in the smoothing of the ETKS2 uses the original forecast states from the ETKF. In other words, the innovation vector in the ETKS2 is written as

$$(\underline{y}_{k+l} - H\underline{x}^f_{i,k+l}) \tag{2.75}$$

Note that the T matrix is calculated from the original forecast perturbations of the ETKF. T is the matrix square root of

$$T_{k+l}T^T_{k+l} = I - (Y^f)^T_{k+l}S^{-1}_{k+l}Y^f_{k+l}. \tag{2.76}$$

$$S = Y^f_{k+l}(Y^f)^T_{k+l} + R. \tag{2.77}$$

Here

$$Y^f_{k+l} = HX^f_{k+l} \tag{2.78}$$

where X^f_{k+l} is the original forecast perturbation matrix calculated in the ETKF. In other words, it is not the updated forecast perturbation matrix calculated by forecasting the smoothed analysis from the previous lag ($X^f_{k+l|k+l-1}$).

2.9.3 Expected differences in ETKS and ETKS2 results

The innovation vector of the ETKS2 (equation 2.75) and the T matrix do not use the forecast from the smoothed analysis unlike the ETKS. The ETKS2 should therefore be cheaper to implement than the ETKS. However, the accuracy of the smoothed analyses of the ETKS2 would be expected to be less than the ETKS for the lag $l > 1$ since the smoothed forecast should be closer to the true solution.

2.9.4 Summary of the ETKS and ETKS2

The ETKS method desribed is a modification of the fixed-lag KS method derived in section 2.7. The smoothed Kalman gain $K_{k|k+l}$ gives a weighting to the observations at the current lag time y_{k+l} according to a ratio between the previous lag error covariance $P^a_{k|k+l-1}$ and current observational error covariance R (see equation 2.67). The analysis error covariance uses the smoothed analysis perturbation matrix (see equation 2.68), which is expressed in terms of the square root matrix $T^T_{k+l|k+l-1}$ defined by equation 2.41 in the EnSRF section 2.4. The ensemble analysis perturbation matrix can then be updated using the T matrix (equation 2.73).

The ETKS2 method differs from the ETKS in that it does not use the forecast from the smoothed analysis in the innovation vector and the square root T matrix. The ETKS2 would therefore be expected to be less expensive to implement but less accurate for the lag $l > 1$.

Chapter 3

The Swinging Spring model

The swinging spring model is the forecast model that will be used for the ETKF and ETKS2 experiments in this report. This chapter begins by describing the swinging spring model and equations from Lynch [2002]. After a brief definition of initialization and its rlevance in data assimilation, the application of normal mode initialization to the swinging spring model will be demonstrated using the implementation and results of Livings [2005].

3.1 The Swinging Spring

The Swinging Spring model of Lynch [2002] is illustrated by the following Figure (3.1) showing coordinates and forces for the swinging spring. The figure is based on the figure from Livings [2005].

A mass m is suspended from a fixed point in a uniform gravitational field, g by a light unstretched spring with length l_0 and elasticity k. The bob is constrained to move in the vertical plane and the string may stretch along its length but cannot bend. The bob is located using polar coordinates (r,θ). The spring length r is measured from the suspension point and the angle θ is measured from the downward path. The generalised momenta are radial momentum $p_r = m\dot{r}$ and angular momentum, $p_\theta = mr^2\dot{\theta}$. The motion is assumed to be simple harmonic (SHM). The following Hamiltonian of the system gives the sum of potential and kinetic energies:

$$H = \frac{1}{2m}(p_r^2 + \frac{p_\theta^2}{r^2}) + \frac{1}{2}k(r - l_0)^2 - mgrcos\theta \tag{3.1}$$

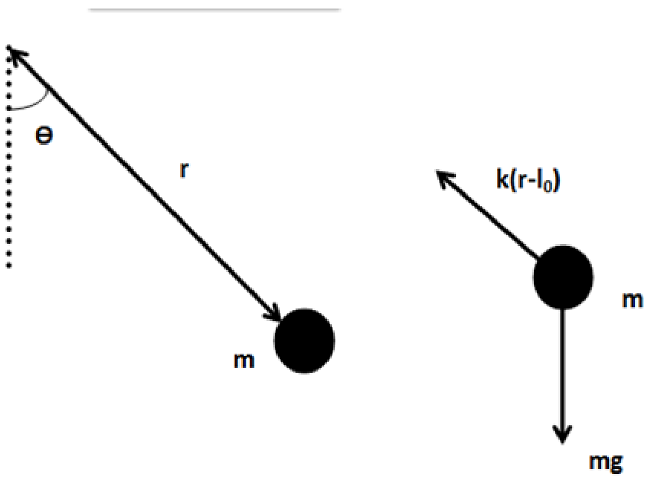

Figure 3.1: Coordinates and forces for the swinging spring. Coordinates are given by angle θ, radius r and bob mass m. The gravitational force is given by mg and the elastic force is given by $k(r - l_0)$ where k is elasticity and r is unstretched length.

From 3.1 one can derive the following equations of motion:

$$\dot{\theta} = \frac{p_\theta}{mr^2} \tag{3.2}$$

$$\dot{p_\theta} = -mgrsin\theta \tag{3.3}$$

$$\dot{r} = \frac{p_r}{m} \tag{3.4}$$

$$\dot{p_r} = \frac{p_\theta^2}{mr^3} - k(r - l_0) + mgcos\theta \tag{3.5}$$

A stable equilibrium point can be considered when the bob is vertically at rest, i.e. $(\theta, p_\theta, r, p_r) = (0, 0, l, 0)$ where l satisfies $k(l - l_0) = mg$. Linearising about this point yields the following equations:

$$\dot{\theta} = \frac{p_\theta}{ml^2} \tag{3.6}$$

$$\dot{p_\theta} = -mgl\theta \tag{3.7}$$

$$\dot{r} = \frac{p_r}{m} \tag{3.8}$$

$$\dot{p_r} = -k(r - l). \tag{3.9}$$

Therefore for small oscillations the system splits into two independent systems of (θ, p_θ)

and (r, p_r). The angular motion satisfies

$$\ddot{\theta} + \frac{g}{l}\theta = 0. \tag{3.10}$$

The elastic motion satisfies

$$\ddot{r'} + \frac{k}{m}r' = 0 \tag{3.11}$$

where $r' = r - l$. These are SHM equations and have angular frequencies

$$\omega_\theta = \sqrt{\frac{g}{l}} \tag{3.12}$$

and

$$\omega_r = \sqrt{\frac{k}{m}}. \tag{3.13}$$

The ratio of frequencies is:

$$\epsilon = \frac{\omega_\theta}{\omega_r} = \sqrt{\frac{mg}{kl}} = \sqrt{\frac{l - l_0}{l}}. \tag{3.14}$$

For the purposes of this work, parameters were used such that $\epsilon << 1$. The rotational mode represents low frequency Rossby waves and contains the slow variables (θ, p_θ). The elastic motion represents high frequency gravity waves and contains the fast variables (r, p_r). The slow and fast variables will interact for finite oscillations. Following the parameters used by Lynch [2002]:

$$\begin{aligned} m &= 1 \\ g &= \pi^2 \\ k &= 100\pi^2 \\ l &= 1 \end{aligned}$$

This gives motions with cyclic frequency $f_\theta = \frac{w_\theta}{2\pi} = 0.5$ and $f_r = \frac{\omega_r}{2\pi} = 5$. Therefore $\epsilon = 0.1$. The initial conditions are $(\theta, p_\theta, r, p_r) = (1, 0, 1.05, 0)$. Using these initial conditions and the equations of motion, the Matlab programme by Livings [2005] was implemented using the numerical method described in section 3.2. The graphs in Figure 3.2 show the motion of the fast and slow variables. It is clear that most of the energy is concentrated around $f = f_\theta$ for the slow variables and most of the energy for the fast variables is concentrated at $f = f_r$. But there is a clear sign that the slow and fast variables have interacted in the peak in the fourier transforms of r and p_r around $f = 1 = 2f_\theta$. This is a clear sign of the non-linear nature of the system in that energy is being transferred from the fast variables to the slow ones.

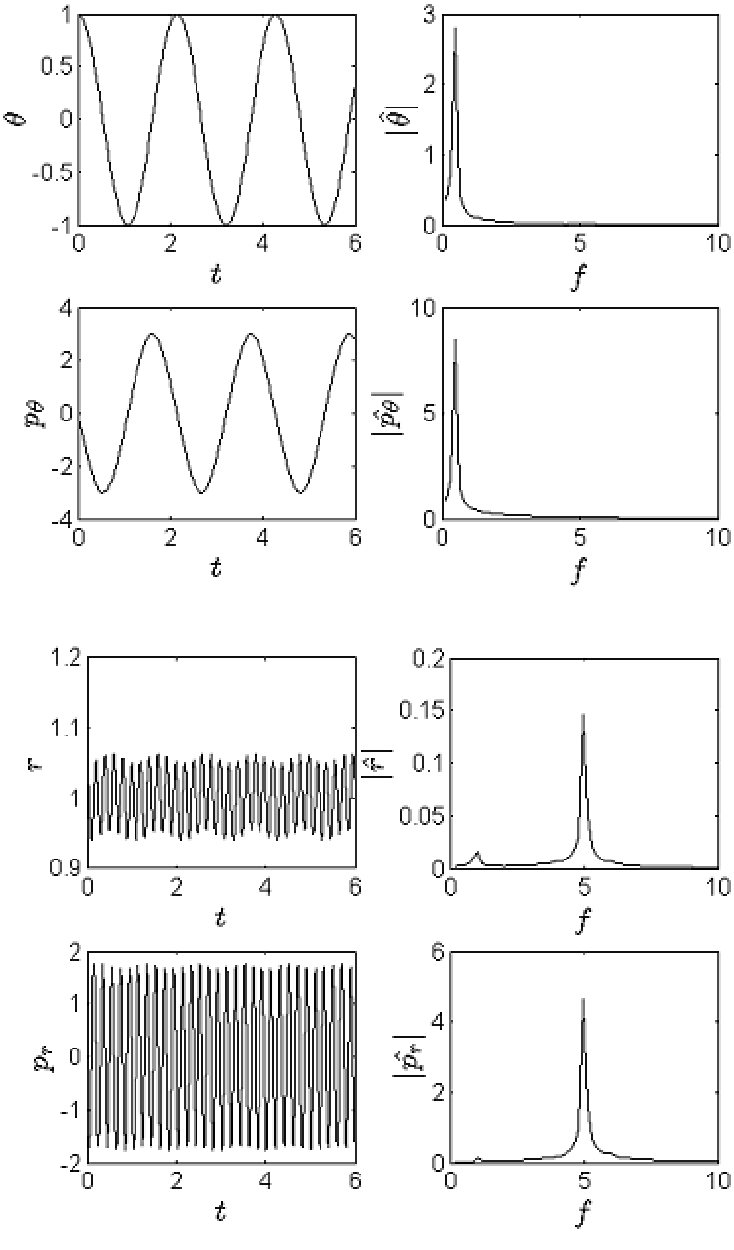

Figure 3.2: Coordinates and their fourier transforms of slow and fast variables for an uninitialized swinging spring. Parameter values give rotational and elastic frequencies of $f_\theta = 0.5$ and $f_r = 5$ respectively. Initial conditions are $(\theta, p_\theta, r, p_r) = (1, 0, 1.05, 0)$.

3.1.1 Summary of the swinging spring system

The swinging spring models the motion of a mass m suspended from a swinging spring using the method of Lynch [2002]. The equations of motion are given by equations 3.2 to 3.5. Linearising about a stable equilibrium point when the bob is at rest gives equations which split the motions of the spring into high frequency (equation 3.11) and low frequency (equation 3.10). Using the method described in the following section (3.2) the motions of the swinging spring were plotted for both high and low frequency variables. The plots revealed that there was some interaction between the high and low frequency motions, indicating the non-linearity of the system.

3.2 Numerical method of integration for the swinging spring

The following information about the implementation of the swinging spring in Matlab comes from the description given in Livings [2005] and the implementation was validated by Livings [2005]. The swinging spring equations could be integrated using the standard Matlab ode45 function. This uses an explicit Runge-Kutta (4,5) pair with a changeable step size. This means that a 5th order Runge-Kutta scheme is used to integrate the equations and the difference between this and a 4th order scheme is an estimate of the truncation error.

The error tolerance can be used to determine a limit on the step size. Let **Reltol** and **Abstol** represent the relative tolerance and absolute tolerance for each state space coordinate. If $y(j)$ is the jth component of the solution vector and $e(j)$ is the corresponding error component from the Runge-Kutta pair, then the step size must be minimized such that:

$$|e(j)| \leq max(\mathbf{Reltol} \times |y(j)|, \mathbf{Abstol}).$$

A primary aim was to keep the truncation error small compared with the observation error. This was done by using 10^{-3} for **Reltol** and 10^{-6} for every component of **Abstol**. The parameter **MaxStep** acts as an upper bound on the step size. This can be used to ensure stability of the numerical method. For information on stability of Runge-Kutta methods with relation to linear systems see Lambert [1991], section 5.12. Using the Runge-Kutta(4,5) scheme the maximum step size h for stability is $h \leq 0.3$. This is explained in Livings [2005], section 4.3. Since the period of a general harmonic oscillator is $T = 2\pi$, taking the upper bound **MaxStep** of the step size to be $\frac{T}{20}$ for an arbitrary harmonic oscillator ensures stability for the system (see Livings [2005]). For the swinging spring system, in the case of no initialization the low frequecy motion has period $T_\theta = 2$ and the high frequency motion has period $T_r = 0.2$. Therefore taking the maximum step-size to be **MaxStep**= $\frac{T_r}{20} = 0.01$ ensures stability. Using this step-size the solution (Figure 3.2) is evidently stable.

3.3 Normal mode oscillations

3.3.1 Initialization

A form of initialization called normal mode initialization (NMI) will be applied to the swinging spring system in section 3.3.2. Firstly, it is important to understand the concept of initialization. The atmosphere consists of a range between very fast changing systems with cycles lasting less than a minute, such as localized pressure variations and very slow changing systems lasting millenia such as climate change (Kasahara, 1976). Weather forecasters are often not interested in the very fast systems. For example, pressure variations with cycles less than a minute are not going to help predict the arrival of a low pressure system at least 24 hours away. The useless high-frequency changes are called noise. If noisy observations are incorporated into the basic equations in NWP the forecast may contain spurious large amplitude high frequency oscillations (Lynch, 2002). The elimination of this noise can be achieved by adjusting the initial fields in a weather forecast, a process called initialization (Lynch, 2002).

In data assimiltation it is important to ensure that the forecast of the previous analysis does not allow large gravity waves to develop. Constraints in the analysis step can be made to ensure that this does not happen.

3.3.2 Normal mode initialization (NMI) of the swinging spring system

The following implementation of NMI to the swinging spring system comes from Lynch [2002]. Both linear and non-linear normal mode initialization are applied. Suppose it is known that high frequency oscillations are not present in the motion of the swinging spring, yet they are still present in the numerical solution due to observational errors. How can we adjust the initial conditions to get rid of the high frequency noise? Linear Normal Mode Initialization (LNMI) aims to get rid of the high frequency oscillations by setting their initial amplitude to zero. Nonlinear Normal Mode Initialization (NNMI) instead sets the initial rates of change of the fast variables to zero. Using the same parameters as for the uninitialized swinging spring (section 3.1), in this experiment LNMI is imposed by setting $r(0) = l$ and $p_r(0) = 0$. NNMI is imposed by setting $\dot{r}(0) = 0$ and $\dot{p}_r(0) = 0$. In the case of

LNMI the initial conditions are $(\theta, p_\theta, r, p_r) = (1, 0, 1, 0)$. For NNMI, to calculate $\dot{p}_r(0) = 0$ set $p_r(0) = 0$ in equation 3.4. For $\dot{p}_r(0) = 0$, we first calculate $\dot{\theta}(0)$ from equation 3.2. This can be substituted in to equation 3.5 and re-arranged to give:

$$r(0) = \frac{l(1 - \epsilon^2[1 - cos\theta(0)])}{1 - (\dot{\theta}(0)/\omega_r)^2} \tag{3.15}$$

Also, to ensure that $\dot{\theta}(0)$ used in 3.15 is consistent with 3.2 and the new value of r(0) we must set:

$$p_\theta(0) = mr(0)^2\dot{\theta}(0) \tag{3.16}$$

Figure 3.3 shows the coordinates and fourier transforms of the swinging spring variables for LNMI. Compared with the uninitialized case (Figure 3.2) the high frequency oscillations have been largely supressed (r and p_r scales are one-tenth those in Figure 3.2). However, it is clear that the slow variables are exciting the fast variables as there is a large peak at $f = 5$, although the amplitudes are much smaller than in Figure 3.2).

The figure showing the NNMI case (Figure 3.4) is even better (p_r is $\frac{1}{4}$ the size than in Figure 3.3) since high frequency oscillations have been massively damped. There is a peak in the fourier transforms in both the LNMI and NNMI graphs at $f = 1 = 2f_\theta$. This is because the spring is stretched twice per angular cycle at the bottom of the swing. This is called 'balanced fast motion'.

3.3.3 Summary of the numerical integration of the swinging spring and NMI

The swinging spring equations were integrated using the standard Matlab ode45 function, as done by Livings [2005]. This uses an explicit Runge-Kutta (4,5) pair with a changeable step-size. It was important to choose a small enough step-size to ensure stability. As discussed by Livings [2005] this had to be smaller than $\frac{T}{20}$, where T is the period of the harmonic oscillator.

Initialization aims to prevent unwanted high-frequency noise from developing in a solution. Linear Normal Mode Initialization (LNMI) aims to get rid of the high frequency oscillations by setting their initial amplitude to zero. Nonlinear Normal Mode Initialization (NNMI) instead sets the initial rates of change of the fast variables to zero (Lynch, 2002). Both were imposed for the swinging spring equations using the method and initial conditions described by Lynch [2002]. It was discovered that NNMI was better at supressing the high-frequency oscillations due to the non-linear nature of the system.

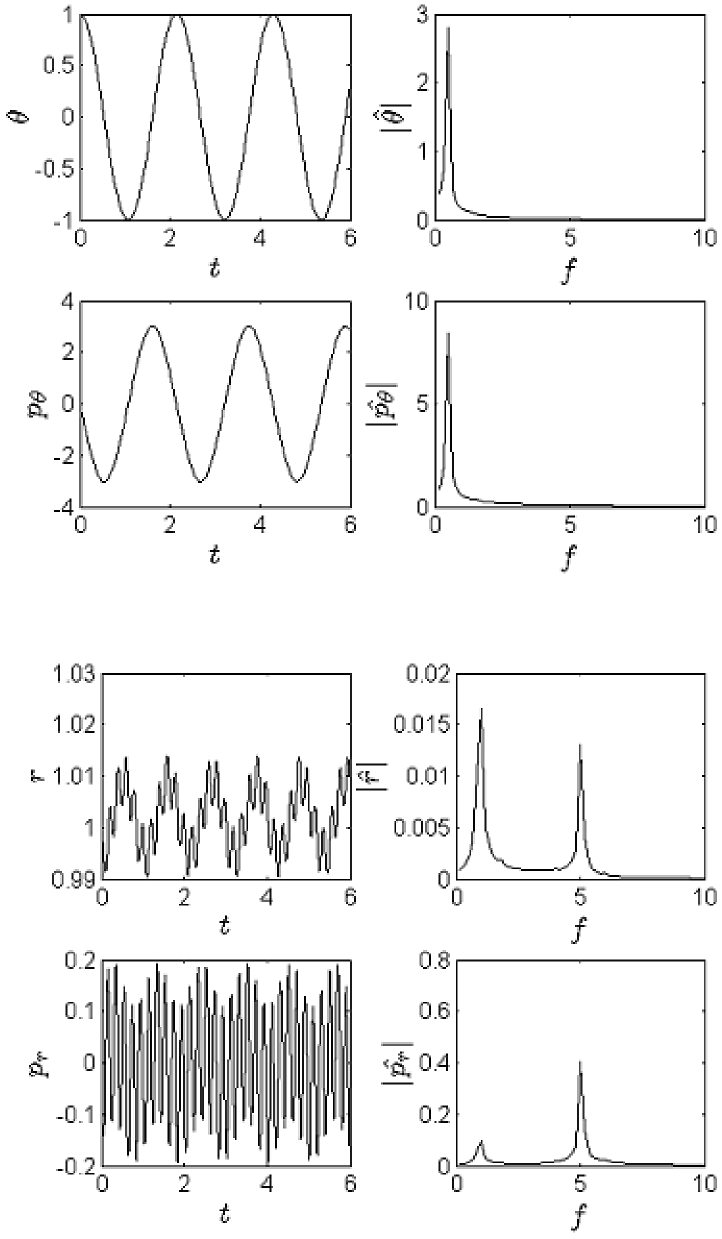

Figure 3.3: Coordinates and their fourier transforms of slow and fast variables for a swinging spring with linear normal mode initialization. Parameter values give rotational and elastic frequencies of $f_\theta = 0.5$ and $f_r = 5$ respectively. Initial conditions are $(\theta, p_\theta, r, p_r) = (1, 0, 1, 0)$.

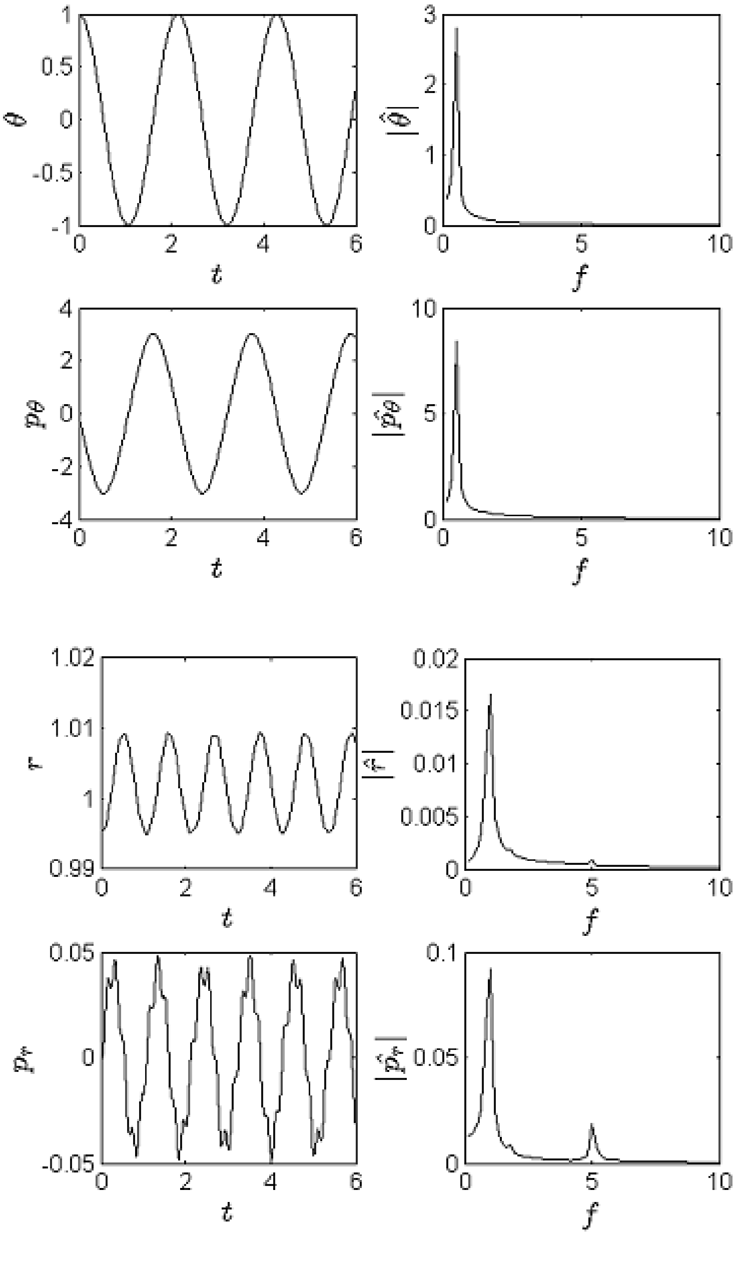

Figure 3.4: Coordinates and their fourier transforms of slow and fast variables for a swinging spring with nonlinear normal mode initialization. Parameter values give rotational and elastic frequencies of $f_\theta = 0.5$ and $f_r = 5$ respectively. Initial conditions are $(\theta, p_\theta, r, p_r) = (1, 0, 0.99540, 0)$.

Chapter 4

Implementing the ETKF, ETKS and ETKS2

The implementation of the ETKF is described using the same method given by Livings [2005]. Then a suggested implementation for the ETKS is given, although this is not used in the experiments for this report. The implementation of an approximation to the ETKS, the ETKS2 (see section 2.9.2) is also given. The ETKF and the ETKS2 are both used in the experimental work for this report (see the results chapter 5 for information about the experiments).

4.1 Implementation of the ETKF

The implementation of the ETKF described here comes from Livings [2005]. The analysis is the same for this method as in section 2.5, but this method is more efficient and more accurate since there are less multiplicative operations, therefore reducing rounding errors. It is also less expensive as it reduces the storing of large matrices such as the Kalman gain matrix (K_e). Firstly, in order to avoid problems with eigenvalue decomposition the scaled forecast observation ensemble perturbation matrix is introduced as

$$\hat{Y}^f = R^{-\frac{1}{2}} Y^f. \tag{4.1}$$

Scaling the observations by the square root of the observation error covariance $R^{-\frac{1}{2}}$ normalises the observations of physical quantities with differing standard deviations in error. This reduces possible loss of accuracy due to rounding errors (Livings, 2005). Writing

$$(Y^f)^T R^{-1} Y^f = (\hat{Y}^f)^T \hat{Y}^f \tag{4.2}$$

then the eigenvalue decomposition is

$$(\hat{Y}^f)^T \hat{Y}^f = U \Lambda U^T \tag{4.3}$$

where U is an $N \times N$ orthogonal matrix. It is possible to avoid performing $(Y^f)^T Y^f$ and use a simpler matrix in the analysis equation, which is more efficient to compute. It also reduces the errors in the smallest eigenvalues (Golub and Van-Loan, 1996). This is done by using the Singular Value Decomposition (SVD) of Golub and Van-Loan [1996]:

$$(\hat{Y}^f)^T = U \Sigma V^T \tag{4.4}$$

Here, Σ is the $N \times p$ diagonal matrix satisfying $\Lambda = \Sigma\Sigma^T$ and V is an $p \times p$ orthogonal matrix. The ensemble perturbation matrix is updated by

$$X^a = X^f T = X^f U (I + \Lambda)^{-\frac{1}{2}} U^T \tag{4.5}$$

The matrix U^T has been added to the version in (Livings, 2005) to make it unbiased (Livings et al., 2008). The ensemble Kalman gain can now be written as

$$\begin{aligned} K_e &= X^f (Y^f)^T (Y^f (Y^f)^T + R)^{-1} \\ &= X^f \hat{Y^f}^T (\hat{Y}^f (\hat{Y}^f)^T + I)^{-1} R^{-\frac{1}{2}} \\ &= X^f U \Sigma (\Sigma^T \Sigma + I)^{-1} V^T R^{-\frac{1}{2}}. \end{aligned} \tag{4.6}$$

The product vector

$$\underline{z} = \Sigma(\Sigma^T \Sigma + I)^{-1} V^T R^{-\frac{1}{2}} (\underline{y} - \overline{\underline{y^f}}) \tag{4.7}$$

is built from right to left. This avoids storing the large K_e matrix. Also, it involves the calculation of matrix-vector products, rather than matrix-matrix products, making it more computationally efficient. The ensemble mean is updated as

$$\overline{\underline{x^a}} = \overline{\underline{x^f}} + X^f U \underline{z}. \tag{4.8}$$

4.2 Implementation of the ETKS and ETKS2 methods

The following implementation for the ETKS and ETKS2 methods use the implementation method of Cohn et al. [1994] for a fixed-lag smoother. In the experimental results chapter 5 the ETKS2 is compared with the ETKF. The ETKS2 is validated by checking that the final analysis ensemble in the time-window is equivalent to the ETKF. Also, the ensemble mean error over the time window should be less for the ETKS2. Both the ETKS and ETKS2 implementations have loops which perform a re-analysis process. Firstly it is important to have calculated the analyses and forecasts for the ETKF in the time interval, as this is used as initial data. The implementation for the ETKF is given in the previous section 4.1.

4.2.1 Description of the ETKS and ETKS2 algorithms

The time interval is $[0, T]$ with I steps. Assume H does not vary with time. The subscript i represents the ensemble members $i = 1, 2, 3...N$. The idea in the loop is to update the filtered analyses with all future observations one by one. The lag l is the time after the current observation time, at which future observations are used to smooth the current analyses. Note that the final filtered analysis and smoothed analysis will be the same. Two loops are created. The first one creates the lag times with which to smooth. The second loop is the time-step t_k. Thus all times less than $I - l$ are smoothed with future observations at time $k + l$. The ensemble perturbation matrix is updated by the equation

$$X^a_{k|k+l} = X^a_{k|k+l-1}T_{k+l|k+l-1} = X^a_{k|k+l}U_{k+l|k+l-1}(I + \Lambda_{k+l|k+l-1})^{-\frac{1}{2}}U^T_{k+l|k+l-1} \tag{4.9}$$

where U, Λ and Σ are calculated from the Singular Value Decomposition (SVD) of Golub and Van-Loan [1996] as in the previous section 4.1 for the ETKF implementation. The Kalman gain K is written as:

$$K_{k|k+l} = X^a_{k|k+l-1}U_{k+l|k+l-1}\Sigma_{k+l|k+l-1}(\Sigma^T_{k+l|k+l-1}\Sigma_{k+l|k+l-1} + I)^{-1}V^T_{k+l|k+l-1}R^{-\frac{1}{2}}_{k+l|k+l-1}. \tag{4.10}$$

The product vector is built according to

$$\underline{z}_{k+l|k+l-1} = \Sigma_{k+l|k+l-1}(\Sigma^T_{k+l|k+l-1}\Sigma_{k+l|k+l-1} + I)^{-1}V^T_{k+l|k+l-1}R^{-\frac{1}{2}}_{k+l|k+l-1}(\underline{y}_{k+l} - \overline{\underline{y}^f_{k+l|k+l-1}}) \tag{4.11}$$

as in section 4.1. The ensemble mean can then be updated by

$$\overline{\underline{x}^a_{k|k+l}} = \overline{\underline{x}^a_{k|k+l-1}} + X^a_{k|k+l}U_{k+l|k+l-1}\underline{z}_{k+l|k+l-1}. \tag{4.12}$$

Then the individual ensemble members can be updated as:

$$\underline{x}^a_{i,k|k+l} = \overline{\underline{x}}^a_{i,k|k+l} + \underline{x}'_{i,k|k+l}{}^a. \tag{4.13}$$

For the ETKS algorithm the new forecast ensemble members are then updated using the smoothed analysis as

$$\underline{x}^f_{i,k+1|k+l} = m(\underline{x}^a_{i,k|k+l}). \tag{4.14}$$

The ETKS2 algorithm is the same as the ETKS algorithm except the forecast is not updated. The following sections show the ETKF algorithms for the ETKS and ETKS2.

4.2.2 ETKS algorithm

Firstly calculate and store the ETKF results using the implementation from section 4.1. This is the initial data for $\underline{x}^f_{i,k}$ and $\underline{x}^a_{i,k}$, $i = 1, 2, 3, ...N$, $k = 1, 2, 3...I$.
Create lag time l after the current time with which to smooth:
Loop from $l = 1$ to I
Smooth all times less than $I - l$ with future observations at lag time l:
Loop from $k = 1$ to $I - l$
Calculate the forecast ensemble perturbation matrix:

$$X^f_{k+l|k+l-1} = \frac{1}{\sqrt{N-1}} \begin{pmatrix} \underline{x}^f_{1,k+l|k+l-1} - \overline{\underline{x}}^f_{k+l|k+l-1} & \cdots & \underline{x}^f_{N,k+l|k+l-1} - \overline{\underline{x}}^f_{k+l|k+l-1} \end{pmatrix}.$$

Calculate the observational ensemble perturbation matrix:

$$Y^f_{k+l|k+l-1} = HX^f_{k+l|k+l-1}$$

Calculate $\hat{Y}^f_{k+l|k+l-1}$:

$$\hat{Y}^f_{k+l|k+l-1} = \frac{1}{\sqrt{N-1}} R^{-\frac{1}{2}}_{k+l|k+l-1} Y^f_{k+l|k+l-1}.$$

Calculate the singular value decomposition (SVD) of $Y^f_{k+l|k+l-1}$:

$$(U, \Sigma, V) = SVD((\hat{Y}^f_{k+l|k+l-1})^T)$$

Calculate the ensemble analysis perturbation matrix using SVD:

$$X^a_{k|k+l} = X^a_{k|k+l-1} U(I + \Sigma\Sigma^T)^{-\frac{1}{2}} U^T$$

Calulate the product vector $z_{k+l|k+l-1}$:

$$\underline{z}_{k+l|k+l-1} = \Sigma(\Sigma^T\Sigma + I)^{-1} V^T R^{-\frac{1}{2}}_{k+l|k+l-1} (\underline{y}_{k+l} - \overline{\underline{y}^f}_{k+l})$$

Update the analysis ensemble mean:

$$\overline{\underline{x}^a_{i,k|k+l}} = \overline{\underline{x}^a_{i,k|k+l-1}} + X^a_{k|k+l}U\underline{z}.$$

Update the analysis state using the ensemble mean and perturbations:

$$\underline{x}^a_{i,k|k+l} = \overline{\underline{x}^a_{i,k|k+l}} + {\underline{x}'_{i,k|k+l}}^{a}.$$

Update the forecast state for time t_{k+1} using the smoothed analysis at time t_k:

$$x^f_{i,k+1|k+l} = m(\underline{x}^a_{i,k|k+l})$$

end loop
end loop

4.2.3 ETKS2 algorithm

The only difference between the ETKS and ETKS2 algorithms is that the ETKS2 forecast states are not updated using the smoothed analysis states. Firstly calculate and store the ETKF results using the implementation from section 4.1. This is the initial data for $\underline{x}^f_{i,k}$ and $\underline{x}^a_{i,k}$), $i = 1, 2, 3, ...N$, $k = 1, 2, 3...I$.

Create lag time l after the current time with which to smooth:
Loop from $l = 1$ to I
Smooth all times less than $I - l$ with future observations at lag time l:
Loop from $k = 1$ to $I - l$
Calculate the forecast ensemble perturbation matrix:

$$X^f_{k+l|k+l-1} = \frac{1}{\sqrt{N-1}} \left(\underline{x}^f_{1,k+l|k+l-1} - \overline{\underline{x}}^f_{k+l|k+l-1} \quad \cdots \quad \underline{x}^f_{N,k+l|k+l-1} - \overline{\underline{x}}^f_{k+l|k+l-1} \right).$$

Calculate the observational ensemble perturbation matrix:

$$Y^f_{k+l|k+l-1} = HX^f_{k+l|k+l-1}$$

Calculate $\hat{Y}^f_{k+l|k+l-1}$:

$$\hat{Y}^f_{k+l|k+l-1} = \frac{1}{\sqrt{N-1}} R^{-\frac{1}{2}}_{k+l|k+l-1} Y^f_{k+l|k+l-1}.$$

```
Calculate the singular value decomposition (SVD) of $Y^f_{k+l|k+l-1}$:
    $(U, \Sigma, V) = SVD((\hat{Y}^f_{k+l|k+l-1})^T)$
Calculate the ensemble analysis perturbation matrix using SVD:
    $X^a_{k|k+l} = X^a_{k|k+l-1}U(I + \Sigma\Sigma^T)^{-\frac{1}{2}}U^T$
Calulate the product vector $z_{k+l|k+l-1}$:
    $\underline{z}_{k+l|k+l-1} = \Sigma(\Sigma^T\Sigma + I)^{-1}V^T R^{-\frac{1}{2}}_{k+l|k+l-1}(\underline{y}_{k+l} - \overline{\underline{y}^f}_{k+l})$
Update the analysis ensemble mean:
    $\overline{\underline{x}^a_{i,k|k+l}} = \overline{\underline{x}^a_{i,k|k+l-1}} + X^a_{k|k+l}U\underline{z}.$
Update the analysis state using the ensemble mean and perturbations:
    $\underline{x}^a_{i,k|k+l} = \overline{\underline{x}^a_{i,k|k+l}} + {\underline{x}'_{i,k|k+l}}^a.$

There is no forecast update for the ETKS2
end loop
end loop
```

4.2.4 Statistical significance method and the ensemble mean error

Since the normal distribution used to generate the initial ensemble analysis is pseudo-random, the ETKF and ETKS2 ensemble analysis over time would be expected to vary in accuracy according to this initial analysis. Therefore when analysing the general behaviour of the ETKF or ETKS2 it is better to take an average over a number of unseeded runs. For the experiments in section 5 it was decided that 100 runs was enough. The new notation used for the 100 run averages is described and the statistical significance method is described, which is used to decide why 100 runs was enough.

Notation

An analysis ensemble averaged over 100 runs is given as $\underline{x}^a_{i,k,\overline{100}}$, where the subscripts $i = 1, 2, 3, ...N$ are the ensemble members, k is the time-step and $\overline{100}$ means it the average over 100 runs. The ensemble mean averaged over 100 runs is given as $\overline{\underline{x}^a_{k,\overline{100}}}$ (the i subscript is no longer there). The analysis ensemble mean averaged over 100 runs and averaged over the time-interval is given by $\overline{\underline{x}^a_{k,\overline{100}}}$, where $\overline{k}$ is the average over the time-interval. $\overline{\underline{x}^a_{k,\overline{100}}}$ gives a single value in each coordinate.

The analysis ensemble mean absolute error is defined as

$$\overline{\underline{e}^a_k} = (|\overline{\underline{x}^a_k} - \underline{x}^t_k|). \tag{4.15}$$

Using the same notation as $\overline{\underline{x}^a_{k,\overline{100}}}$, the ensemble mean absolute error averaged over 100 runs and averaged over the time interval is $\overline{\underline{e}^a_{k,\overline{100}}}$ and gives a single value in each coordinate.

Statistical significance

The statistical significance of the analysis ensemble mean error averaged over 100 runs was measured in the following way. Firstly, the standard deviation of the analysis ensemble absolute error averaged over 100 runs ($\underline{\sigma}(\underline{e}^a_{k,\overline{100}})$) was calculated. This standard deviation was calculated from

$$\underline{\sigma}(\underline{e}^a_{k,\overline{100}}) = (\frac{1}{N}\sum_{i=1}^{N}(\underline{e}^a_{i,k,\overline{100}} - \overline{\underline{e}^a_{k,\overline{100}}})^2)^{\frac{1}{2}}. \tag{4.16}$$

This standard deviation was then averaged over the time interval to give a single value in each coordinate ($\underline{\sigma}(\underline{e}^a_{k,\overline{100}})$). The statistical significance measure used here is in the form of a confidence interval, as described by Mood and Greybill [1963]. The aim is to be 95% sure that the expectation of the analysis error lies within a certain interval of the ensemble mean ($\overline{\underline{e}^a_{k,\overline{100}}}$). The smaller this interval the higher the level of confidence that the ensemble mean error averaged over 100 runs is a good approximation of the expected behaviour of the ETKF or ETKS2. Using a cumulative distribution function for a normal distribution and the derivation given in Mood and Greybill [1963] (see section 11.7, pages 262-264) the 95% confidence interval is estimated as

$$\left(\overline{\underline{e}^a_{k,\overline{100}}} - \frac{1.96\underline{\sigma}(\underline{e}^a_{k,\overline{100}})}{\sqrt{N}}, \overline{\underline{e}^a_{k,\overline{100}}} + \frac{1.96\underline{\sigma}(\underline{e}^a_{k,\overline{100}})}{\sqrt{N}}\right). \tag{4.17}$$

4.2.5 Summary of ETKF and ETKS/ETKS2 implementations and statistical significance

The ETKF is implemented using the same method of Livings [2005]. It is more efficient and more accurate than the method described in section 2.5 since there are less multiplicative operations. A product vector is built from right to left (equation 4.7) and saves having to calculate the expensive Kalman gain matrix. The ETKS and ETKS2 algorithms use the ETKF filter run as initial conditions. Then the analyses are smoothed using future observations. The implementaion for the smoothing process uses the features of the ETKF implementaion to make it more efficient, such as he tproduct vector. The key difference between the ETKS and ETKS2 algorithms is that the ETKS algorithm updates the forecast states from the smoothed analyses. This is important in calculating the smoothed innovation vector and square root matrix. The ETKS 2 uses the original filtered forecast state in the innovation vector and square root matrix.

In order to determine the general behaviour of the ETKF/ETKS2 in the experimental work (section 5) it is better to average over a number of unseeded runs. Using the statistical significance test from Mood and Greybill [1963] a 95% confidence interval is given for a set of normally distributed data. This determines a 95% chance that the expected value lies within a certain boundary.

Chapter 5

Results

The methodology described in chapter 4 is now applied to the Ensemble Transform Kalman Filter (ETKF - see section 4.1) and the Ensemble Transform Kalman Smoother 2 (ETKS2 - see section 4.2.3). The ETKS2 differs from the ETKS (section 4.2.2) in that it does not smooth the forecast state in the innovation vector and square root matrix T but instead uses the filtered version. The NNMI trajectory from section 3.3 is used as the true trajectory. The model used in the forecast step is the same as the model used to generate the true trajectory. The ETKS2 is validated by checking the final ensemble values for a seeded initial distribution are equivalent to the final ensemble values of the ETKF. Also the ensemble mean error averaged over the time interval should be smaller for the ETKS2. The experiments differ in the analysis step, the ensemble size and whether the observations are perfect (noise free) or imperfect. The fixed-lag smoother ETKS2 is also tested with different lags.

5.1 Comparison of the ETKF with fixed-interval ETKS2 using perfect observations

5.1.1 ETKF

The following application of the ETKF uses the programme created by Livings [2005]. Firstly, the results using the ETKF will be observed for frequent, perfect observations on all four coordinates $(\theta, p_\theta, r, p_r)$. The first observation is at time 0.1 and all subsequent observations have intervals of 0.1. Since the observations are perfect, the actual observation errors are zero. The covariance matrix R passed to the filter has uncorrelated errors with the standard deviations listed in table 5.1 and uses the same values as in Livings [2005]. These standard deviations are close to one-tenth of the amplitude of the oscillations in the true solution. The covariance matrix R is also used in generating the initial ensemble. An ensemble of pseudo random vectors is drawn from a normal distribution with the given covariance matrix as its covariance matrix and the true initial state as its mean. This ensemble is then translated slightly so that the ensemble mean coincides exactly with the initial state. This translated random ensemble is used as an initial ensemble (Livings, 2005).

For a fair comparison between the ETKF and ETKS2 it is necessary to use an identical (seeded) initial ensemble distribution for both runs. The result of an experiment with ensemble size $N = 10$ is shown in Figure 5.1. The graph shows the difference between the ETKF analysis and the truth for all four coordinates. Error bars show the observations (centred on the true solution) with the radius representing the standard deviation passed to the filter. The observations follow the truth since they are perfect. There is a decrease in the amplitudes of the filter error over time for most of the ensemble members, most significantly over the first few seconds. This is expected since the analysis becomes more accurate (close to the true solution) as more observations are assimilated. The values of the ensemble errors at the final time (t=6) are in table 5.2. Clearly the errors are all smaller for each coordinate than the observation error standard deviations in table 5.1 and are generally less than a tenth of the magnitude. There is variability in the ensemble member errors for each coordinate but generally they are the same order of magnitude.

Coordinate	Standard deviation
θ	0.1
p_θ	0.3
r	7×10^{-4}
p_r	5×10^{-3}

Table 5.1: Observation error standard deviations passed to the filter in experiments with perfect observations. These standard deviations are also used to generate the initial ensemble.

As described in section 4.2.4 when trying to determine the general behaviour of the ETKF or ETKS2 it is better to take an average over a number of unseeded runs. The statistical significance test described in section 4.2.4 was used when determining the ensemble mean absolute error averaged over 100 runs and averaged over the time interval ($\overline{\underline{e}^{a}_{k,100}}$). Table 5.4 shows the ETKF value of $\overline{\underline{e}^{a}_{k,100}}$ in each coordinate using perfect observations. The 95% confidence range is also shown. For example, for the ETKF θ observation there is a 95% chance that when calculating $\overline{\underline{e}^{a}_{k,100}}$ that the expectation of the error in θ lies in the range $9.40 \times 10^{-3} \pm 1.11 \times 10^{-4}$. Since the 95% confidence range is always less than

Ensemble member	θ	p_θ	r	p_r
1	-0.0170	0.0413	0.0004	-0.0001
2	0.0029	-0.0181	0.0000	-0.0004
3	-0.0083	0.0051	0.0001	0.0022
4	-0.0045	0.0083	0.0001	-0.0004
5	0.0038	0.0049	0.0001	0.0005
6	0.0055	-0.0080	0.000	-0.0001
7	0.0076	-0.0299	-0.0001	-0.0012
8	0.0050	-0.0001	0.0000	-0.0012
9	-0.0021	0.0191	0.0001	0.0017
10	0.0001	0.0116	0.0001	-0.0013

Table 5.2: Errors of the ETKF 10 ensemble members for the seeded distribution at t=6 using perfect observations.

$\frac{1}{10}$ the size of $\overline{\underline{e}^a_{k,100}}$, 100 runs is considered a good enough average. The same statistical significance test is given to the ETKF and ETKS2 experiments for perfect and imperfect observations when measuring $\overline{\underline{e}^a_{k,100}}$.

Figure 5.2 shows the absolute value of the ensemble mean error over time for the ETKF, averaged over 100 filter runs ($\overline{\underline{e}^a_{k,100}}$). This shows clearly the fall in error across all four coordinates before levelling off after about one second.

The analysis ensemble standard deviation was averaged over 100 runs and averaged over the time interval ($\underline{\sigma}(\overline{\underline{x}^a_{k,100}})$). The values for each coordinate are shown in table 5.3. Notice that these values are all less than 20% the size of the ensemble mean absolute error in table 5.4. A smaller standard deviation means a clearer indication of the behaviour of the ensemble mean analysis averaged over a number of runs. However, this is not the same as saying the ensemble mean analysis is more accurate for a smaller standard deviation. It is well known that the ETKF usually underestimates ensemble spread (see for example Ehrendorfer [2007]). The standard deviation should be large enough to reflect the true uncertainty.

Experiment	θ	p_θ	r	p_r
ETKF, perfect obs, $N = 10$	1.02×10^{-3}	3.22×10^{-3}	1.43×10^{-5}	8.50×10^{-5}
ETKS2, perfect obs, $N = 10$	6.86×10^{-4}	2.08×10^{-3}	9.66×10^{-6}	5.60×10^{-4}
ETKF, imperfect obs, $N = 10$	1.77×10^{-3}	7.31×10^{-3}	3.67×10^{-4}	1.06×10^{-2}
ETKS2, imperfect obs, $N = 10$	8.44×10^{-4}	1.96×10^{-3}	6.09×10^{-4}	2.06×10^{-2}

Table 5.3: Standard deviation of the analysis ensemble averaged over 100 runs and averaged over the time interval for the ETKF and ETKS2 using perfect and imperfect observations.

Experiment (perfect obs)	θ	p_θ	r	p_r
ETKF $\overline{\underline{e}_{\bar{k},\overline{100}}}^a$	9.40×10^{-3}	2.53×10^{-2}	2.01×10^{-4}	1.43×10^{-3}
95% confidence	$\pm 1.11 \times 10^{-4}$	$\pm 2.81 \times 10^{-4}$	$\pm 4.30 \times 10^{-7}$	$\pm 3.44 \times 10^{-5}$
ETKS2 $\overline{\underline{e}^a_{\bar{k},\overline{100}}}$	4.77×10^{-3}	1.34×10^{-2}	7.71×10^{-5}	7.48×10^{-4}
95% confidence	$\pm 7.31 \times 10^{-5}$	$\pm 2.09 \times 10^{-4}$	$\pm 1.43 \times 10^{-6}$	$\pm 1.70 \times 10^{-6}$

Table 5.4: Time averaged errors of the ensemble mean ± the 95% confidence range. Computed from 100 runs of the ETKF and ETKS2 with different random initial conditions and perfect observations.

5.1.2 Fixed-interval ETKS2

The following application of the ETKS2 uses the implementation in section 2.9.2 and smoothes using all the available obervations, hence it is fixed-interval. The ETKS2 uses the ETKF filter runs calculated in the programme created by Livings [2005]. A new section of the programme calculates the smoothed analyses. Using the same seeded distribution as used for the ETKF in 5.1, the results for the ETKS2 run are shown in Figure 5.3. Clearly the initial errors for the ETKS2 are much less. Also there is no general trend in the magnitude of the errors in the time window. The magnitudes of the errors for one ensemble member (blue line) appear to increase over the time interval but this is an isolated case. At the end of the time window the ETKS2 errors for each ensemble member are the same as the ETKF errors. This is shown by comparing table 5.2 with table 5.5. This is the expected result since both the final analysis of the ETKF and the final analysis of the ETKS2 are calculated from the same forecast state and observations.

The standard deviation of the analysis ensemble for the ETKS2 averaged over 100 runs was averaged over the time interval for all four coordinates. The values are shown in table 5.3. Compared with the ETKF standard deviations in table 5.3 these are at least an order of magnitude smaller in all four coordinates and are of order 10^{-3} or less. This smaller standard deviation also reveals less variability in the ensemble members for the ETKS2 and thus a smoother solution.

Figure 5.4 shows the absolute value of the error over time using the ensemble mean smoothed analysis averaged over 100 runs for the ETKS2. Comparing Figure 5.4 with Figure 5.2 it is evident that the ETKS2 errors are much smaller near the beginning of the time-window than the ETKF ones. Since the smoothed analyses assimilate all the observations in the time-window the accuracy is expected to be similar at each time, which is the case here. Also, as expected the final analysis for the filter and the smoother have a similar error since they both assimilate all the observations. Table 5.4 shows that the time averaged errors of the ensemble mean for the ETKS2 are smaller than for the ETKF in all four coordinates. The ETKS2 errors are all approximately half the size of the ETKF errors. Thus the ETKS2 solution is smoother and more accurate on average than the ETKF one.

Ensemble member	θ	p_θ	r	p_r
1	-0.0170	0.0413	0.0004	-0.0001
2	0.0029	-0.0181	0.0000	-0.0004
3	-0.0083	0.0051	0.0001	0.0022
4	-0.0045	0.0083	0.0001	-0.0004
5	0.0038	0.0049	0.0001	0.0005
6	0.0055	-0.0080	0.000	-0.0001
7	0.0076	-0.0299	-0.0001	-0.0012
8	0.0050	-0.0001	0.0000	-0.0012
9	-0.0021	0.0191	0.0001	0.0017
10	0.0001	0.0116	0.0001	-0.0013

Table 5.5: Errors of the ETKS2 10 ensemble members for the seeded distribution at t=6 using perfect observations.

5.2 Comparison of the ETKF with the ETKS2 using imperfect observations

5.2.1 ETKF

The following imperfect method for the system is equivalent to the method implemented by Livings [2005]. So far in the experiments, the observations have been noise-free, frequent and made on all four coordinates of the system. For the next experiment, these assumptions are relaxed so as to see the effect on the ensemble statistics. For the next experiment the time interval between observations has been increased to 0.37. Not only is this larger than the previous interval of 0.1 but it has been chosen as it is not a sub-multiple of the oscillation periods $T_\theta = 2$ and $T_r = 0.2$. Instead of observing all four coordinates only θ is observed. The observation error standard deviation added to the filter for θ is 0.1 as before, but now errors of this magnitude are really added to the observations. The initial ensemble is generated using a diagonal covariance matrix corresponding to the standard deviations in table 5.7. The standard deviation for θ is the same as that used for observations. The standard deviations for the other coordinates are equivalent to the amplitudes of the uninitialised oscillations in Figure 3.2. The idea is that the initial ensemble is not influenced by these coordinates. The initial ensemble is generated using pseudo-random vectors as with the perfect observations. However, there is no final translation to make the ensemble mean coincide exactly with the true initial state.

Firstly an ETKF run with a seeded initial ensemble distribution will be looked at for 10 ensemble members. This seeded distribution is the same used for the perfect case (Figure 5.1). It is shown in Figure 5.5. It is obvious when comparing figure 5.5 with 5.1 that the amplitudes of the errors are much larger for the imperfect case. Also, the errors do not decrease over time in the fast coordinates as they do for the perfect case, instead they continue to oscillate at a similar magnitude. In the slow coordinates there is a slight decrease in the errors over time. The timescale for the fast coordinates is also much slower in the imperfect case than the perfect case.

The time averaged analysis ensemble standard deviation over 100 runs for the ETKF using imperfect observations is shown in table 5.3. Compared with the ETKF using perfect observations the standard deviations of all four coordinates are higher. The higher standard deviations in the imperfect case imply more variability of the ensemble members than the perfect case. This is especially apparent in the fast coordinates.

Figure 5.6 shows the 100 run averaged ensemble errors for the ETKF. It is evident that the magnitude of the errors is much larger for all four coordinates. There is also a less rapid decrease in filter error over time than for the perfect case in the slow coordinates. The fast coordinates errors again appear to oscillate at the same magnitude throughout the time-window.

The time averaged errors of the ensemble mean shown in table 5.6 are significantly larger than in the perfect case, especially in the fast coordinates. The fast coordinate errors would be expected to be higher since they are not observed.

Experiment (imperfect obs)	θ	p_θ	r	p_r
ETKF $\overline{\underline{e}^{a}_{k,\overline{100}}}$	7.03×10^{-2}	2.12×10^{-1}	3.48×10^{-2}	1.09
95% confidence	$\pm 1.62 \times 10^{-3}$	$\pm 3.71 \times 10^{-3}$	$\pm 1.30 \times 10^{-3}$	$\pm 4.38 \times 10^{-2}$
ETKS2 $\overline{\underline{e}^{a}_{k,\overline{100}}}$	4.25×10^{-2}	1.20×10^{-1}	3.36×10^{-2}	1.06
95% confidence	$\pm 5.23 \times 10^{-4}$	$\pm 1.21 \times 10^{-3}$	$\pm 3.77 \times 10^{-4}$	$\pm 1.28 \times 10^{-2}$

Table 5.6: Time averaged errors of the ensemble mean ± the 95% confidence range. Computed from 100 runs of the ETKF and ETKS2 with different random initial conditions. The imperfect observations have different random observation errors.

Coordinate	Standard deviation
θ	0.1
p_θ	3
r	0.06
p_r	1.5

Table 5.7: Observation error standard deviations passed to the filter in experiments with imperfect observations. These standard deviations are also used to generate the initial ensemble.

5.2.2 Fixed-interval ETKS2

Using the seeded initial distribution, the ETKS2 was run using imperfect observations. The ETKS2 run in Figure 5.7 shows no significant change in the error magnitudes over the time window. The errors are also smaller across all 4 coordinates than for the ETKF run in Figure 5.5.

Figure 5.8 shows the absolute value of the error over time using the ensemble mean smoothed analysis averaged over 100 filter runs for the ETKS2 for the imperfect observations. Comparing Figures 5.8 and 5.6 it is noticeable that the ETKS2 errors are only significantly better than the ETKF errors for the slow coordinates. This would make sense sinse the slow coordinate θ is the only variable being observed. Hence the smoothing process in ETKS2 is going to be more beneficial in the slow coordinates since actual future observations are being assimilated. However, the acccuracy of the slow variables will have some influence on the accuracy of the fast variables in the equations of motion (section 3.1). Comparing Figure 5.8 with Figure 5.4 shows that the ETKS2 has also lost a lot of accuracy when using imperfect observations instead of perfect ones. Table 5.6 shows that the time averaged errors of the ensemble mean for the ETKS2 are smaller than for the ETKF in all four coordinates when using imperfect observations. However, as discussed this is only significant for the slow coordinates.

5.2.3 Ensemble error of the ETKF and ETKS2 using different ensemble sizes with imperfect observations

Table 5.8 shows the time averaged errors of the ensemble mean (averaged over 100 runs) in all four coordinates of the ETKF for 2-10 ensemble members. The ETKS2 version is shown in table 5.9. As expected the ETKS average errors are smaller than the ETKF ones.

Both the ETKF and ETKS2 errors in all four variables decrease significantly when increasing the ensemble size from two to four. When the ensemble size is too small in relation to the model problems can develop in the filter. The ensemble must adequately span the model subspace. The smaller the ensemble is, the greater the chance of sampling errors and underestimated forecast error covariances or imbreeding (Ehrendorfer, 2007). Thus increasing the number of ensemble members from two to four improves the accuracy in the four-dimensional swinging spring model. However, the errors decrease less when increasing the size from four to six and when increasing the ensemble size further than 6 the average errors in all four coordinates are generally getting worse (see tables 5.8 and 5.9). Thus increasing the ensemble size above the number of model variables eventually increases the error. The following reasons given come from (Nichols, 2009). The state of the system lies in an n-dimensional space. The estimate of the state obtained by the ensemble method lies in the p-dimensional space spanned by the ensemble members. When $p > n$, then you have spanned the full space in which the states lie. As you increase the number of ensembles, you are adding dependent vectors to the basis for the space in which the state estimate lies. It is possible therefore that you end up using a very poor selection of vectors from the basis (nearly dependent) to form your new state estimate. The more vectors added to the basis (the larger the number of ensembles) the more likely this is to happen. This effect is exacerbated by the round-off errors introduced in the calculations. These mean that your computation of the state estimate can be very poor, even if the basis is not a linearly dependent set, but is close to being dependent. Round-off error can also cause the basis derived from the ensembles to become dependent before you have reached the point where $p > n$, even though in exact arithmetic this should be independent.

As well as measuring the affect of increasing the ensemble size on the error, it is important to see if the ensemble standard deviation is behaving as it should be. However, this was only looked at for the ETKF. Using the Matlab programme of Livings [2005] it was possible to calculate the fraction of analyses having an ensemble mean within one standard deviation of the truth for each coordinate, averaged over 100 runs. For unbiased, normally distributed analysis errors with standard deviation equal to the ensemble standard deviation, this fraction should be about 0.68 (see Kreyszig, 1999 Appendix 5, table A7). If the fraction is below 0.68 this suggests that the standard deviation could be too small. Table 5.10 shows that this fraction increases significantly for larger ensemble sizes and goes from being well below 0.68 in all coordinates to being well above this fraction in all 4 coordinates. However, the very large fractions of 1.00 seen in the fast coordinates may be due to ignorance of the imperfect ETKF in the fast coordinates as the fast coordinates are not observed (Livings, 2005). It is above 0.68 in all coordinates for ensemble sizes of 6 or larger.

Ensemble size	θ	p_θ	r	p_r
2	0.1183	0.3464	0.0580	1.8381
4	0.0592	0.1823	0.0411	1.3073
6	0.0584	0.1795	0.0382	1.2170
8	0.0589	0.1818	0.0398	1.2672
10	0.0703	0.2124	0.0348	1.0880
50	0.0604	0.1873	0.0427	1.3580

Table 5.8: Time averaged errors of the ensemble mean for different ensemble sizes. Ensemble mean averaged from 100 runs of the ETKF with different random initial conditions.

Ensemble size	θ	p_θ	r	p_r
2	0.1156	0.3122	0.0553	1.7489
4	0.0406	0.1103	0.0395	1.2478
6	0.0390	0.1055	0.0355	1.1707
8	0.0391	0.1066	0.0394	1.2484
10	0.0425	0.1201	0.0336	1.0589
50	0.0392	0.1047	0.0426	1.3529

Table 5.9: Time averaged errors of the ensemble mean for different ensemble sizes. Ensemble mean averaged from 100 runs of the ETKS2 with different random initial conditions.

5.2.4 ETKS2 with varying lags using imperfect observations

So far, the ETKS2 experiments have been of fixed-interval type, smoothing the analyses using all available observations in the time window. Now the ETKS2 will be run for varying lags. Figure 5.9 is a graph plotted with one quarter the lag of the fixed-interval problem. In other words the analyses for the ETKS2 will be smoothed using a maximum of 4 future observations instead of a maximum of 16 observations (all observations available) in the fixed-interval problem. The graph shows that the errors are significantly higher in the slow coordinates at the beginning of the time window, but are more even in the fast coordinates. This is expected since the forecast is carrying forward less information from θ observations from the past earlier in the time-window. The effect is less noticeable in the fast coordinates since they are not observed. Table 5.11 shows the absolute value of the error over time using the ensemble mean smoothed analysis averaged over 100 runs for the different lags. Halving the lag from 16 to 8 increases the average error, except in the θ component. Decreasing the lag further increases the error in all four coordinates. This is expected since less observations are being assimilated at the beginning of the time-window for lower lags.

Ensemble size	θ	p_θ	r	p_r
2	0.282	0.281	0.0678	0.0679
4	0.640	0.634	0.530	0.530
6	0.699	0.690	0.785	0.786
8	0.722	0.717	0.906	0.906
10	0.0701	0.700	0.954	0.954
50	0.754	0.764	1.000	1.000

Table 5.10: Fraction of analyses with ensemble mean within one ensemble standard deviation of the truth. Computed from 100 runs of the ETKF with different random initial conditions. The observations are imperfect and include random observation errors.

Lag	θ	p_θ	r	p_r
1	0.0571	0.1595	0.0402	1.2817
4	0.0466	0.1290	0.0400	1.2698
8	0.0411	0.1101	0.0398	1.2623
Fixed interval=16	0.0425	0.1201	0.0336	1.0589

Table 5.11: Time averaged errors of the ensemble mean for different lags. Ensemble mean averaged from 100 runs of the ETKS2 with different random initial conditions.

5.3 Summary of results

To summarise, the ETKF and fixed-interval ETKS2 were tested using the swinging spring model for perfect and imperfect observations. For both the ETKF and fixed-interval ETKS2 the mean ensemble errors were smaller for the perfect case. The ETKS2 mean ensemble errors were smaller than the ETKF ones for both the perfect and imperfect case. This was expected since the smoother assimilates all the available observations at each analysis step.

Increasing the ensemble members up to the model dimension improved the accuracy of the ETKF and fixed-interval ETKS2 significantly when using imperfect observations. This may have been due to a reduction in sampling errors. However, increasing the ensemble size above the model dimension generally decreased the accuracy. Increasing the ensemble size may have caused a poorer selection of vectors from the basis for the space in which the state estimate lies being used to form the new analysis state. For the ETKF with imperfect observations increasing the ensemble size greatly increased the fraction of analyses with ensemble mean within one ensemble standard deviation of the truth. The expected fraction given by Kreyszig [1999] is 0.68 for unbiased, normally distributed analysis errors with the standard deviation equal to the ensemble standard deviation. Using an ensemble size lower than 6 produced a value smaller than this in each coordinate. Thus an ensemble size lower than about 6 could mean the standard deviation is too small for the ETKF. However, the very large fractions of 1.00 seen in the fast coordinates for large ensemble sizes may be due to ignorance of the imperfect ETKF in the fast coordinates as the fast coordinates are not observed (Livings, 2005)

Decreasing the lag for ETKS2 caused the average errors to increase. This is expected since the forecast is carrying forward information from less observations in the analysis step near the beginning of the time window.

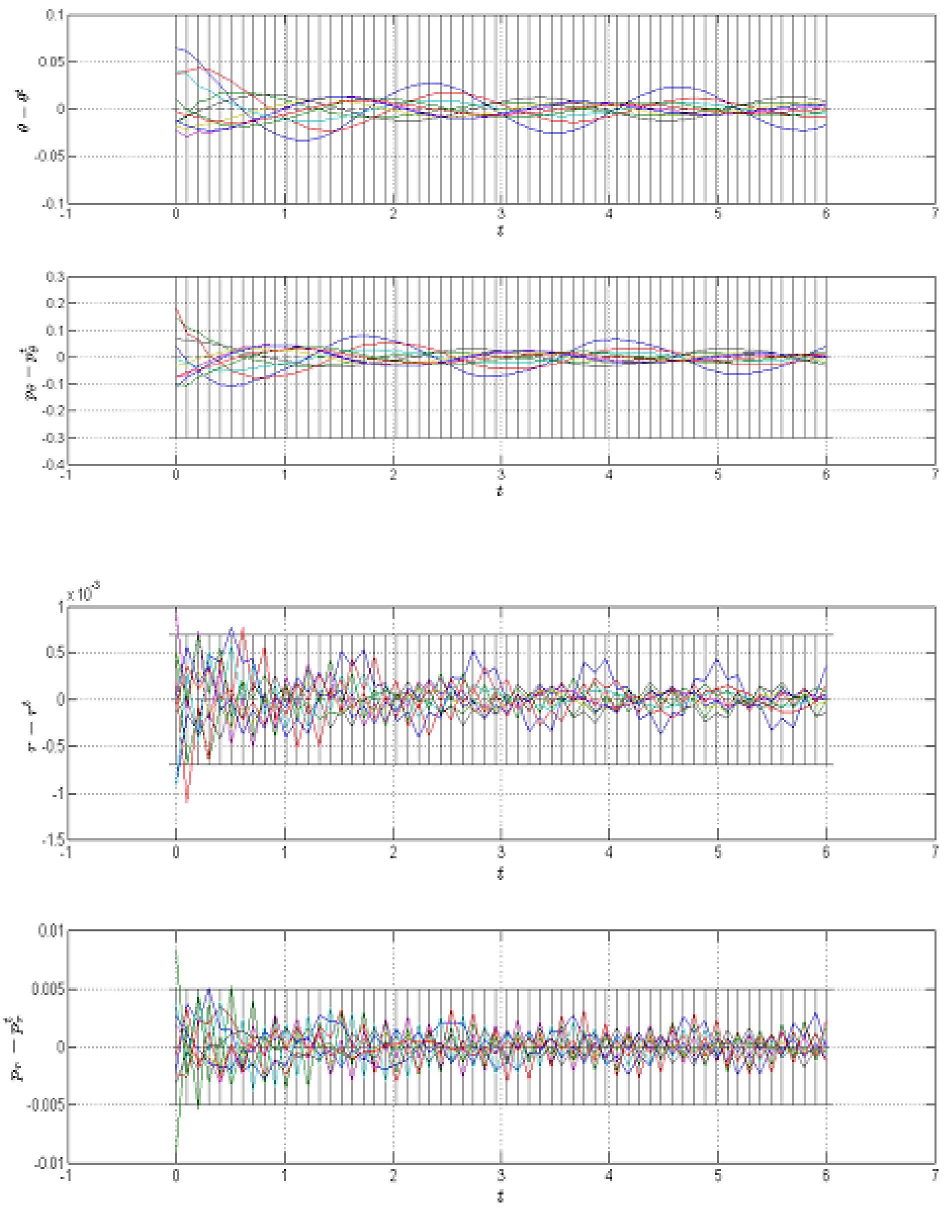

Figure 5.1: ETKF, perfect observations, 10 ensemble members. Coordinates are plotted relative to the truth. Lines showing individual ensemble members are superimposed on observations plotted as error bars. Radius of error bars equals standard deviation of filter. The error bars form a vertical grid centred on zero since the observations are frequent and perfect.

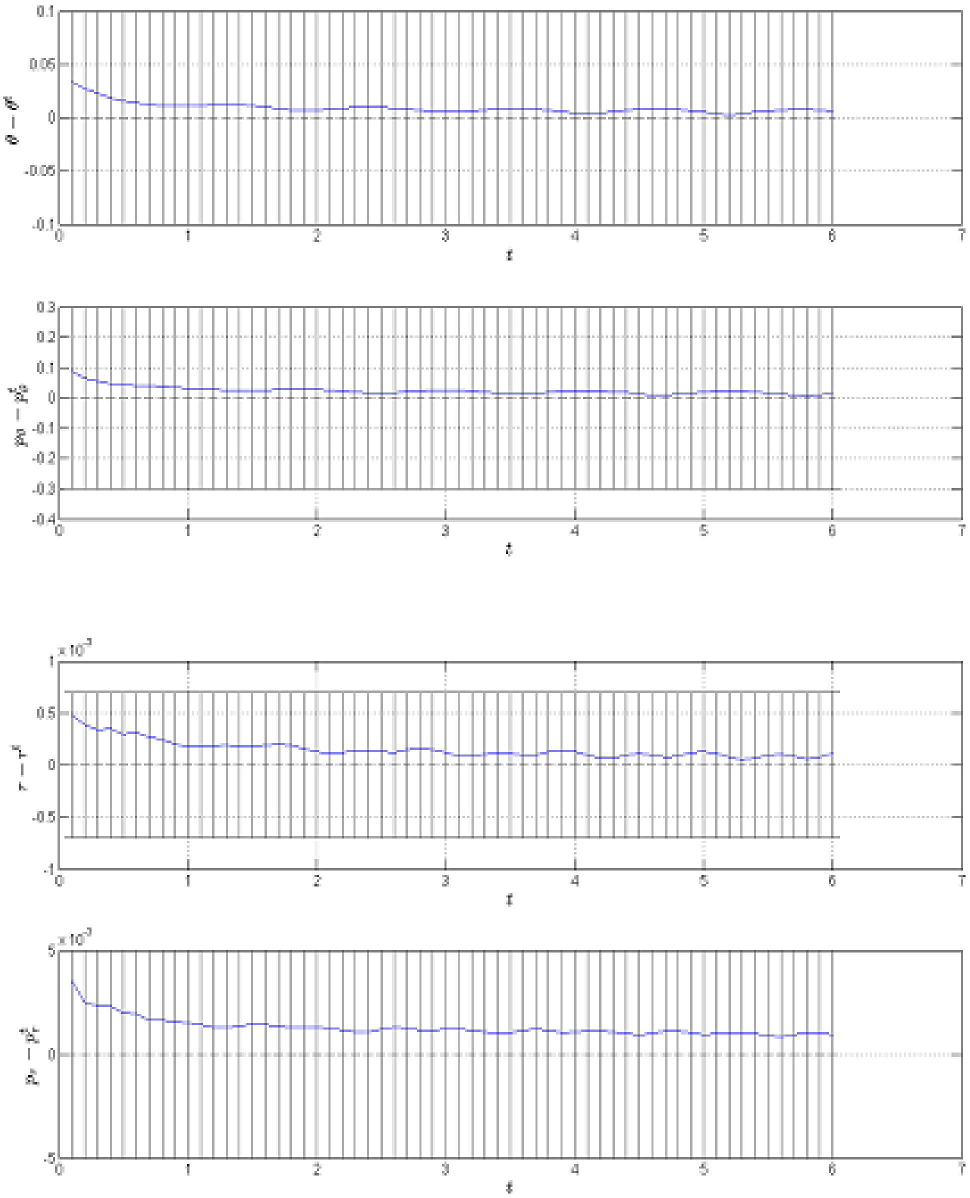

Figure 5.2: ETKF, perfect observations, absolute error trajectory of the ensemble mean ($N = 10$), averaged over 100 filter runs. Coordinates are plotted relative to the truth. Lines showing individual ensemble members are superimposed on observations plotted as error bars. Radius of error bars equals standard deviation of filter. The error bars form a vertical grid centred on zero since the observations are frequent and perfect.

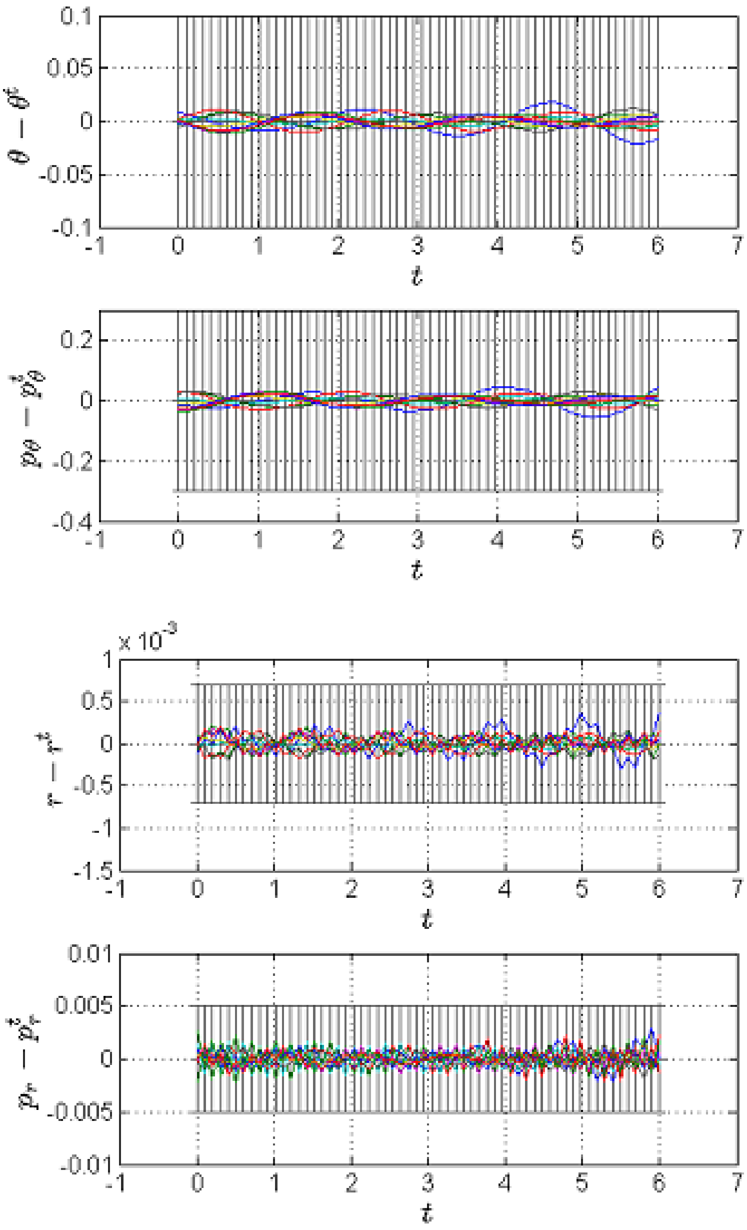

Figure 5.3: ETKS2, perfect observations, 10 ensemble members. Coordinates are plotted relative to the truth. Lines showing individual ensemble members are superimposed on observations plotted as error bars. Radius of error bars equals standard deviation of filter. The error bars form a vertical grid centred on zero since the observations are frequent and perfect.

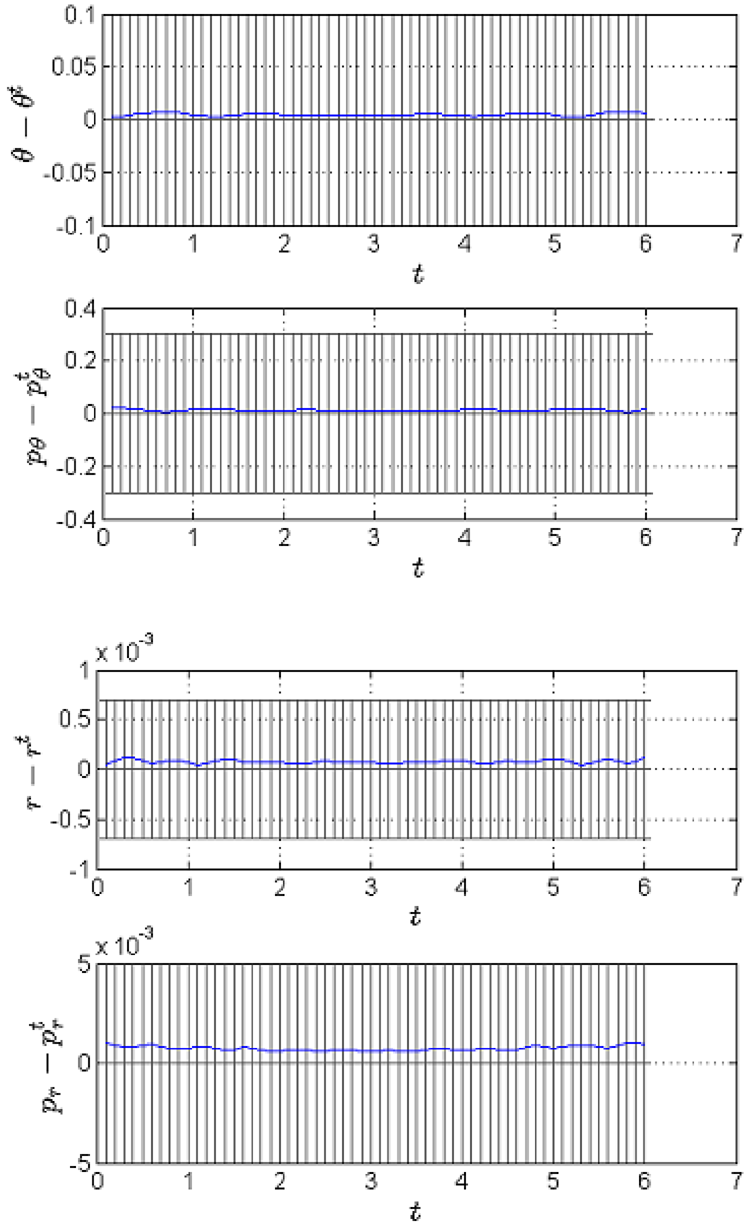

Figure 5.4: ETKS2, perfect observations, absolute error trajectory of the ensemble mean ($N = 10$), averaged over 100 smoothed runs.

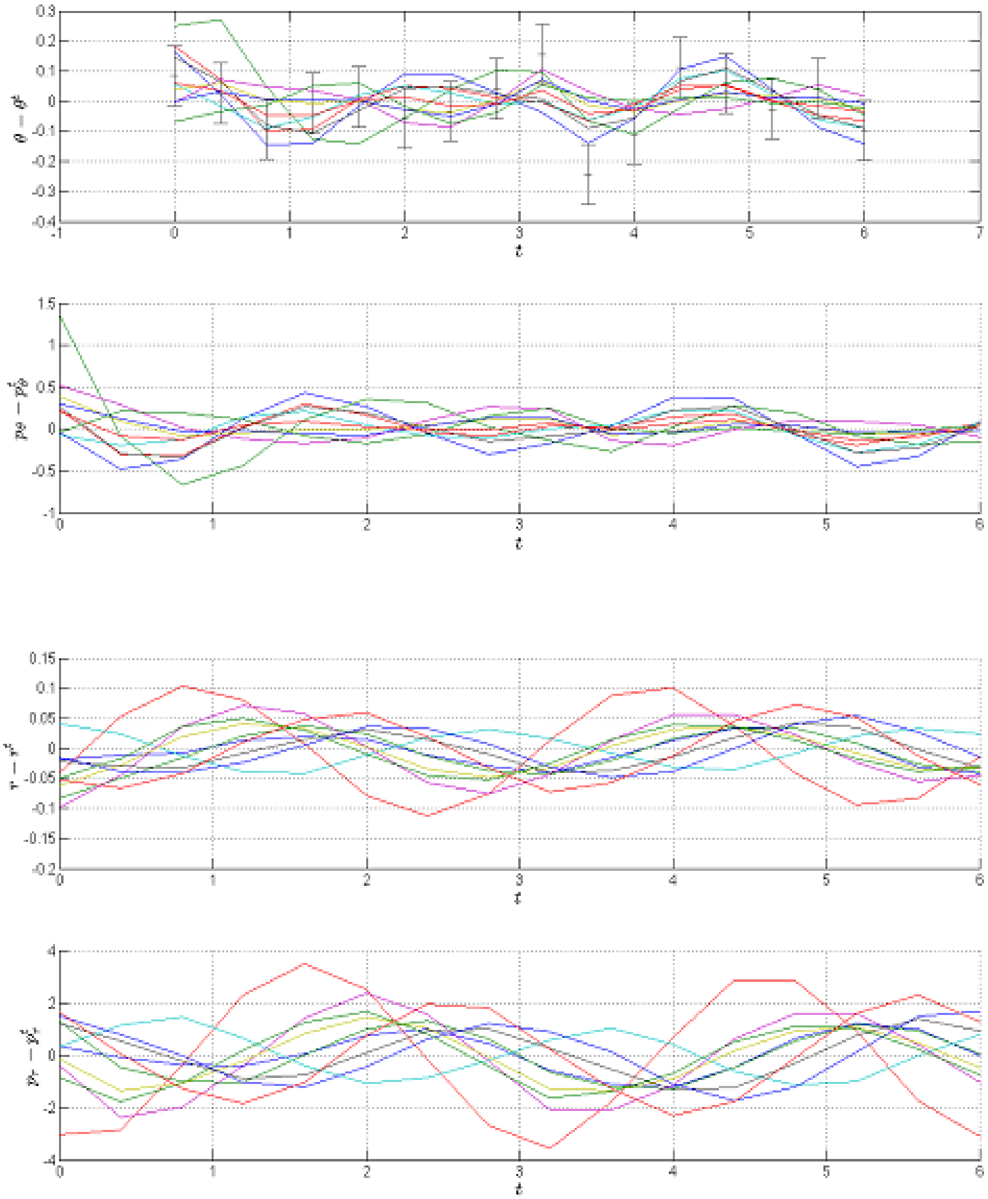

Figure 5.5: ETKF, imperfect observations, 10 ensemble members. Coordinates are plotted relative to the truth. Plotting conventions as for 5.1. Note that only the θ component has error bars since this is the only variable observed.

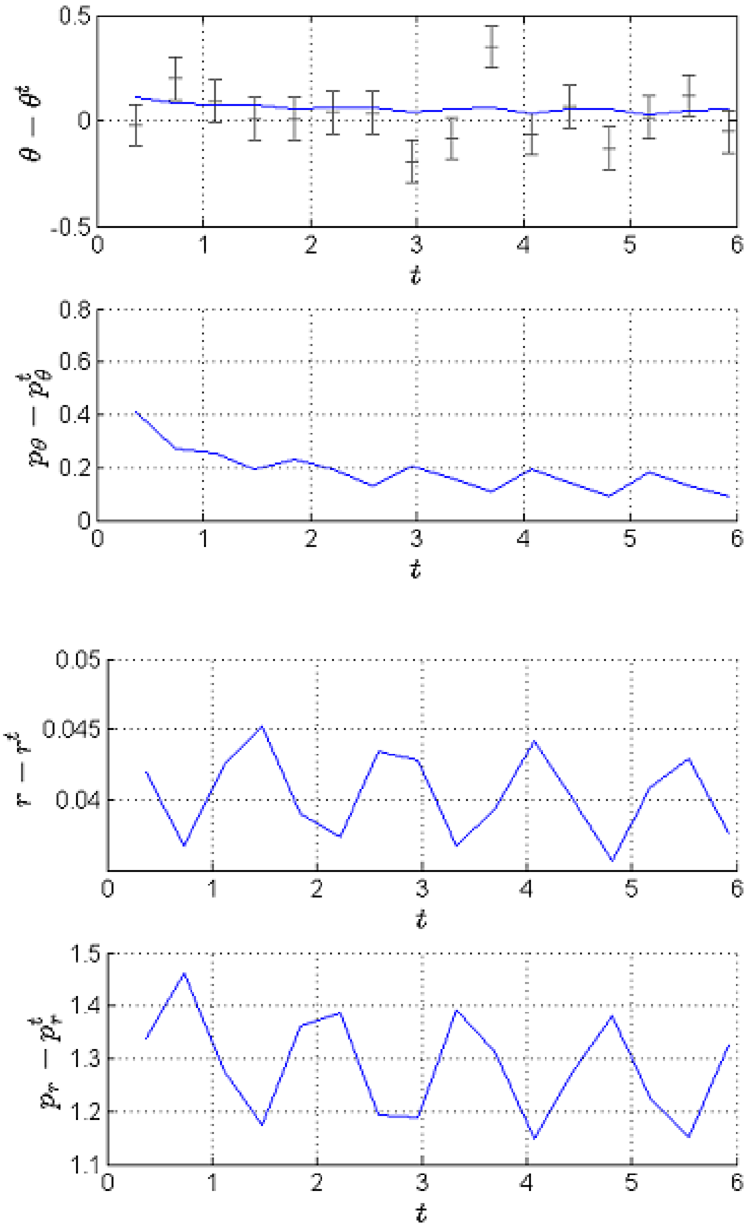

Figure 5.6: ETKF, imperfect observations, absolute error trajectory of the ensemble mean ($N = 10$), averaged over 100 smoothed runs. Plotting conventions as in Figure 5.2. Only the first graph shows error bars since only the first coordinate is observed.

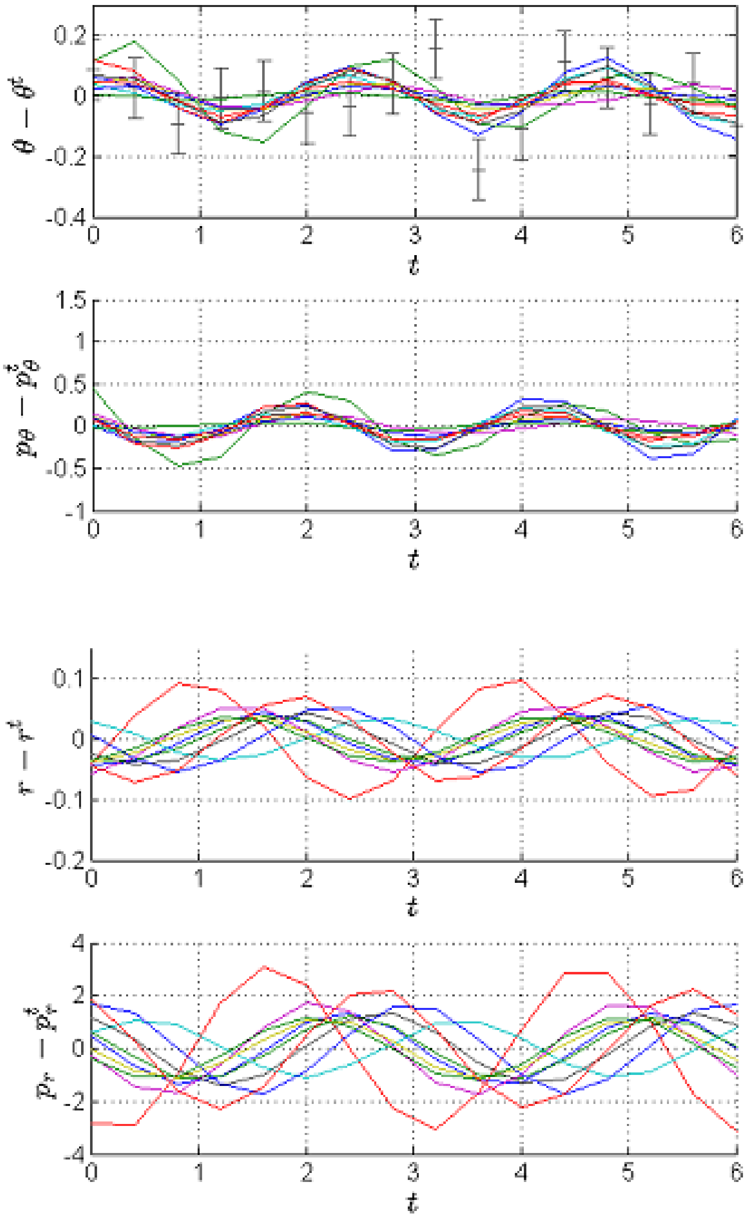

Figure 5.7: ETKS2, imperfect observations, 10 ensemble members. Coordinates are plotted relative to the truth. Plotting conventions as for 5.1. Note that only the θ component has error bars since this is the only variable observed.

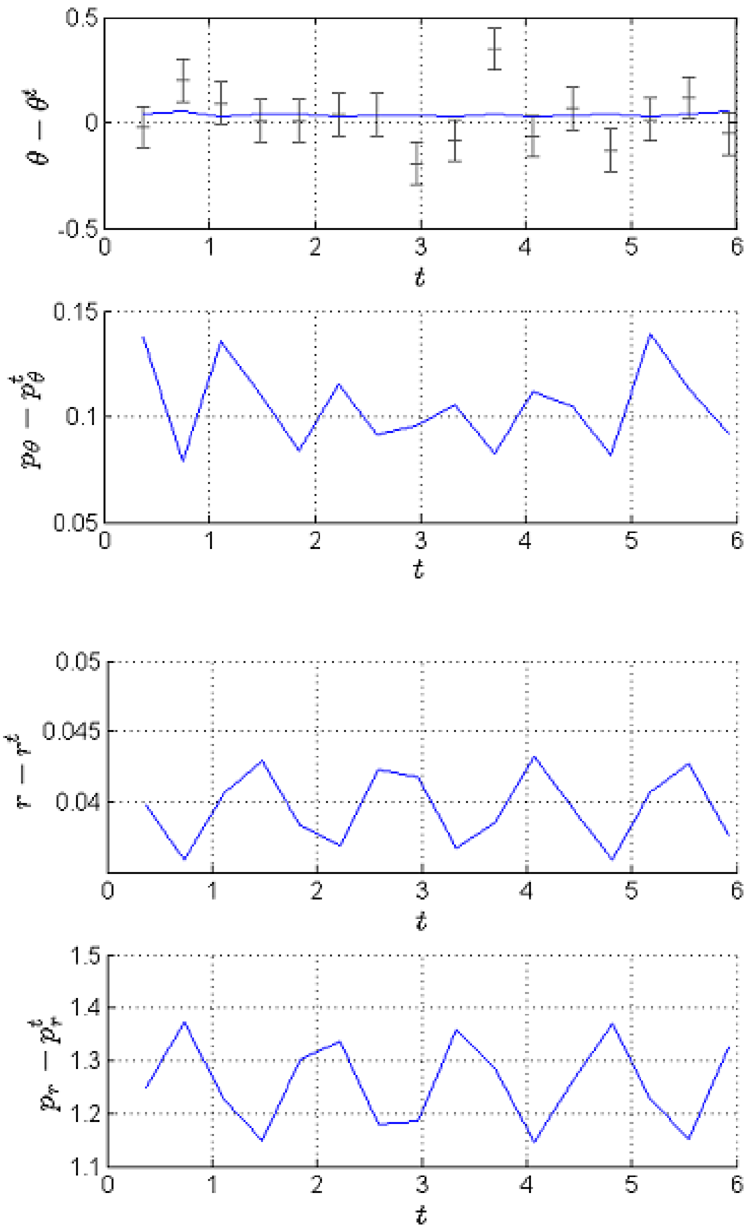

Figure 5.8: ETKS2, imperfect observations, absolute error trajectory of the ensemble mean ($N = 10$), averaged over 100 smoothed runs. Plotting conventions as in Figure 5.2. Only the first graph shows error bars since only the first coordinate is observed.

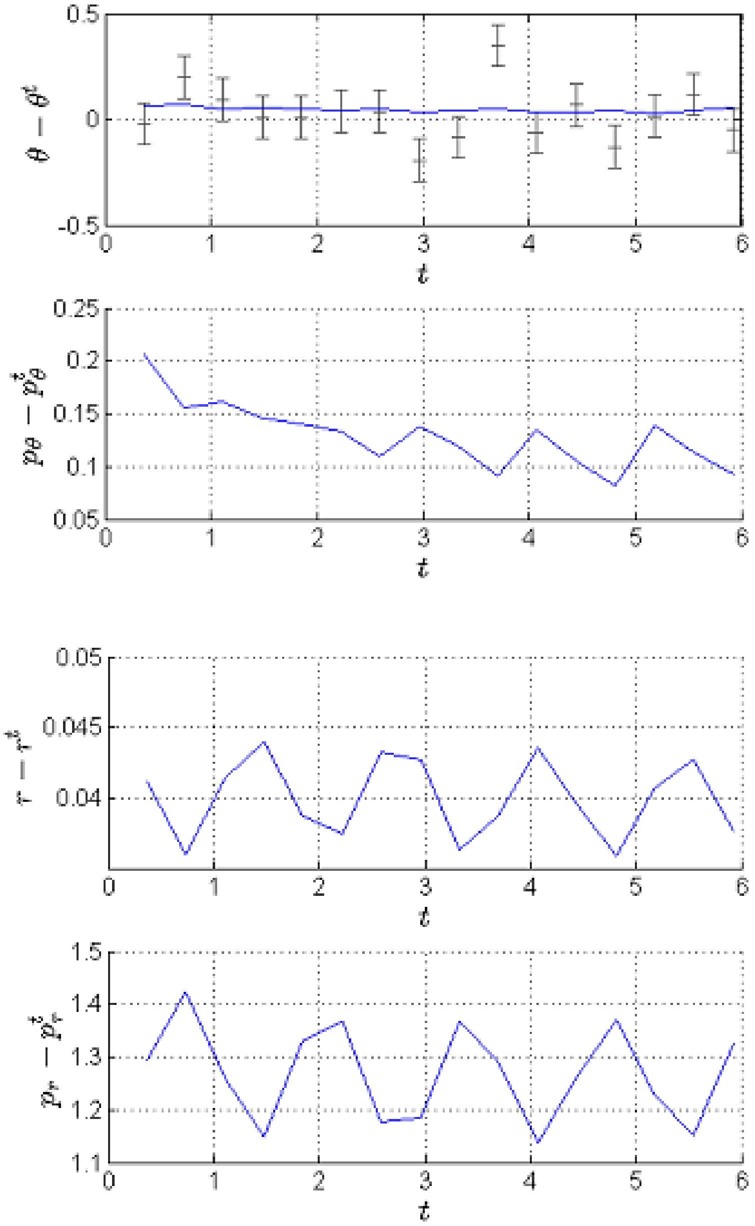

Figure 5.9: ETKS2-lag 4, imperfect observations, absolute error trajectory of the ensemble mean ($N = 10$), averaged over 100 smoothed runs. Plotting conventions as in Figure 5.2. Only the first graph shows error bars since only the first coordinate is observed.

Chapter 6

Conclusions and Future work

6.1 Summary and discussion

The goals for this project were to compare two different sequential data assimilation methods, the Ensemble Transform Kalman Filter (ETKF) of Bishop et al. [2001] and the Ensemble Transform Kalman Smoother type 2 (ETKS2) introduced in this report (see section 2.9.2). The aim was to compare the errors of the ensemble members for the ETKF and ETKS2 for perfect (noise-free) and imperfect observations. In the imperfect case, different ensemble sizes and different lags for the smoother are looked at.

Chapter 2 introduces the Kalman Filter (KF) of Kalman [1960] and the formulations of the KF including the ETKF that is used for experiments in this report. In a filter a forecast model is used to produce a forecast state estimate using the analysis (best estimate of the system) of the previous time. This forecast state is updated to produce the current analysis by giving a weighting to the observations according to the ratio of estimated observation covariances and forecast covariances. Two types of ensemble filters are introduced, stochastic and deterministic, which differ in the analysis step. The main advantage of the deterministic filter is that perturbed observations are not used, which introduce additional noise. The Ensemble Kalman Filter (EnKF) of Evensen [1994] (see section 2.3) is stochastic and the Ensemble Square Root Filter (EnSRF) of Tippett et al. [1999] (see section 2.4) is deterministic. The ETKF (see section 2.5) is a form of square root filter. Chapter 2 also introduces smoothing methods that differ from filtering methods in that future observations are assimilated in the analysis step (Ravela and McLaughlin, 2007). Smoothers would therefore be expected to be more accurate but more expensive to implement (Ravela and McLaughlin, 2007). The Ensemble Transform Kalman Smoother (ETKS) is introduced, which is based on the linear smoothing method of Jazwinski [1970] and the square root smoother of Whitaker and Compo [2002]. The smoothing method used in the experimental part of this report is called ETKS2 (see section 2.9.2) and differs slightly from the ETKS (section 2.9) in that a fixed forecast state is used in the innovation vector and square root matrix.

Chapter 3 discusses the two-dimensional swinging spring model. The concept of initialization is introduced and normal mode initialization is implemented for the swinging spring experiment of Lynch [2002]. Non-linear Normal Mode Initialization (NNMI) and Linear Normal Mode Initialization (LNMI) results are compared. Due to the non-linear nature of the system, NNMI is better than LNMI at removing the high frequency oscillations in the fast variables.

Chapter 4 discusses the implementation of the ETKF, the ETKS and the ETKS2. The formulations are equivalent to the formulations in Chapter 2 but the implementation describes an efficient way to compute the analysis. The implementation of the ETKF is the same as that of Livings [2005]. Algorithms are given for the implementation of the ETKS and ETKS2. The only difference between the ETKS and ETKS2 algorithms is that the forecast state is updated using the smoothed analysis state in the ETKS but the original forecast state from the filter is used in the ETKS2. This results in a smoothed innovation vector and square root matrix for the ETKS but a fixed innovation vector and square root matrix for the ETKS2. Thus the ETKS would be expected to be more expensive but more accurate.

In Chapter 5 firstly the ETKF and fixed-interval ETKS2 were compared using perfect observations for the Swinging Spring model. These observations are noise-free, frequent and measured on all four variables of the model. In order to validate the ETKS2 a seeded distribution was used and the final times and absolute errors for the ETKF and ETKS2 compared. Results for both methods had the same final ensemble values, since the same number of obervations are assimilated for both at the end of the time and the forecast states used for the innovation vector and square root matrix were identical. Also the absolute error for the ETKS2 was smaller over time. This was also expected since the ETKS2 uses all the observations to form the analysis at each step, instead of just past and present in the ETKF. A statistical significance method was given using the method for a normal distribution of Mood and Greybill [1963]. This could be used to determine how many runs would be needed as an average for the general behaviour of the ETKF and ETKS2. This method was in the form of a confidence interval, giving the 95% chance that the average of a certain number of runs would have a value in a certain interval.

In order to check whether the ETKS2 was improving the accuracy of the ensemble members compared with the ETKF the errors in the ensemble mean were averaged over

100 runs and averaged over the time interval. The statistical significance test measured the confidence in the analysis errors for the 100 run average. The mean ensemble absolute errors averaged over the 100 runs were less for the ETKS2 than the ETKF, proving that the smoother has improved accuracy over the filter. Also since the 100 run average standard deviations were less for the smoother, there was less variation in the ensemble spread. However, the ETKF is known to underestimate ensemble spread so a smaller standard deviation is not necessarally a good thing for either the ETKF or ETKS2. Ensemble mean absolute errors averaged over 100 runs were plotted over the time interval for both the ETKF and ETKS2. The ETKF errors were worse at the beginning of the time window. This was expected since the filter error decreases as more observations are assimilated. The ETKS2 errors were consistent over the time interval, as expected since the fixed-interval smoother assimilates all available observations at each analysis step.

Imperfect observations were tested for both the ETKF and ETKS2 in the same way as the perfect observations. Only the slow coordinate θ was observed. The analysis errors proved to be much greater than for the perfect case, especially in the unobserved fast coordinates. The average ETKS2 errors were also greater than the ETKF ones.

The ETKF and ETKS2 analysis errors were compared for different ensemble sizes using imperfect observations. The Errors improved significantly for both cases when the ensemble size increased up to the model dimension ($n = 4$). This may have been due to less sampling errors for the larger ensemble size. After about $n = 6$ the errors generally increased for higher ensemble sizes. Increasing the ensemble beyond the model dimension may have been causing a poor selection of vectors from the basis for the space in which the state estimate lies to form the new analysis state estimate (Nichols, 2009). The change in ensemble size did not change the fact that the smoother was always more accurate than the filter in all four coordinates. For the ETKF with imperfect observations increasing the ensemble size greatly increased the fraction of analyses with ensemble mean within one ensemble standard deviation of the truth. This suggests that a greater number of ensemble members means the standard deviation is less likely to be underestimated. However, the very large fractions of 1.00 seen in the fast coordinates maybe due to ignorance of the imperfect ETKF in the fast coordinates as the fast coordinates are not observed (Livings, 2005).

The ETKS2 with imperfect observations was tested for the fixed-lag case using a range of different lags, varying from 1 to 16, where 16 is the fixed-interval smoother. The mean ensemble analysis errors increased near the beginning of the time-window for lower lags since it is here that the smallest number of observations were being assimilated. However, the time averaged ensemble mean errors for all the fixed-lag ETKS2 were less than the ETKF errors. This makes sense since more observations are being assimilated for the fixed-lag ETKS2 than the ETKF in most of the analysis states.

6.2 Limitations of the experiments

6.2.1 The Swinging Spring system

There are some limitations in the experiments in this report. Firstly, the swinging spring system has only four parameters. In Numerical Weather Prediction the number of parameters p of the model is of $O(10^7)$. The number of ensemble members $N << p$. In order to relate the experiments for the smoother to NWP a much higher dimensional system would be needed. For example, the square root smoother of Whitaker and Compo [2002] gives results using a 40-dimensional model. Using this model it is discovered that sampling errors in the cross-covariance matrix are produced by higher lags. Thus higher ensemble sizes are needed to take advantage of observations further removed from the analysis. The four-dimensional swinging spring does not have a large enough range of ensemble sizes for $N < p$ to find a relationship between lag, sampling errors and ensemble size.

6.2.2 Experimental work

In the experimental work in this report, only two different cases were considered for the ETKF anf ETKS2. The perfect case was first looked at, using noise-free, high frequency and observations in all four parameters. Then the imperfect case looked at noisy observations at a lower frequency and with only the θ component observed. It would have been better to look at the imperfect case with two and three parameters observed first to see the impact on changing things one by one. This could have given a clearer relationship between observations in the fast and slow coordinates. It may also have been useful to experiment with different ensemble sizes and lags for the perfect case as well as the imperfect case. It would also have been a good idea to look at the fraction of ensembles with ensemble mean within one ensemble standard deviation of the truth for different ensemble sizes for the ETKS2. This could have then be compared with the ETKF.

6.3 Future work

6.3.1 Experiments with the ETKS

Although the ETKS was derived and a form of implementation suggested, it was not experimented with. It would be useful to experiment with the ETKS and compare the results with the ETKS2 and ETKF. The ETKS results would be expected to be more accurate than the ETKS2 results since the square root matrix T uses smoothed forecast states for each time step instead of using only the original filtered forecast states. However the extra smoothing of the forecast states would make it more expensive to implement. The ETKS closely resembles the EnSRS of Whitaker and Compo [2002] in that it uses the smoothed forecast states to calculate the square-root matrix. Thus it would be expected to exhibit similar behaviour to the EnSRS when tested with a similar model.

6.3.2 Initialization

The key experimental work in this report involved the comparison of the ETKF and ETKS2 with an initialized system. Although initialization is not the main subject of this report, it could be investigated in the furture. Firstly it is important to understand the meaning of initialization. The spectrum of atmospheric motions is vast, having time-scales ranging from seconds to millenia (Lynch, 2002). For example, localised pressure fluctuations can have time-scales less than a minute. Contrastingly, climatic changes such as ice ages can have cycles of tens of thousands of years. According to Kasahara [1976] the natural oscillations of the atmosphere fall into two groups. On longer length scales (>> 1km) and longer timescales (> 1 day) occur low frequency oscillations. To a good appoximation these low frequency oscillations are in Geostrophic balance. Geostrophic balance is a balance beteen the pressure gradient force and the Coriolis force. The equations are given as

$$u_g = -\frac{g}{f}\frac{\partial z}{\partial y} \tag{6.1}$$

and

$$v_g = -\frac{g}{f}\frac{\partial z}{\partial x} \tag{6.2}$$

where g is the acceleration due to gravity, f is the Coriolis parameter, z is the geopotential height of the constant pressure surface. x and y are the zonal and meridional coordinates with velocities u_g and v_g respectively. Geostrophic flow is divergence-free, meaning that the net volume of air leaving a volume is the same as the net volume of air entering the volume.

The second group of atmospheric oscillations are high frequency waves, which are far from geostrophic balance. These waves typically have phase speeds of hundreds of metres per second and have large divergence. For typical atmospheric conditions of large scale atmospheric flow the two types of motion are clearly separated and the interaction is weak (Kasahara, 1976). High frequency oscillations serve as a good approximation on longer spacial and time scales. However, in certain situations it may be necessary to predict high frequency variations, such as localised mountain updraughts where the rising condensing air could cause cloud hazardous to aircraft.

The low frequency variations that are of no interest are called noise. If noisy observations are incorporated into the basic equations in numerical prediction the forecast may contain spurious large amplitude high frequency oscillations (Lynch, 2002). This is because the balance between the mass and velocity fields is not fairly reflected in the forecast. In other words, the numerical model cannot cope with the high frequency variables. The elimination of this noise can be achieved by adjusting the initial fields, a process called initialization. Detail on different forms of initialization that can be found is in Lynch [2002]. Initialization is an important part of data assimilation. Any linear balances in a system should be preserved with the ETKF since the ETKF takes a linear combination of the ensemble members to create the analysis. However, a smoother would be expected to give a smoother solution that fits the observations better. Thus a smoother might be expected to dampen the high frequency and high amplitude errors in the solution more than the filter. The relationship between initialization, the ETKS/ETKS2 and ETKF could be investigated using the Swinging Spring model or another model by measuring the spread of the errors in the ensemble members in the solutions. This could be done by plotting the ensemble mean and the ensemble mean ± standard deviation for the ETKS/ETKS2 and comparing the results with the ETKF plots from Livings [2005]. Neef et al. [2005] investigate initialization properties of a stochastic EnKF using a four-dimensional system. Perhaps the deterministic ETKF, ETKS and ETKS2 could be applied using similar experiments.

Bibliography

B. D. Anderson and J. B. Moore. *Optimal filtering.* Prentice-Hall, 1979.

R. N. Bannister. Elementary 4D-Var. *DARC technical report. No 2. University of Reading.*, 2007.

P. Bergthorsson and B. Doos. Numerical weather map analysis. *Tellus*, 7:329–340, 1955.

C. H. Bishop, B. J. Etherton, and S. J. Majumdar. Adaptive sampling with the Ensemble Transform Kalman Filter. *Monday Weather Review*, 129(Part 1:Theoretical aspects): 420–436, 2001.

G. Burgers, P. J. van Leeuwen, and G. Evensen. Analysis scheme in the Ensemble Kalman Filter. *Monday Weather Review*, 126:1719–1724, 1997.

S. E. Cohn, N. S. Sivakumaran, and R. Todling. A Fixed-Lag Kalman Smoother for Retrospective Data Assimilation. *Monday Weather Review*, 122:2838–2867, 1994.

F. X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations:theoretical aspects. *Tellus*, 38A:97–110, 1986.

M. Ehrendorfer. Four Dimensional data assimilation: comparison of variational and sequential algorithms. *Quart. J. Roy. Meteor. Soc*, 118:673–713, 1992.

M. Ehrendorfer. A review of issues in ensemble-based Kalman filtering. *Meteorol. Z.*, 16: 795–818, 2007.

G. Evensen. Sequential data assimilation with a nonlinear quasigeostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophy. Res*, 99(C5):10143–10162, 1994.

G. Evensen. The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

G. Evensen. Sampling strategies and square root analysis schemes for the ENKF. *Ocean Dynamics*, 54:539–560, 2004.

G. Evensen and P. J. van Leeuwen. An Ensemble Kalman Smoother for Linear Dynamics. *Monthly Weather Review*, 128:1852–1867, 2000.

G. H. Golub and C. F. Van-Loan. *Matrix computations.* The Johns Hopkins University Press, third edition, 1996.

T. M. Hamill. *Predictability of weather and climate*, pages 124–156. Cambridge University Press, 2006.

A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic press, 1970.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):3545, 1960.

E. Kalnay. *Atmospheric modelling, Data Assimilation and Predictability*, pages 136–210. Cambridge University Press, 2003.

A. Kasahara. Normal modes of ultra long waves in the atmosphere. *Monthly Weather Review*, 104:669–690, 1976.

E. Kreyszig. *Advanced engineering mathematics.* John Wiley and Sons, 1999.

J. D. Lambert. *Numerical methods for ordinary differential systems.* John Wiley and Sons, 1991.

D. M. Livings. Aspects of the Ensemble Kalman Filter. Master's thesis, Maths department. Reading University, October 2005.

D. M. Livings, S. L. Dance, and N. K. Nichols. Unbiased ensemble square root filters. *Physica D*, 237/8:1021–1028, 2008.

E. N. Lorenz and K. A. Emanuel. Optimal sites for supplementary weather observations:simulation with a small model. *J. Atmos. Sci*, 55:399–414, 1998.

P. Lynch. *Large-Scale Atmosphere-Ocean Dynamics II: Geometric Methods and Models*, pages 64–108. Cambridge University Press, 2002.

R. Menard and R. Daley. The application of the Kalman Smoother theory to the estimation of 4D-VAR error statistics. *Tellus*, 48A:221–237, 1996.

M. Mood and F. Greybill. *Introduction to the theory of statistics*, pages 262–267. McGraw-Hill Book Company, 1963.

L. Neef, S. Polavarapu, and T.Shepard. Four-Dimensional data assimilation and balanced dynamics. *Journal of Atmospheric Science*, 63:18401858, 2005.

N. Nichols. personal communication, 2009.

R. Petrie. Localization in the Ensemble Kalman Filter. Master's thesis, Department of Meteorology. Reading University, August 2008.

H. E. Rauch. Advanced orbit mechanics studies: Differential correction and least squares estimation. *Lockhead Missiles and Space Company*, 1967. C-64-67-1.

S. Ravela and D. McLaughlin. Fast Ensemble Smoothing. *Ocean Dynamics*, 57:123–134, 2007.

M. K. Tippett, J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker. Ensemble square root filters. *Monday Weather Review*, 131:1485–1490, 1999.

UKMO. Observations. From http://www.metoffice.gov.uk/research/nwp/observations/, Last accessed July 2009.

J. S. Whitaker and G. P. Compo. An Ensemble Kalman Smoother for reanalysis. *American Meteorological Society*, pages 144–147, 2002.

J. S. Whitaker and T. M. Hamill. Ensemble data assimilation without perturbed observations. *Monday Weather Review*, 130:1913–1924, 2002.

Y. Zhu, R. Todling, J. Guo, S. E. Cohn, M. Navon, and Y. Yang. The GEOS-3 Retrospective Data Assimilation System: The 6-hour Lag Case. *Monthly weather review*, 131:2129–2150, 1999.