

The University of Reading

Flood modelling and data assimilation

N. Raoult, S.L. Dance

MATHEMATICS REPORT 1/2011

*Department of Mathematics and Statistics
The University of Reading
Whiteknights, PO Box 220
Reading
Berkshire RG6 6AX*

Department of Mathematics and Statistics

Flood Modelling and data assimilation

Nina Raoult¹ and Sarah L. Dance²

¹Exeter College, The University of Oxford, Oxford, OX1 3DP

²School of Mathematical and Physical Sciences, The University of Reading, Reading, RG6 6BB

Corresponding author's email: s.l.dance@reading.ac.uk

Abstract

Flooding is considered to be one of the most frequent and costly types of natural disaster. Over recent years, flood events in the UK have demonstrated some problems with the flood forecasting systems being used, in particular with regard to modelling extreme fluvial floods, rapidly responding river catchments and urban surface water flooding. Even with a perfect model, for accurate forecasts we require accurate initial data and rainfall inputs. The aim of this report is to investigate the use of a data assimilation technique to compensate for initialization and rainfall input errors. By applying an Ensemble Transform Kalman Filter (ETKF) to a simple 1-D flood model based on the Horton-Izzard equation, we first investigated the effect of varying the assimilation parameters: ensemble size, observation and background error covariances and observation frequency. Numerical 'identical twin' experiments were carried out, where a particular forecast of the model is treated as a trajectory of the 'true' system and synthetic observations generated from this trajectory. It was found that increasing the number of ensemble members up to 10 members gave the best results: no further improvements were seen for ensemble sizes larger than 10. The effects of changing the initial background error covariance and the observation error covariance were seen to be linked, since the analysis is dependent on their ratio rather than the absolute value of these individual covariances. The best choice of initial background error was equal to or slightly larger than the observation error covariance. A higher frequency of observations often gave better results although this was not conclusive. The use of noisy observations (rather than perfect observations of the truth) gave better results since this helped to reduce ensemble collapse. Through these experiments the basic behaviour of the filter was understood and allowed us to address the question of how well the filter could compensate for errors in the rainfall inputs. It was found that the filter compensated well for rainfall input magnitude errors. For errors in rainfall timing the filter was able to compensate for errors where the rainfall started earlier or later than predicted, but was less able to improve the solutions where the rainfall event finished before the predicted rainfall. We also considered the effect of running the filter with incorrect parameter values and saw that errors in model parameters may be used to compensate other errors of the assimilation system by increasing the ensemble spread. A collapse in ensemble spread was the main problem observed using the filter. In future experiments this might be compensated for by introducing a covariance inflation factor. All the results given here are based on a 1-D model and observations and in reality models and observations are of higher dimensions. It is not clear how generally applicable the conclusions drawn in this work are.

1 Introduction

Flooding is considered to be one of the most frequent and costly types of natural disaster (e.g., Vos et al., 2010). Floods can cause disease due to loss of basic sanitation, food shortages, injuries, destruction of infrastructure etc. Flooding is a real problem in the UK. Over 5 million people in the UK live and work in properties that are at risk of flooding from rivers or the sea (The Environment Agency, 2010).

Over recent years, flooding events in the UK have shown us there are some problems with the flood models being used, in particular with regard to modelling extreme fluvial floods, rapidly responding river catchments and urban surface water flooding (The Pitt Review, 2008). Even with perfect models, for accurate forecasts, we require accurate initial data and rainfall inputs. These are major sources of uncertainty in flood forecasting (Leahy et al., 2007). The aim of this project is to investigate the use of a data assimilation technique to compensate for initialization and rainfall input errors. Data assimilation is a method of state estimation that combines model and observational data, taking account of the uncertainty in the data, to provide improved estimates of the current state of a modelled system. This report describes a simplified ODE flood model that has been combined with an associated ensemble data assimilation system, and the results of investigations into the performance of this system.

The rest of this paper is organised as follows. Section 2 briefly introduces the simplified 1-D flow model used and discusses its implementation. The implementation is verified for a few simple rainfall scenarios in section 3. Additional rainfall scenarios are discussed in section 4. Section 5 explores the model parameter space.

Section 6 introduces the concept of data assimilation and describes the Kalman Filter which is valid for linear models and Gaussian statistics. The Ensemble Kalman Filter (EnKF), which uses statistical sample of forecasts to calculate a state estimate and its uncertainty, allows for the use of more general models, and is described in section 7. Section 8 describes the results of experiments investigating the effects of varying the assimilation parameters: ensemble size, observation and background error covariances and observation frequency. Section 9 investigates the filter's ability to compensate for errors in the rainfall inputs and section 10 considers the effects of using incorrect model parameters with the filter. We conclude in section 11.

2 The Model

Fluvial flooding has non-linear dynamics. It is difficult to model because there are so many factors affecting it: river flow, river size, groundwater saturation etc. We will limit ourselves to a simple 1-D model.

The Horton-Izzard equation (Dooge, 1973, as described by Moore and Weiss (1980)) gives a simple non-linear flood model, as

$$\frac{dx}{dt} = a(r - x)x^b. \quad (1)$$

The equation describes the flow $x(t)$ in a river channel at time t in response to the rainfall input $r(t)$. a and b are parameters which will be discussed further in section 5. The flow x and rainfall r are assumed to be non-negative so that our solutions have physical meaning. In reality, there should be a time delay between the rainfall and the change in flow to allow the water to penetrate the ground, but to simplify the model we have chosen to ignore this delay.

The differential equation (1) must be solved subject to the initial conditions

$$x(0) = x_0, \quad (2)$$

where x_0 is some prescribed initial flow rate. Unless otherwise stated, $x_0 = 1$ in the experiments described in this report.

2.1 Model Implementation in MATLAB

The model is implemented in MATLAB using the `ode45` function (MathWorks, 2010a) to solve the initial value problem (1), (2). The numerical solver is based on an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair (Dormand and Prince, 1980). At each step, the solver estimates the local error \mathbf{e} in the i th component of the solution. The solver works in such a way that this error must be less than or equal to the acceptable error. This acceptable error is a function of the specified relative tolerance, `RelTol`, and the specified absolute tolerance, `AbsTol`, such that

$$|\mathbf{e}(i)| \leq \max(\text{RelTol} * \text{abs}(\mathbf{x}(i)), \text{AbsTol}(i)). \quad (3)$$

`RelTol` is a measure of the error relative to the size of each solution component. In these experiments, we use the default value of 1×10^{-3} which corresponds to 0.1% accuracy. The value of `AbsTol` determines the accuracy when the solution approaches zero by giving a threshold below which the value of the i th solution component is negligible. For our experiments we use `AbsTol` = 1×10^{-6} . An additional parameter, `Maxstep` gives the upper bound on the solver step size. For the experiments described here, `Maxstep` is set to $\frac{2\pi}{10}$.

2.2 Rainfall scenarios

In (1), the rainfall is prescribed as an external input. We have developed several different scenarios for synthetic rainfall inputs to test the model and assimilation system.

2.2.1 Drought scenario

In the ‘drought’ scenario there is no rainfall input, i.e. $r \equiv 0$.

2.2.2 Constant rainfall

In this scenario $r \equiv r_0$, a constant value.

2.2.3 Regular rainfall

The ‘regular’ rainfall scenario corresponds to constant rainfall intermittent over equal intervals of the time period. It is defined as

$$r(t) = \begin{cases} 1 & 0 < t < \pi \\ 0 & \pi \leq t \leq 2\pi \end{cases}, \quad (4)$$

and by periodic extension $r(t) = r(t + 2\pi)$ outside this range.

2.2.4 Random rainfall

‘Random’ rainfall is simulated by a bounded random walk. Here the forecast period is divided into a number of short, equal time intervals T_i . For each time interval, a pseudo-random number, p_i is drawn from the standard uniform distribution on the interval $[0,1]$, using the MATLAB `rand` function (MathWorks, 2010b). Then, writing $r(t) = r_i$ for $t \in T_i$, we have

$$r_i = \begin{cases} r_{i-1} - 1 & 0 \leq p_i \leq 1/3, \\ r_{i-1} & 1/3 < p_i \leq 2/3 \\ r_{i-1} + 1 & 2/3 < p_i \leq 1 \end{cases}. \quad (5)$$

A maximum value for the rainfall is set to 10 units; once the rainfall has reached this value, it will remain constant if $p_i > 2/3$. Similarly, the minimum value of rainfall is equal to zero; once the rainfall has reached this value, it will remain constant if $p_i < 1/3$. The initial rainfall is set to be zero.

2.2.5 Semi-Random rainfall

‘Semi-random’ rainfall is a hybrid of the regular and random rainfall scenarios. Here, the rainfall is determined using the ‘random’ function for $t \in (2n\pi, (2n+1)\pi)$, $n = 0, 1, 2, \dots$ and $r = 0$ for other values of t .

3 Model Verification

One method for validating the model is to solve the equation analytically for certain parameters and then compare these solutions to the outputs given by the model. This approach only allows simple solutions to be checked so it will not extensively test the model. If all the solutions could be found analytically, there would be no need to find them numerically.

Test 1: Drought, $a = 1$, $b = 0$

Drought is the scenario where there is no measured rainfall, i.e. $r = 0$. Equation (1) reduces to

$$\frac{dx}{dt} = -x, \quad (6)$$

which with initial condition, $x(0) = 1$, has solution

$$x(t) = e^{-t}. \quad (7)$$

This exponentially decaying solution makes physical sense: no rain means no input into the model and hence the flow will decay as there is less and less water available.

Fig. 1 depicts the absolute difference between the analytical and model solutions. It shows an oscillating error with magnitude that decreases with time. This decrease with time is due to the fact that both the numerical and analytical solutions decay with time. Given the choice of `ode45` parameters explained in section 2.1, this error is within the range of the expected magnitude.

Test 2: Drought, $a = 1$, $b = 0.5$

This is similar to Test 1, but with b non-zero. By substituting these values into (1), and using the initial condition $x(0) = 1$, we find

$$\frac{dx}{dt} = -x^{3/2}; \quad x(t) = \frac{4}{(t+2)^2}. \quad (8)$$

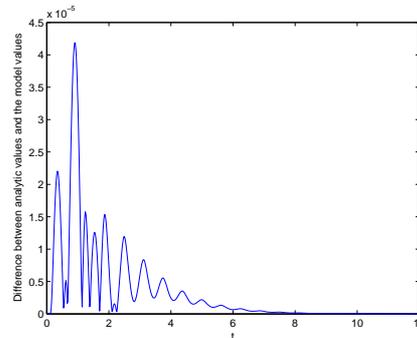


Figure 1: Absolute error of numerical solution for Test 1 against time.

The absolute difference between the exact and numerical solutions are shown in Fig. 2. As in Test 1, the graphs shows an oscillating error which decreases with time. The magnitude of error is again small, here $O(10^{-4})$, which is consistent with the error expected from the numerical scheme.

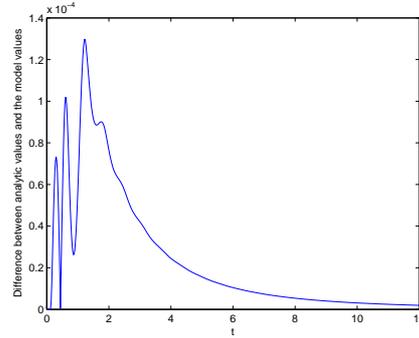


Figure 2: Absolute error of numerical solution for Test 2 against time.

Test 3: Regular, $a = 1$, $b = 0$ We consider the regular rainfall scenario, and seek the exact solution for $0 \leq t \leq 3\pi$. Solutions for subsequent times can be found similarly. In this case, (1), reduces to

$$\frac{dx}{dt} = \begin{cases} 1 - x, & \text{if } 0 < t < \pi, 2\pi < t < 3\pi \\ -x, & \text{otherwise} \end{cases} \quad (9)$$

We solve the ODE over three separate intervals: $0 < t < \pi$, $\pi \leq t \leq 2\pi$ and $2\pi < t < 3\pi$.

Interval 1: $0 < t < \pi$

$$\frac{dx}{dt} = 1 - x.$$

The general solution for this ODE is $x = Ae^{-t} + 1$, where A is a constant. To find the value of the constant, the initial condition $x(0) = 1$ is substituted into the solution giving $A=0$ so

$$x(t) \equiv 1.$$

Interval 2: $\pi \leq t \leq 2\pi$

$$\frac{dx}{dt} = -x.$$

This is as in the drought scenario but this time the initial condition is $x(\pi) = 1$. The general solution for this ODE is $x = Ae^{-t}$, where A is a constant. Substituting the initial conditions gives:

$$x(t) = e^{\pi-t}.$$

Interval 3: $2\pi < t < 3\pi$

This interval has the same equation as in the first interval, so has the same general solution. This time, however, the initial condition is $x(2\pi) = e^{-2\pi}$. This is found by matching to the solution at interval two, since 2π is the boundary between these two intervals and the solution we are looking for is continuous. The particular solution obtained is:

$$x(t) = (1 - e^{2\pi})e^{-t} + 1.$$

These equations describe the analytic solution. They can now be compared to the model solution to see if the numerical solutions agree. The results are plotted in Fig. 3.

Looking at the graphs in Fig. 3, the numerical and analytic answers match up visually. A more quantitative analysis shows that the errors in the numerical solution are of the expected size. Over the range $0 < t < \pi$, $r = 1$ and $x = 1$, the flow is at the same speed as the input, there is no change in the water volume and so the flow remains the same. Over the range $\pi < t < 2\pi$ the graph is similar to the drought scenario; no rainfall causes the flow to decay exponentially. Over the range $2\pi < t < 3\pi$ the flow increases towards $x=1$. When the flow reaches 1, the flow speed is the same as the water input, as in the case in the first interval.

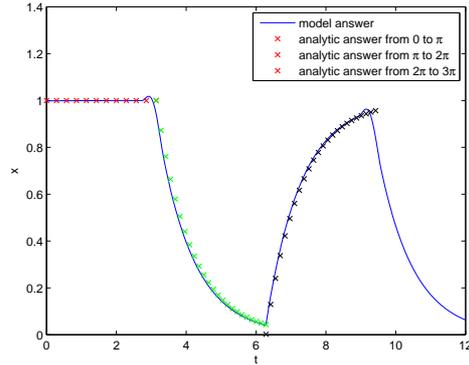


Figure 3: Numerical and analytical solutions for regular rainfall, Test 3, against time.

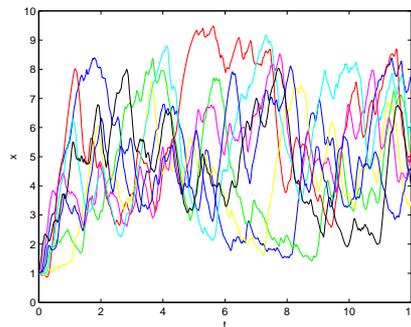


Figure 4: Random rainfall scenarios with, $a = 1.0$ and $b = 0.5$ for multiple realisations against time. Each realisation is plotted in a different colour.

Test 4: Random rainfall

This scenario cannot really be validated against an analytical solution. The rainfall is constructed using a random walk function, as described in section 2.2.4, so that outputs will be different every time. The only thing to test currently is that the rainfall is indeed random; when running with exactly the same conditions, different outputs are produced. The plots in Fig. 4 show no correlation between runs. The random function generates a different output with each run.

In this section we successfully verified the correctness of the numerical model for various different parameters and rainfall scenarios. We also carried out additional experiments (not illustrated) that also confirmed that the numerical model was behaving correctly.

4 Exploring other rainfall scenarios

When testing the model we looked closely at three of the given scenarios. In this section we briefly consider the behaviour of the other scenarios (constant rainfall and semi-random).

First let us consider constant rain with initial condition $x(0) = 1$. Numerical solutions for this scenario are shown in Fig. 5. In these cases, the flow grows or decays until the flow is at the same rate as the rainfall input.

Some example solutions for the semi-random rainfall scenario are plotted in Fig. 6. These plots are qualitatively similar to river hydrographs from real data. During the intervals with no rain, the flow

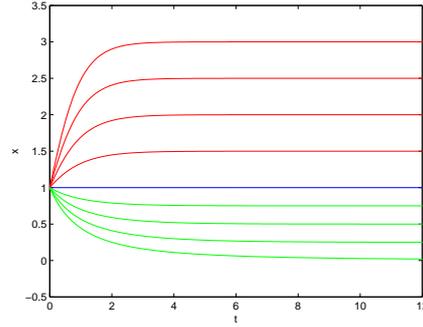


Figure 5: Numerical solutions for the constant rainfall scenario versus time, for $r = 3, 2.5, 2, 1.5, 1, 0.75, 0.5, 0.25, 0.0$

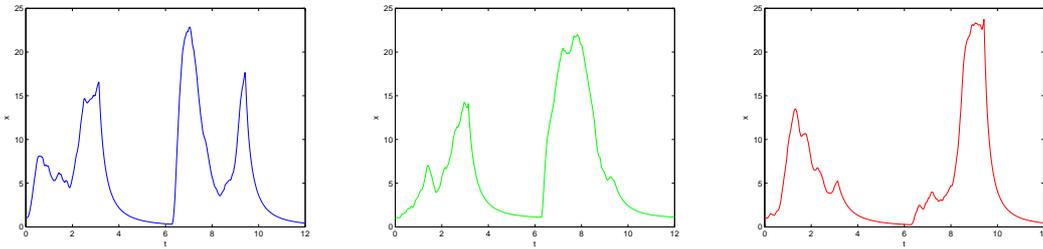


Figure 6: Examples of output from the model with the semi-random scenario. Plots of flow x against time.

decays towards zero, as in the drought scenario.

5 Exploring the parameter space

5.1 Modelling considerations

To acquire a sense of the parameter space, let us consider the derivation of (1) from Moore (2007). The magnitude of flow in a river is considered to be proportional to some power of the catchment storage, $S \equiv S(t)$, i.e. the volume of water held in the storage per unit area. So

$$x = kS^n, \quad k > 0, n > 0, \quad (10)$$

where k is the storage rate coefficient and n is the store exponent. By combining (10) with the equation of continuity

$$\frac{dS}{dt} = (r - x) \quad (11)$$

and denoting

$$a = nk^{1/n}, \quad b = \frac{n-1}{n}, \quad (12)$$

we obtain equation (1). Thus, since k and n are positive, we have $a > 0$ and $0 < b < 1$.

In Moore and Bell (2002), k is given the value $\frac{1}{340}$. Horton (1945), as cited in Moore and Bell (2002), used nonlinear storage models such as this one to describe the overland flow processes and found that $n = \frac{5}{3}$ for fully turbulent flow and $n = 3$ for fully laminar flow. Therefore taking n between $\frac{5}{3}$ and 3 makes sure the model is somewhat realistic. In fact $n=3$ is considered to represent groundwater storage the best (Moore, 2007). Thus, according to this literature, a is between $(\frac{5}{3})(\frac{1}{340})^{\frac{3}{5}} \approx 0.05$ and $3(\frac{1}{340})^{\frac{1}{3}} \approx 0.43$ and b should be between $\frac{2}{5} = 0.4$ and $\frac{2}{3} = 0.6$.

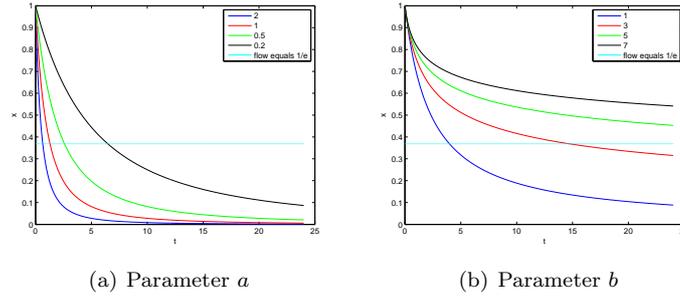


Figure 7: Model solutions as a function of time, using the drought scenario for different values of parameters a and b .

5.2 Model Solutions with a Range of Parameters

Let us consider the qualitative features of model solutions for a range of a and b . The drought setting is used as it is the simplest rainfall scenario. Consider figure 7(a). For $a > 1$, the flow decreases very quickly. Even with $a = 1$, the curve is steep with a flow less than 0.1 after 5 units, $a = 0.5$ seems to give a good maximum value giving 10 units of time for the flow to reach 0.1. For $b > 1$, the graph in figure 7(b) shows that it takes a very long time for the flow to decrease to zero: they do not seem to decay asymptotically to zero as expected in the ‘drought’ scenario. This analysis shows us that $0 < a \leq 0.5$ is a good range for a and $0 \leq b \leq 1$. However this is just by considering the plots. A more quantitative way to find the values of a and b would be to consider the e-folding time.

The e-folding time, in the case of exponential decay, refers to the timescale for a quantity to decrease to $1/e$ of its previous value. Here the initial flow is given a value of 1 so the e-folding time is the time taken for the flow x to reach $\frac{1}{e} \approx 0.3679$. This time will help tell us if the parameters a and b set a realistic model timescale. For the drought scenario, the e-folding time t_e and the parameters a, b are related by the following equation,

$$t_e = \frac{1}{ab}(e^b - 1). \quad (13)$$

Proof. With $r=0$ (since the rainfall scenario is drought), (1) becomes

$$\frac{dx}{dt} = -ax^{b+1}. \quad (14)$$

With the initial condition $x(0) = 1$, we find that the solution is

$$x = \frac{1}{\sqrt[b]{abt + 1}}. \quad (15)$$

To find the e-folding time, t_e , the flow x is equal to $\frac{1}{e}$,

$$\frac{1}{e} = \frac{1}{(abt_e + 1)^{\frac{1}{b}}}. \quad (16)$$

Rearranging gives equation (13) as required. \square

Equation (13) tells us a lot about the values of a and b . In particular, the bigger the value of a , the smaller the value of t_e . As b gets bigger, t_e grows exponentially larger.

Using the range of values for a and b given in section 5.1, we have approximately the bounds, $3.307 < t_e < 24.367$.

6 Data assimilation and the Kalman filter

There are three main sources of uncertainty in flood modelling: input (rainfall) uncertainty, model uncertainty and output uncertainty (Giannakopoulou, 2008). Concentrating on the first two types: input uncertainty breaks down into forecast and measured rainfall errors and the model uncertainty into parameter errors and model structure errors.

Given a noisy discrete model of a dynamical system and noisy observations, data assimilation seeks to find improved estimates of the state of the system (Livings, 2005). Observations of the current state of a system and the results from the forecast are combined to produce an analysis, taking into account the uncertainty in the observed and modelled data (Petrie and Dance, 2010). It is considered to be the best estimate of the current state of the system. The results are used to initialize the next forecast.

The Kalman Filter is an established form of sequential data assimilation (Giannakopoulou, 2008) which breaks the problem into a two-part cycle:

Forecast Step - the model is used to evolve a previous state estimate (analysis) to find a forecast state at the time of the latest observations.

Analysis Step - the observations are used to update the forecast state to give an improved analysis.

'Filter' is a technical term for a data assimilation scheme that uses only observational data valid up to and including the analysis time (Petrie and Dance, 2010). The Kalman filter is the filter that obtains the minimum mean-square state error estimate and assumes the problem is set up with linear model and observation operators as well as Gaussian statistics (Grewal, 2001).

6.1 Notation

The dimension of the state space of the system is denoted by n and the dimension of the observation space by m .

\mathbf{x} - state vector of size n that describes the state of the forecast model

$\mathbf{x}^t(t_k)$ - the true state of the system at time t_k

$\mathbf{x}^a(t_k)$ - the analysis at this time

$\mathbf{x}^f(t_k)$ - the forecast at this time

$\mathbf{y}(t_k)$ - observation vector of size m at this time

Note that when of the quantities examined occur at the same time, the t_k argument will be ignored. Now assuming that we can use random variables to model errors in the flood forecast let

$$\mathbf{e}^f = \mathbf{x}^f - \mathbf{x}^t \quad \text{and} \quad \mathbf{e}^a = \mathbf{x}^a - \mathbf{x}^t, \quad (17)$$

be the *prior* error, (i.e. the error in the forecast), and the *posterior* error, (i.e. the error in the analysis), respectively. We want the forecasts and analyses to be unbiased so we require that

$$\langle \mathbf{e}^f \rangle = 0, \quad \langle \mathbf{e}^a \rangle = 0, \quad (18)$$

where the angle-brackets denote the expectation.

The error covariance matrices give information on the size and correlation of the forecast and analysis error components,

$$\mathbf{P}^f = \langle (\mathbf{x}^f - \mathbf{x}^t)(\mathbf{x}^f - \mathbf{x}^t)^T \rangle, \quad (19)$$

$$\mathbf{P}^a = \langle (\mathbf{x}^a - \mathbf{x}^t)(\mathbf{x}^a - \mathbf{x}^t)^T \rangle. \quad (20)$$

6.2 Algorithm

The Kalman Filter Algorithm was developed for linear dynamical systems and observations. Let us assume that the system dynamics can be modelled by

$$\mathbf{x}^t(t_{k+1}) = \mathbf{M}\mathbf{x}^t(t_k) + \boldsymbol{\eta}(t_k). \quad (21)$$

Here \mathbf{M} is the state transition matrix of the process from time k to $k + 1$ (and is a known matrix of size $n \times n$). In practice \mathbf{M} might change each time step but here we assume it remains constant, for notational simplicity. $\boldsymbol{\eta}(t_k)$ is a random, Gaussian, unbiased and uncorrelated model noise n -vector, with known $n \times n$ covariance matrix \mathbf{Q} , i.e.

$$\mathbf{Q} = \langle \boldsymbol{\eta}\boldsymbol{\eta}^T \rangle. \quad (22)$$

We assume the noise process is stationary so \mathbf{Q} is not a function of time. Observations are assumed to be modelled by

$$\mathbf{y}(t_k) = \mathbf{H}\mathbf{x}^t(t_k) + \boldsymbol{\epsilon}(t_k). \quad (23)$$

\mathbf{H} is the noiseless $m \times n$ matrix which maps the state vector to observed variable. Here it is assumed to remain stationary over time, for notational simplicity, although in general it need not be. $\boldsymbol{\epsilon}(t_k)$ is a random, Gaussian, unbiased and uncorrelated model noise, with known covariance $m \times m$ matrix \mathbf{R} ,

$$\mathbf{R} = \langle \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T \rangle. \quad (24)$$

We assume the noise process is stationary so \mathbf{R} is not a function of time, although this is not a requirement for the general Kalman filter.

The Kalman Filter algorithm at time t_k is given by the following sequence of equations. The first two project the system into $k + 1$ time and are called time update equations (Welch and Bishop, 2006). The remainder are measurement update equations.

1. State forecast:

$$\mathbf{x}^f(t_{k+1}) = \mathbf{M}\mathbf{x}^a(t_k) \quad (25)$$

2. Forecast Error Covariance matrix:

$$\mathbf{P}^f(t_{k+1}) = \mathbf{M}\mathbf{P}^a(t_k)\mathbf{M}^T + \mathbf{Q} \quad (26)$$

3. Kalman Gain:

$$\mathbf{K}(t_{k+1}) = \mathbf{P}^f(t_{k+1})\mathbf{H}^T[\mathbf{H}\mathbf{P}^f(t_{k+1})\mathbf{H}^T + \mathbf{R}]^{-1} \quad (27)$$

4. Analysis:

$$\mathbf{x}^a(t_{k+1}) = \mathbf{x}^f(t_{k+1}) + \mathbf{K}(t_{k+1})(\mathbf{y}(t_{k+1}) - \mathbf{H}\mathbf{x}^f(t_{k+1})) \quad (28)$$

5. Analysis Error Covariance matrix:

$$\mathbf{P}^a(t_{k+1}) = [\mathbf{I} - \mathbf{K}(t_{k+1})\mathbf{H}]\mathbf{P}^f(t_{k+1}) \quad (29)$$

The algorithmic loop is summarised in Fig. 8. The numbers relate to the steps in the algorithm as defined above.

6.3 Derivation

The following derivations are adapted from the works of Thacker and Lacey (1998) and Grewal (2001).

1. $\mathbf{x}^f(t_{k+1}) = \mathbf{M}\mathbf{x}^a(t_k)$

We take the mean of (21), to provide an equation relating the forecast and analysis state estimates.

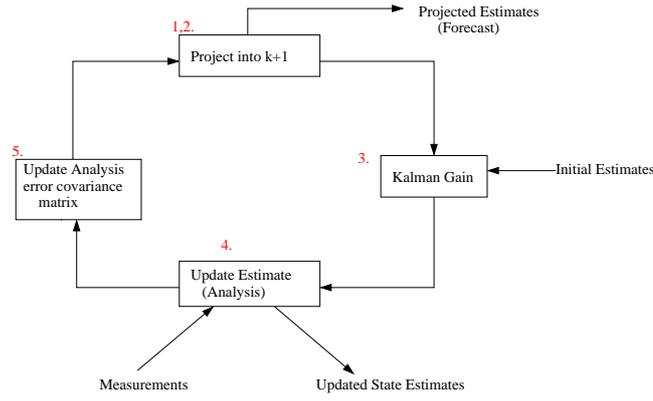


Figure 8: Schematic of the Kalman Filter algorithm adapted from Thacker and Lacey (1998). See text for further details.

$$2. \mathbf{P}^f(t_{k+1}) = \mathbf{M}\mathbf{P}^a(t_k)\mathbf{M}^T + \mathbf{Q}$$

Proof. First let us find an expression for the prior error, using (21) and (25)

$$\begin{aligned} \mathbf{e}^f(t_{k+1}) &= \mathbf{x}^f(t_{k+1}) - \mathbf{x}^t(t_{k+1}) \\ &= \mathbf{M}\mathbf{x}^a(t_k) - (\mathbf{M}\mathbf{x}^t(t_k) - \boldsymbol{\eta}(t_k)) \\ &= \mathbf{M}\mathbf{e}^a(t_k) + \boldsymbol{\eta}(t_k). \end{aligned}$$

Extending equation (19) to time $k + 1$,

$$\begin{aligned} \mathbf{P}^f(t_{k+1}) &= \langle (\mathbf{x}^f(t_{k+1}) - \mathbf{x}^t(t_{k+1}))(\mathbf{x}^f(t_{k+1}) - \mathbf{x}^t(t_{k+1}))^T \rangle \\ &= \langle \mathbf{e}^f(t_{k+1})\mathbf{e}^{fT}(t_{k+1}) \rangle \\ &= \langle [(\mathbf{M}\mathbf{e}^a(t_k) + \boldsymbol{\eta}(t_k))(\mathbf{M}\mathbf{e}^a(t_k) + \boldsymbol{\eta}(t_k))^T] \rangle \\ &= \langle [(\mathbf{M}\mathbf{e}^a(t_k)(\mathbf{M}\mathbf{e}^a(t_k))^T] \rangle + \langle \boldsymbol{\eta}(t_k)\boldsymbol{\eta}^T(t_k) \rangle \\ &= \mathbf{M}\langle (\mathbf{e}^a(t_k))(\mathbf{e}^a(t_k))^T \rangle \mathbf{M}^T + \langle \boldsymbol{\eta}(t_k)\boldsymbol{\eta}^T(t_k) \rangle \\ &= \mathbf{M}\mathbf{P}^a(t_k)\mathbf{M}^T + \mathbf{Q}. \end{aligned}$$

□

$$3. \mathbf{K}(t_{k+1}) = \mathbf{P}^f(t_{k+1})\mathbf{H}^T[\mathbf{H}\mathbf{P}^f(t_{k+1})\mathbf{H}^T + \mathbf{R}]^{-1}$$

This choice of \mathbf{K} minimises the mean squared error.

Proof. In this proof we will primarily use equation (28) which is proved in our discussion of step 4 below. Consider this equation at time k instead of $k + 1$,

$$\mathbf{x}^a(t_k) = \mathbf{x}^f(t_k) + \mathbf{K}(t_k)(\mathbf{y}(t_k) - \mathbf{H}\mathbf{x}^f(t_k)).$$

Substitute (23),

$$\mathbf{x}^a(t_k) = \mathbf{x}^f(t_k) + \mathbf{K}(t_k)(\mathbf{H}\mathbf{x}(t_k) + \boldsymbol{\epsilon}(t_k) - \mathbf{H}\mathbf{x}^f(t_k)).$$

Substitute this into (20), condensing the notation to subscript k to indicate time t_k

$$\begin{aligned} \mathbf{P}_k^a &= \langle [(\mathbf{I} - \mathbf{K}_k\mathbf{H})(\mathbf{x}_k^t - \mathbf{x}_k^f) - \mathbf{K}_k\boldsymbol{\epsilon}_k][(\mathbf{I} - \mathbf{K}_k\mathbf{H})(\mathbf{x}_k^t - \mathbf{x}_k^f) - \mathbf{K}_k\boldsymbol{\epsilon}_k]^T \rangle \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\langle [(\mathbf{x}_k^t - \mathbf{x}_k^f)(\mathbf{x}_k^t - \mathbf{x}_k^f)^T] \rangle (\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\langle \boldsymbol{\epsilon}_k\boldsymbol{\epsilon}_k^T \rangle \mathbf{K}_k^T \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^f(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T. \end{aligned}$$

This formula is sometimes known as the ‘‘Joseph’’ form of the error covariance update equation (Grewal, 2001). By choosing \mathbf{K}_k as the optimal Kalman gain, we will see that this gives the equation in step 5 of the algorithm.

Expanding the Joseph form gives,

$$\mathbf{P}_k^a = \mathbf{P}_k^f - \mathbf{K}_k \mathbf{H} \mathbf{P}_k^f - \mathbf{P}_k^f \mathbf{K}_k^T \mathbf{H}^T + \mathbf{K}_k (\mathbf{H} \mathbf{P}_k^f \mathbf{H}^T + \mathbf{R}) \mathbf{K}_k^T. \quad (30)$$

Now, the diagonal of the covariance matrix contains the mean squared errors:

$$\text{diag}(\mathbf{P}_k^a) = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix}.$$

Thus the total mean squared error can be minimised by minimising the trace of \mathbf{P}_k^a , which is given by

$$T[\mathbf{P}_k^a] = T[\mathbf{P}_k^f] - 2T[\mathbf{K}_k \mathbf{H} \mathbf{P}_k^f] + T[\mathbf{K}_k (\mathbf{H} \mathbf{P}_k^f \mathbf{H}^T + \mathbf{R}) \mathbf{K}_k^T], \quad (31)$$

where $T[\mathbf{M}]$ is the trace of matrix \mathbf{M} . Note that the trace of the a matrix and the trace of its transpose are equal. To minimise the trace of \mathbf{P}_k^a , we differentiate it with respect to \mathbf{K}_k to give

$$\frac{dT[\mathbf{P}_k^a]}{d\mathbf{K}_k} = -2(\mathbf{H} \mathbf{P}_k^f)^T + 2\mathbf{K}_k (\mathbf{H} \mathbf{P}_k^f \mathbf{H}^T + \mathbf{R}) \quad (32)$$

and set the result to zero (Thacker and Lacey, 1998). Rearranging, we find

$$\begin{aligned} (\mathbf{H} \mathbf{P}_k^f)^T &= \mathbf{K}_k (\mathbf{H} \mathbf{P}_k^f \mathbf{H}^T + \mathbf{R}) \\ \mathbf{K}_k &= \mathbf{P}_k^f \mathbf{H}^T [\mathbf{H} \mathbf{P}_k^f \mathbf{H}^T + \mathbf{R}]^{-1} \end{aligned}$$

It can shown that this value of \mathbf{K}_k indeed minimises \mathbf{P}_k^a (Gelb, 1974). \square

4. $\mathbf{x}^a(t_{k+1}) = \mathbf{x}^f(t_{k+1}) + \mathbf{K}(t_{k+1})(\mathbf{y}(t_{k+1}) - \mathbf{H}\mathbf{x}^f(t_{k+1}))$

Proof. In this derivation, we start off by wanting to find unbiased minimum variance estimator of the state at time $k + 1$ of the form

$$\mathbf{x}^a(t_{k+1}) = \mathbf{K}' \mathbf{x}^a(t_k) + \mathbf{K} \mathbf{y}(t_{k+1}). \quad (33)$$

We want to find \mathbf{K}' , which should give us then the equation we want. Note in this proof $\mathbf{K}(t_{k+1})$ and $\mathbf{K}'(t_{k+1})$ are just written as just \mathbf{K} and \mathbf{K}' and are unknown matrices. Subtract $\mathbf{x}^t(t_{k+1})$ from both sides of (33) and add and subtract the term $\mathbf{K}' \mathbf{x}^t(t_k)$ on the RHS to give

$$\mathbf{x}^a(t_{k+1}) - \mathbf{x}^t(t_{k+1}) = \mathbf{K}' \mathbf{x}^a(t_k) + \mathbf{K} \mathbf{y}(t_{k+1}) - \mathbf{x}^t(t_{k+1}) - \mathbf{K}' \mathbf{x}^t(t_k) + \mathbf{K}' \mathbf{x}^t(t_k). \quad (34)$$

Rearrange and substitute (23) in (34), to give

$$\mathbf{x}^a(t_{k+1}) - \mathbf{x}^t(t_{k+1}) = \mathbf{K}' [\mathbf{x}^a(t_k) - \mathbf{x}^t(t_k)] + \mathbf{K} (\mathbf{H} \mathbf{x}^t(t_{k+1}) + \boldsymbol{\epsilon}(t_{k+1})) - \mathbf{x}^t(t_{k+1}) + \mathbf{K}' \mathbf{x}^t(t_k). \quad (35)$$

Substitute (21) in the right hand side of (35) to give

$$\begin{aligned} \mathbf{x}^a(t_{k+1}) - \mathbf{x}^t(t_{k+1}) & \quad (36) \\ &= \mathbf{K}' [\mathbf{x}^a(t_k) - \mathbf{x}^t(t_k)] + \mathbf{K} (\mathbf{H} (\mathbf{M} \mathbf{x}^t(t_k) + \boldsymbol{\eta}(t_k)) + \boldsymbol{\epsilon}(t_{k+1})) - (\mathbf{M} \mathbf{x}^a(t_k) + \boldsymbol{\eta}(t_k)) + \mathbf{K}' \mathbf{x}^t(t_k) \\ &= \mathbf{K}' [\mathbf{x}^a(t_k) - \mathbf{x}^t(t_k)] + \mathbf{x}^t(t_k) [\mathbf{K} \mathbf{H} \mathbf{M} - \mathbf{M} + \mathbf{K}'] + \boldsymbol{\eta}(t_k) + \mathbf{K} \boldsymbol{\epsilon}(t_{k+1}) \end{aligned} \quad (38)$$

Taking the expectation of both sides, using the unbiased property as shown in (18) yields,

$$\langle \mathbf{x}^a(t_{k+1}) - \mathbf{x}^t(t_{k+1}) \rangle = \langle \mathbf{e}^a \rangle = 0 = [\mathbf{K} \mathbf{H} \mathbf{M} - \mathbf{M} + \mathbf{K}'] \langle \mathbf{x}^t(t_k) \rangle. \quad (39)$$

Since $\langle \mathbf{x}^t(t_k) \rangle$ is non-zero, we must have that $[\mathbf{K} \mathbf{H} \mathbf{M} - \mathbf{M} + \mathbf{K}'] = 0$. So we get $\mathbf{K}' = \mathbf{M}(\mathbf{I} - \mathbf{K} \mathbf{H})$, hence

$$\mathbf{x}^a(t_{k+1}) = (\mathbf{I} - \mathbf{K} \mathbf{H}) \mathbf{M} \mathbf{x}^a(t_k) + \mathbf{K} \mathbf{y}(t_{k+1}) \quad (40)$$

$$\mathbf{x}^a(t_{k+1}) = \mathbf{M} \mathbf{x}^a(t_k) + \mathbf{K} (\mathbf{y}(t_{k+1}) - \mathbf{H} \mathbf{M} \mathbf{x}^a(t_k)) \quad (41)$$

Using (25) we get $\mathbf{x}^a(t_{k+1}) = \mathbf{x}^f(t_{k+1}) + \mathbf{K}(t_{k+1})(\mathbf{y}(t_{k+1}) - \mathbf{H}\mathbf{x}^f(t_{k+1}))$ as required. \square

This equation relates $\mathbf{x}^t(t_k)$ and $\mathbf{x}^a(t_k)$ by writing $\mathbf{x}^a(t_k)$ as a linear combination of the old estimate, $\mathbf{x}^f(t_k)$, and a weighted differences between the true measurement and the measurement prediction $\mathbf{H}\mathbf{x}^f(t_k)$ (Giannakopoulou, 2008). Note that the fact the forecast and analysis were unbiased was crucial to this proof.

$$5. \mathbf{P}^a(t_{k+1}) = [\mathbf{I} - \mathbf{K}(t_{k+1})\mathbf{H}]\mathbf{P}^f(t_{k+1})$$

Proof. As noted in the discussion of step 3 above this is derivative of the ‘Joseph form’ specific to the Kalman filter since it depends on Kalman gain equation. Let us start with the Joseph form and expand it as done above, again using the notation subscript k for time t_k

$$\begin{aligned} \mathbf{P}_k^a &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^f(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T \\ &= \mathbf{P}_k^f - \mathbf{K}_k\mathbf{H}\mathbf{P}_k^f - \mathbf{P}_k^f\mathbf{K}_k^T\mathbf{H}^T + \mathbf{K}_k(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})\mathbf{K}_k^T. \end{aligned}$$

Now substitute the Kalman gain equation (27),

$$\begin{aligned} \mathbf{P}_k^a &= \mathbf{P}_k^f - \mathbf{P}_k^f\mathbf{H}^T[\mathbf{H}\mathbf{P}_{k+1}^f\mathbf{H}^T + \mathbf{R}]^{-1}\mathbf{H}\mathbf{P}_k^f \\ &= \mathbf{P}_k^f - \mathbf{K}_k\mathbf{H}\mathbf{P}_k^f \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^f. \end{aligned}$$

□

7 The Ensemble Kalman Filter (EnKF)

The Kalman Filter is a good method of data assimilation for linear systems, but the model we will be applying a filter to is a non-linear system. The Ensemble Kalman Filter is intended as a method to adapt the Kalman filter for non-linear problems. The idea behind it is to use a statistical sample (ensemble) of state estimates instead of a single estimate. Then the forecast error covariance matrix may be calculated from this sample and used to calculate a common Kalman gain, in order to update each ensemble member in the analysis step (Livings, 2005). The mean of this sample represents the “best” state estimate and the variance a measure of the spread of the ensemble.

7.1 Notation

Let \mathbf{x}_i , ($i = 1, \dots, N$) be a member of an ensemble in state space. N denotes the size of the ensemble and the dimension of a state vector is n . We write the ensemble mean as

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i. \quad (42)$$

We also define an ensemble perturbation matrix as

$$\mathbf{X} = \frac{1}{\sqrt{N-1}} (\mathbf{x}_1 - \bar{\mathbf{x}} \quad \mathbf{x}_2 - \bar{\mathbf{x}} \quad \dots \quad \mathbf{x}_N - \bar{\mathbf{x}}). \quad (43)$$

The n -vectors $\mathbf{x}_i - \bar{\mathbf{x}}$ form the columns of \mathbf{X} so the matrix is of dimension $(n \times N)$. With is definition, the ensemble covariance matrix, of dimension $n \times n$ is given by

$$\mathbf{P}_e = \mathbf{X}\mathbf{X}^T = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T. \quad (44)$$

The division by $N-1$ ensures that \mathbf{P}_e is an unbiased estimate of the true state error covariance matrix \mathbf{P} . Superscript a and f refer to analysis and forecast respectively as before, so that \mathbf{x}_i^a and \mathbf{x}_i^f are analysis and forecast ensemble members respectively. Similarly, \mathbf{X}^f , represents a forecast ensemble perturbation matrix, and $\mathbf{P}_e^f = \mathbf{X}^f(\mathbf{X}^f)^T$, the ensemble forecast error covariance.

Let \mathbf{y} be an observation of dimension p with the observation error covariance matrix \mathbf{R} of size $p \times p$. The p -vector \mathbf{y}_i^f represents the model equivalent observation for each ensemble member, given by

$$\mathbf{y}_i^f = \mathbf{H}\mathbf{x}_i^f, \quad (45)$$

where \mathbf{H} is an observation operator which maps the state vector to the observation vector. We can also construct \mathbf{Y}^f , a $p \times N$ ensemble perturbation matrix as before. If \mathbf{H} is linear, this is simply given by

$$\mathbf{Y}^f = \mathbf{H}\mathbf{X}^f. \quad (46)$$

This definition is easily generalisable to a nonlinear observation operator (Livings, 2005).

7.2 Algorithm properties

The forecast step projects each ensemble member forwards in time using a nonlinear model, M , such that

$$\mathbf{x}_i^f(t_k) = M(\mathbf{x}_i^a(t_{k-1})) + \boldsymbol{\eta}_i(t_{k-1}). \quad (47)$$

Here $\boldsymbol{\eta}_i(t_{k-1})$ is noise process from a distribution with mean zero and known covariance matrix \mathbf{Q} (Evensen, 2003).

In the linear case, the Kalman Filter update equations are optimal so we want to mimic these in the ensemble version of the analysis step. Thus, we wish to satisfy

$$\begin{aligned} \mathbf{K}_e &= \mathbf{P}_e^f \mathbf{H}^T [\mathbf{H} \mathbf{P}_e^f \mathbf{H}^T + \mathbf{R}]^{-1} \\ &= \mathbf{X}^f (\mathbf{X}^f)^T \mathbf{H}^T [\mathbf{H} \mathbf{X}^f (\mathbf{X}^f)^T \mathbf{H}^T + \mathbf{R}]^{-1} \\ &= \mathbf{X}^f (\mathbf{H} \mathbf{X}^f)^T [\mathbf{H} \mathbf{X}^f (\mathbf{H} \mathbf{X}^f)^T + \mathbf{R}]^{-1} \\ &= \mathbf{X}^f (\mathbf{Y}^f)^T [\mathbf{Y}^f (\mathbf{Y}^f)^T + \mathbf{R}]^{-1} \\ &= \mathbf{X}^f (\mathbf{Y}^f)^T \mathbf{S}^{-1}, \end{aligned} \quad (48)$$

where \mathbf{S} is defined by

$$\mathbf{S} = \mathbf{Y}^f (\mathbf{Y}^f)^T + \mathbf{R}. \quad (49)$$

We want the ensemble mean update to satisfy

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{K}_e (\mathbf{y} - \overline{\mathbf{y}}^f), \quad (50)$$

and the analysis ensemble covariance matrix to satisfy

$$\begin{aligned} \mathbf{X}^a (\mathbf{X}^a)^T = \mathbf{P}_e^a &= (\mathbf{I} - \mathbf{K}_e \mathbf{H}) \mathbf{P}_e^f \\ &= (\mathbf{I} - \mathbf{X}_f (\mathbf{Y}^f)^T \mathbf{S}^{-1} \mathbf{H}) \mathbf{X}^f (\mathbf{X}^f)^T \\ &= \mathbf{X}^f (\mathbf{I} - (\mathbf{Y}^f)^T \mathbf{S}^{-1} \mathbf{Y}^f) (\mathbf{X}^f)^T. \end{aligned} \quad (51)$$

Observe that we may write this equation in a square root form,

$$\mathbf{X}^a = \mathbf{X}^f \mathbf{T}, \quad (52)$$

where \mathbf{T} is an $(N \times N)$ matrix which satisfies the equation

$$\mathbf{T} \mathbf{T}^T = \mathbf{I} - (\mathbf{Y}^f)^T \mathbf{S}^{-1} \mathbf{Y}^f. \quad (53)$$

Note that \mathbf{T} is not unique since it can be replaced by $\mathbf{T}\mathbf{U}$ where \mathbf{U} is an arbitrary $(N \times N)$ orthogonal matrix¹(Livings et al., 2008). Unless the choice of \mathbf{T} is made carefully, it can lead to some undesirable properties of the algorithm (Livings et al., 2008). Ensemble Kalman Filter implementations differ in the way they compute \mathbf{T} .

¹An orthogonal matrix satisfies $\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I}$. So $(\mathbf{T}\mathbf{U})(\mathbf{T}\mathbf{U})^T = \mathbf{T}\mathbf{U}\mathbf{U}^T \mathbf{T}^T = \mathbf{T}\mathbf{T}^T$ and so $\mathbf{T}\mathbf{U}$ also satisfies the equation if \mathbf{T} does.

7.3 The Ensemble Transform Kalman Filter (ETKF)

In this project we use a version of the Ensemble Transform Kalman Filter, introduced in (Bishop et al., 2001). The method uses the following identity²

$$\mathbf{I} - (\mathbf{Y}^f)^T \mathbf{S}^{-1} \mathbf{Y}^f = (\mathbf{I} + (\mathbf{Y}^f)^T \mathbf{R}^{-1} \mathbf{Y}^f)^{-1}. \quad (54)$$

This identity means we can work out $(\mathbf{Y}^f)^T \mathbf{R}^{-1} \mathbf{Y}^f$ instead of $(\mathbf{Y}^f)^T \mathbf{S}^{-1} \mathbf{Y}^f$ which is usually much easier to do. \mathbf{R} usually has a simple structure, whereas \mathbf{S} is usually hard to invert (Livings, 2005).

Then we consider the eigenvalue decomposition given by

$$(\mathbf{Y}^f)^T \mathbf{R}^{-1} \mathbf{Y}^f = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T, \quad (55)$$

where \mathbf{U} is an $(N \times N)$ orthogonal matrix and $\mathbf{\Lambda}$ is a diagonal one. So the identity (54) becomes

$$\begin{aligned} \mathbf{I} - (\mathbf{Y}^f)^T \mathbf{S}^{-1} \mathbf{Y}^f &= (\mathbf{I} + (\mathbf{Y}^f)^T \mathbf{R}^{-1} \mathbf{Y}^f)^{-1} \\ &= (\mathbf{I} + \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T)^{-1} \\ &= \mathbf{U} (\mathbf{I} + \mathbf{\Lambda})^{-1} \mathbf{U}^T. \end{aligned} \quad (56)$$

From equation (53), we get $\mathbf{T} \mathbf{T}^T = \mathbf{U} (\mathbf{I} + \mathbf{\Lambda})^{-1} \mathbf{U}^T$ and hence

$$\mathbf{T} = \mathbf{U} (\mathbf{I} + \mathbf{\Lambda})^{-\frac{1}{2}} \mathbf{U}^T. \quad (57)$$

is a symmetric matrix square root. Note that $(\mathbf{I} + \mathbf{\Lambda})$ is a diagonal matrix and easy to compute. The ETKF using this value of \mathbf{T} is known as the Revised ETKF (Giannakopoulou, 2008). It is unbiased and gives us the updated ensemble perturbation matrix as

$$\begin{aligned} \mathbf{X}^a &= \mathbf{X}^f \mathbf{T} \\ &= \mathbf{X}^f \mathbf{U} (\mathbf{I} + \mathbf{\Lambda})^{-\frac{1}{2}} \mathbf{U}^T. \end{aligned} \quad (58)$$

The actual code used is the revised ETKF code of Livings (2005). This differs from the implementation described here in that it uses the singular value decomposition rather than the eigenvalue decomposition. This is expected to give better numerical accuracy. The filter code was validated by Livings (2005) in his experiments.

8 Varying the assimilation parameters

We now consider some initial assimilation experiments. To concentrate on the effects of varying the assimilation parameters, let us fix the model parameters as $a = 3(\frac{1}{340})^{\frac{1}{3}} \approx 0.43$ and $b = \frac{2}{3} = 0.6$. These values are considered the optimal values for these parameters as seen in section 5.1. We will start the experiments by using the ‘regular’ rainfall scenario. Let us also fix the time span at 15 units. The assimilation parameters we will consider are: (1) ensemble size, (2) observation and background error covariances, (3) observation frequency.

In the following experiments we generate a “truth” trajectory by running the model forward in time. The observations are generated by applying the observation operator to the truth trajectory, to create perfect observations. Imperfect observations are generated by adding Gaussian random noise consistent with the observation error covariance to the perfect observations. The initial ensemble is generated by adding Gaussian random noise consistent with the initial background error covariance to the true initial conditions. Some care is taken to ensure that there are no ensemble members with initially unphysical negative flows.

The graph in Fig. 9 shows how the ETKF works, with the experimental parameters explained in the caption. The observations are spread out evenly every 1.5 timesteps in this example. The filter brings

²This identity is easy to prove by multiplying $\mathbf{I} - (\mathbf{Y}^f)^T \mathbf{S}^{-1} \mathbf{Y}^f$ by $\mathbf{I} + (\mathbf{Y}^f)^T \mathbf{R}^{-1} \mathbf{Y}^f$ and using the definition of \mathbf{S} .

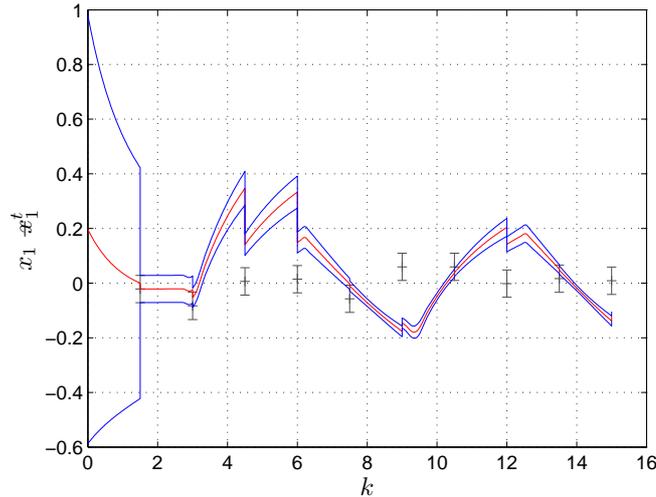


Figure 9: The red line represents the ensemble mean - true solution against time, so that the true state would be represented by the dotted line at zero. The two blue lines either side show the ensemble mean \pm the ensemble standard deviation. The observations are plotted as + with error bars which indicate the standard deviation of the observations. Experimental parameters: Imperfect observations, regular rainfall, 10 observations, observation error variance = 0.05, 50 ensemble members.

the ensemble back to towards the true value; at every observation the graph is ‘pulled’ back towards the truth. This can be seen very clearly at the first observation where the ensemble mean \pm the ensemble standard deviation lines show a spread of about 1 immediately before the first observation is assimilated and just 0.5 immediately after the first observation is assimilated.

8.1 Ensemble size

Let us investigate the differences caused by ensemble size. For the following experiments, we use ‘regular’ rainfall, 10 observations and an initial background error covariance that is equal to 1. We consider two cases: perfect and imperfect observations.

Case I: Perfect observations

For these experiments we use an observation error variance of 0.05. The results are shown in Fig. 10. We expect that as the ensemble size increases, the filter estimates should get closer to the truth. The expected rapid initial decrease in the filter error can be seen on all the graphs however after about 6 units in time, the filter settles into an oscillation around the truth, the general error only decreasing slightly. In fact, the true solution is not always captured in the spread of the ensemble, indicating that the ensemble error covariance is an underestimate of the true statistics. Since our filter does not include any inflation or localization steps, this is to be expected for small ensemble sizes (Hamill et al., 2001). The ensemble mean does get closer to the true trajectory as the ensemble size increases, however after about ten ensemble members, neither the mean nor the ensemble spread improves further.

Case II: Imperfect observations

For these experiments the observation error variance is set at a slightly large value of 0.2. The results for this set of experiments are shown in Fig. 11. The results are similar to the case with perfect observations. There is also a rapid initial decrease in error in the ensemble mean, which then oscillates around the truth. The 10 ensemble members are shown to be a sufficient size for experiments because more members does not improve the filter output further. Overall, these experiments have given better results than those with the perfect observations, however it is not

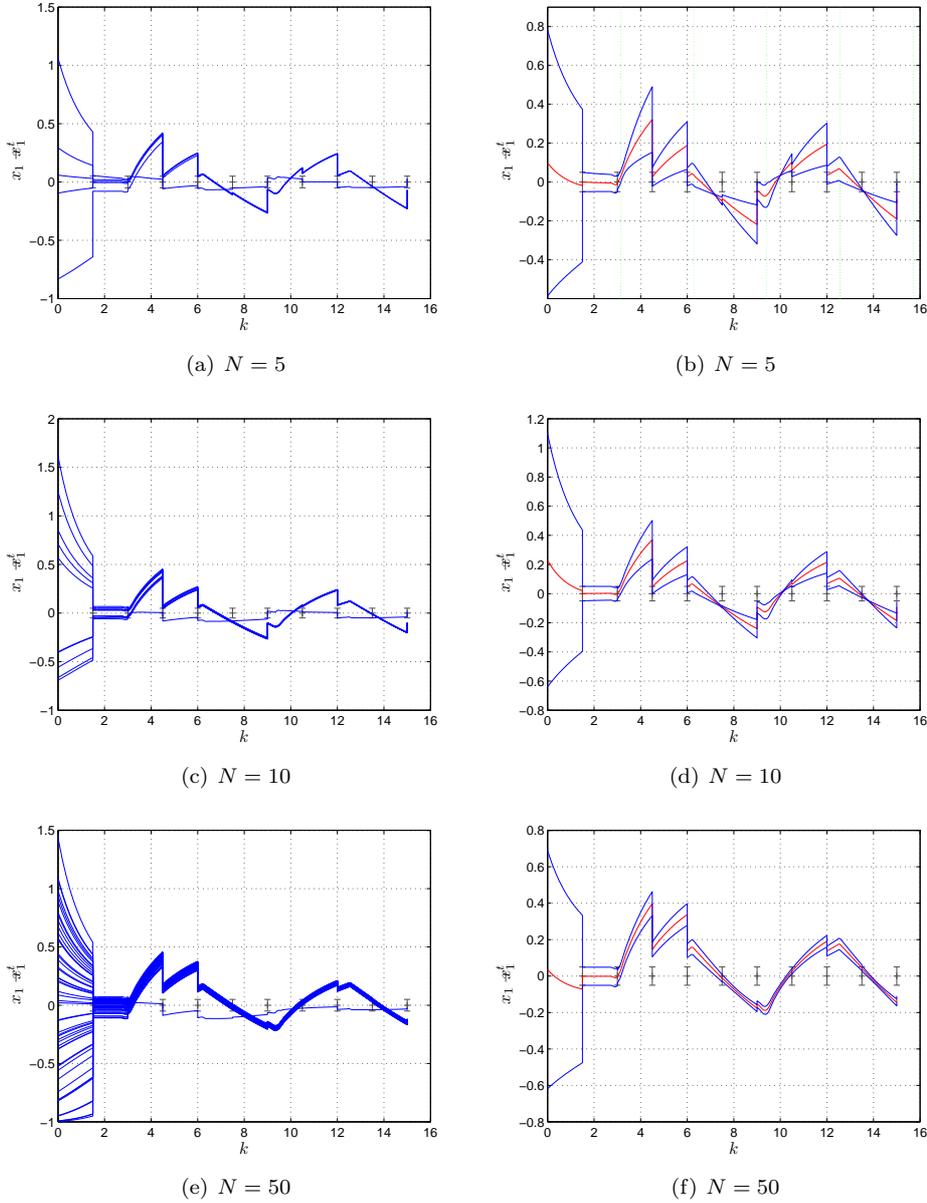


Figure 10: Ensemble size experiments with perfect observations. Left column: The solutions for each ensemble member relative to the truth, as a function of time, are plotted as solid lines. The truth is indicated by the dotted line at zero. The observations are plotted as + with error bars which indicate the standard deviation of the observations. Right column: The red line represents the ensemble mean - true solution against time, so that the true state would be represented by the dotted line at zero. The two blue lines either side show the ensemble mean \pm the ensemble standard deviation. The observations are plotted as + with error bars which indicate the standard deviation of the observations.

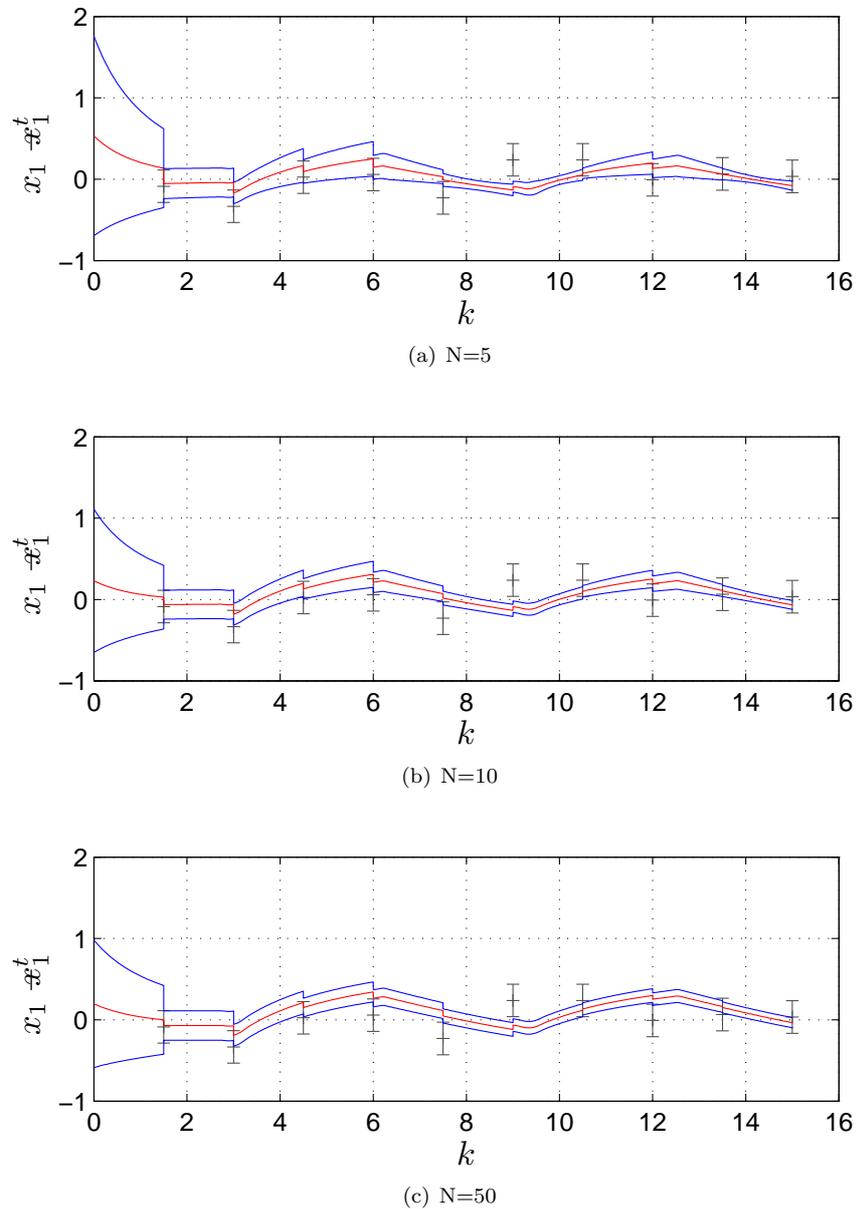


Figure 11: Ensemble size experiments with imperfect observations. The red line represents the ensemble mean - true solution against time, so that the true state would be represented by the dotted line at zero. The two blue lines either side show the ensemble mean \pm the ensemble standard deviation. The observations are plotted as + with error bars which indicate the standard deviation of the observations.

clear if this is due to the increased observation error variance, or the presence of noise on the observations.

8.2 Observation and background error covariances

In this section we consider the effects of varying the observation and background error covariances. We will use imperfect observations, ‘regular’ rainfall, 10 observations and 10 ensemble members.

Case I : Observation error covariance

For these experiments the background error covariance was set equal to 1 whilst the observation error covariance was changed. The results are illustrated in Fig. 12. For small observation error covariance values, (≤ 0.1) the ensemble mean lies very close to the noisy observations. There are intervals where the true trajectory is not inside the range set by the ensemble mean \pm ensemble standard deviation. For intermediate size observation error covariances the analysis ensemble mean values tend to lie between the forecast value and the observation. The difference between the ensemble mean and the true trajectory is smaller than in the previous case since there are fewer intervals where the true state is not included in the range of the ensemble spread. When the observation error covariance, however, is bigger than 1 the filter seems to almost ignore the observations and stay close to the model forecast.

Case II : Background error covariance

For these experiments, we fix the observation error covariance at 0.5, and vary the initial background error covariance. The initial background error covariance affects the initial spread of the ensemble members. These results are shown in Fig. 13. As the initial background error covariance increases the spread of the ensemble members increases, the maximum difference between the truth and the filter being approximately equal to its value. In cases where the spread is larger than twice the observation error covariance, (as in the bottom four plots), the assimilation of the first observation rapidly decreases the spread and the error in the filter mean. However when the initial background covariance is less than the observation error covariance, the range denoted by the ensemble mean \pm ensemble standard deviation does not always include the truth but oscillates around this point with an error of approximately 0.2. This is because the ensemble is overconfident and ignores the first few observations, even though the ensemble members differ from the truth. When the initial background and observation covariances are equal, the filter spread starts to include the true trajectory at some values but still has intervals where the true state is outside the ensemble mean \pm ensemble standard deviation lines.

We can explain the behaviour we have seen in these two cases if we consider the Kalman gain, (27). Since our model is very simple, \mathbf{H} is taken to equal the identity, and \mathbf{P}^f and \mathbf{R} are scalars, Thus

$$\mathbf{K} = \frac{\mathbf{P}^f}{\mathbf{P}^f + \mathbf{R}} \quad (59)$$

and the solution for the analysis may be written

$$\mathbf{x}^a = \frac{\mathbf{R}\mathbf{x}^f + \mathbf{P}^f\mathbf{y}}{\mathbf{P}^f + \mathbf{R}}. \quad (60)$$

Thus the value of the ratio between \mathbf{P}^f and \mathbf{R} determines the relative weighting on the background and observations.

8.3 Observation frequency

Let us examine the differences in filtering solutions caused by observation frequency. Still using ‘regular’ rainfall, 10 ensemble members and an observation variance of 0.5, let us break the experiment down into

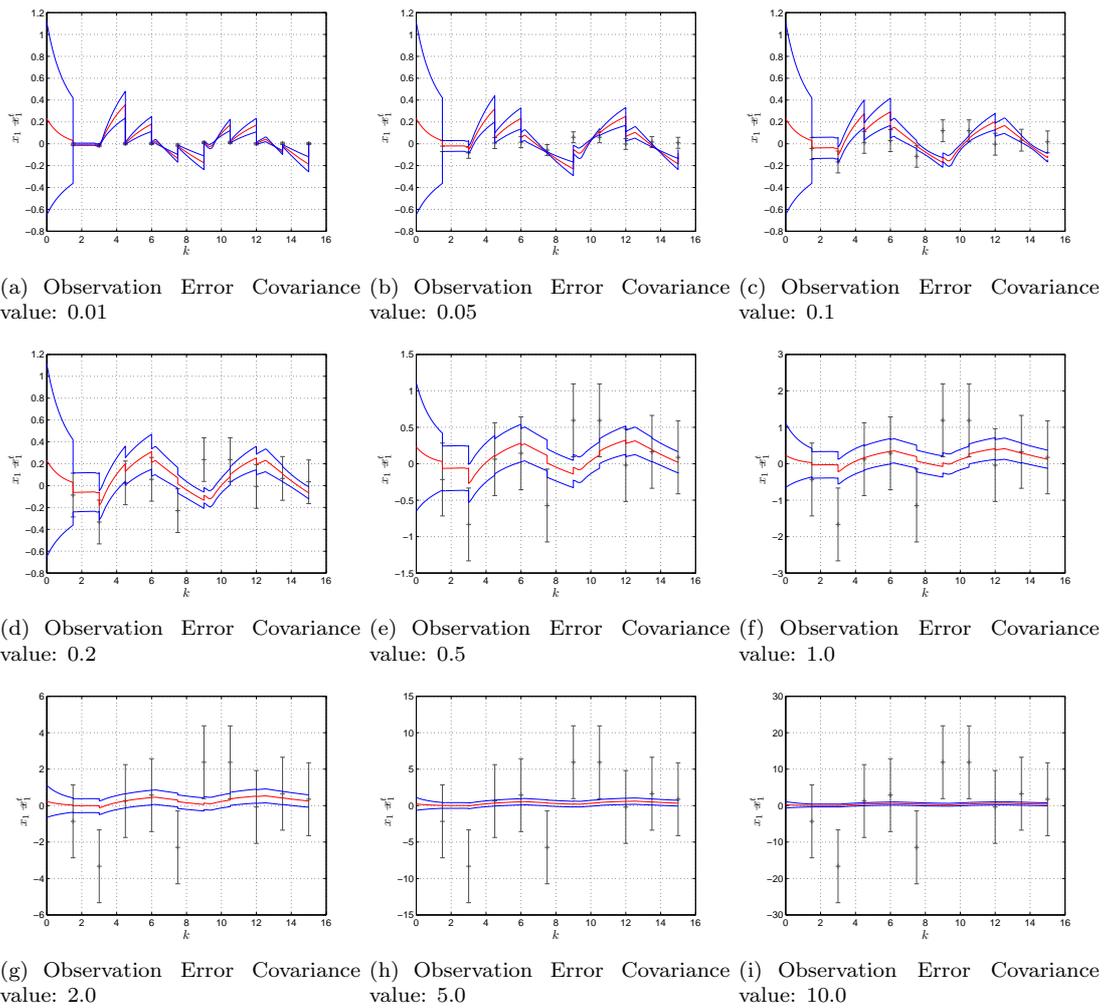
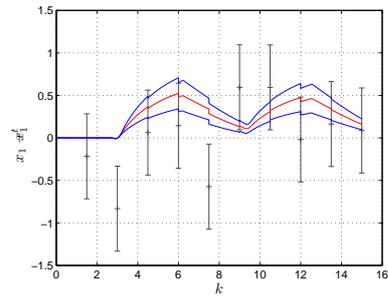
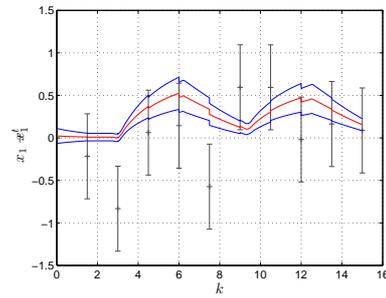


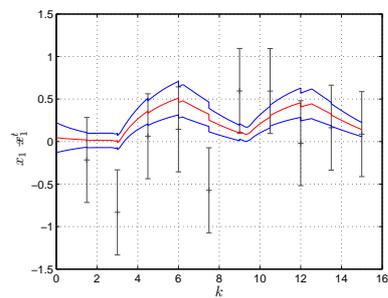
Figure 12: Experiments varying the size of the observation error covariance. Colours and symbols as in Fig. 11.



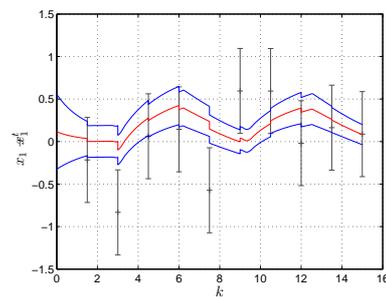
(a) Background Error Covariance value: 0.0



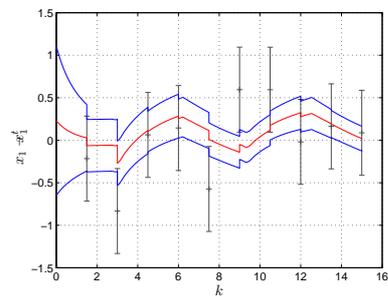
(b) Background Error Covariance value: 0.1



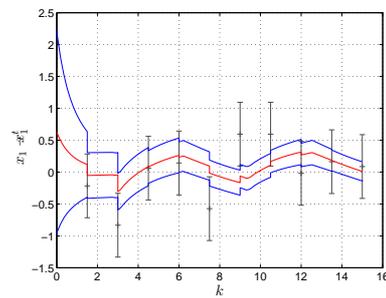
(c) Background Error Covariance value: 0.2



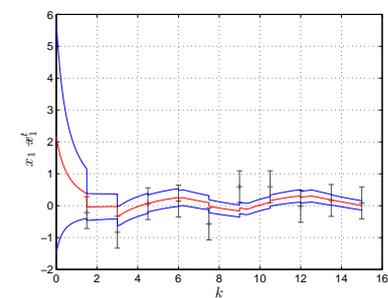
(d) Background Error Covariance value: 0.5



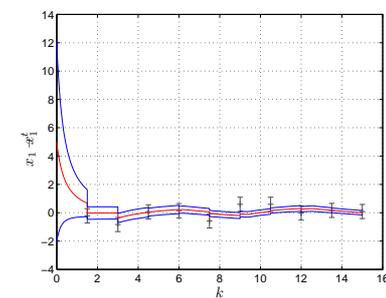
(e) Background Error Covariance value: 1.0



(f) Background Error Covariance value: 2.0



(g) Background Error Covariance value: 5.0



(h) Background Error Covariance value: 10.0

Figure 13: Different initial background error covariance values with imperfect observations. Colours and symbols as in Fig. 11.

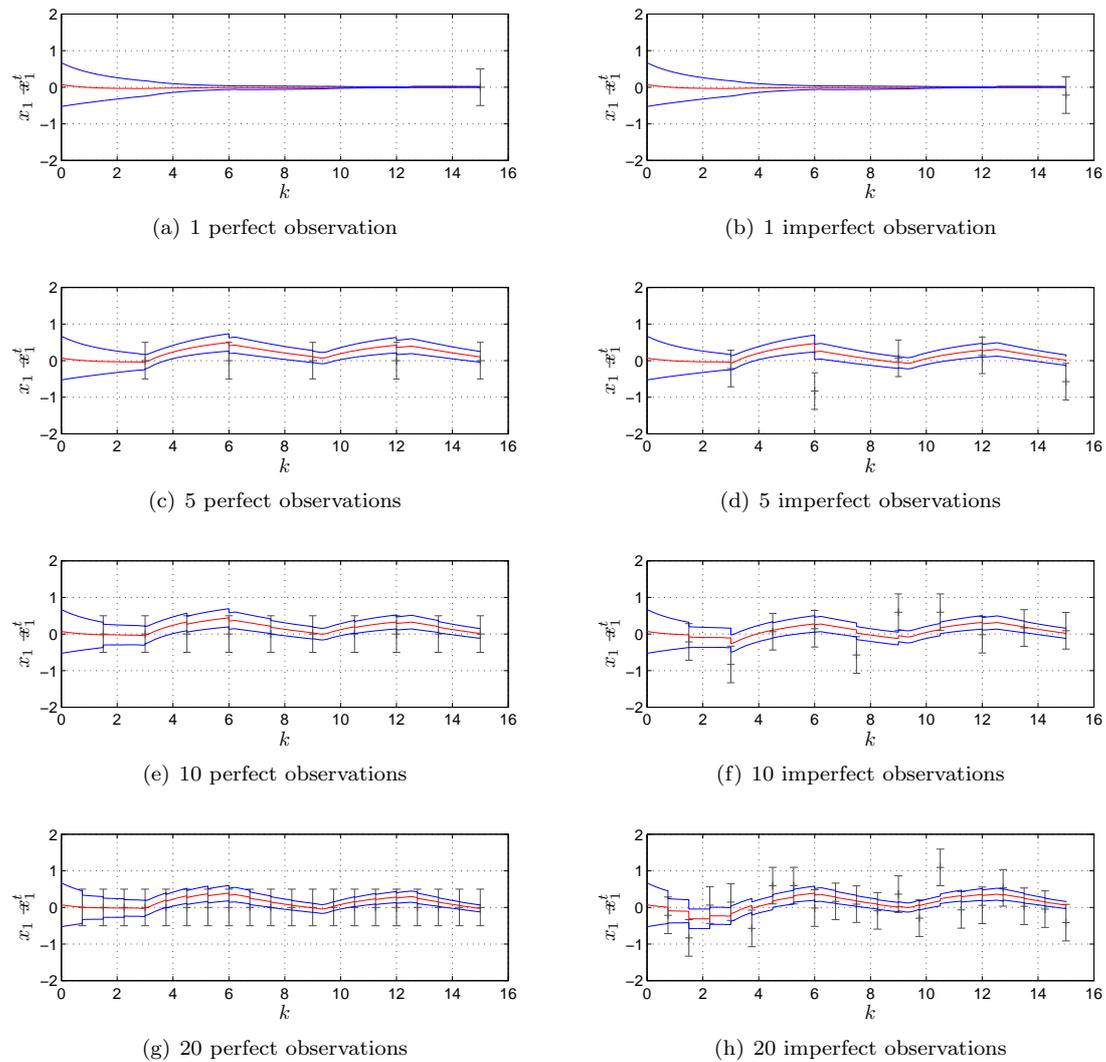


Figure 14: Filter solutions for perfect (left column) and perfect (right column) observations with different frequencies with regular rainfall and an observation error variance of 0.5. The observations are spread over a 15 unit time window. Colours and symbols as in Fig. 11.

two cases, one where the observations are perfect and the other when they are not. This will also help us see the difference between the perfect and imperfect observations.

Fig. 14 shows the results of the two cases of these experiments. When there is only one observation, imperfect and perfect realisations give essentially the same result, since the observation is assimilated right at the end of the time window once the ensemble estimate of the forecast error covariance is very small. Since all solutions for the regular rainfall are characterized by piecewise exponential decay, all ensembles converge towards the truth, regardless of the initial data. For other rainfall scenarios, which do not have such stable solutions, we would expect much worse results with only one observation assimilated.

As we increase the frequency of observations to every third time unit, the filter solutions are more strongly affected by the observations. As we increase the number of observations further, the ensemble standard deviation decreases slightly but the ensemble mean estimate does not improve much further. Indeed with high frequency observations, the standard deviation can decrease too much and the truth is no longer in the ensemble mean \pm ensemble standard deviation range all the time.

A more comprehensive study of the effects of observation frequency should consider different rainfall scenarios such as random or semi-random rainfall.

9 Uncertainty due to rainfall inputs

We have already seen the error in rainfall input is an important area of uncertainty to consider in flood models. For example, errors in rainfall measurement or forecast will lead to inaccurate values of water in store which will affect the flow (Giannakopoulou, 2008).

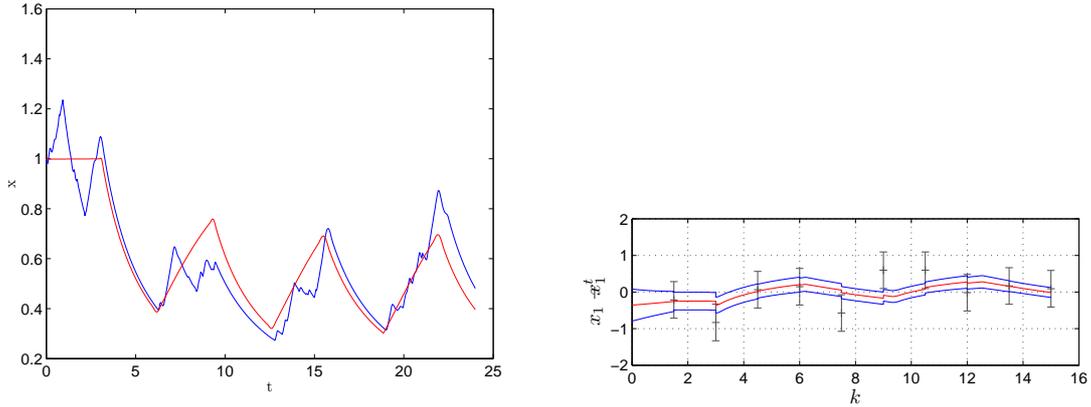
The error from forecast rainfall is one of the biggest sources of uncertainty (Leahy et al., 2007). Using Numerical Weather Prediction (NWP) model outputs, meteorologists forecast the rainfall used in flood warning operations. However NWP models often use a grid size which is larger than the size needed in flood forecasting models. This means the data is not always of the right resolution (Leahy et al., 2007). Errors in the locations or timing of the weather systems by NWP models can result in an inaccurate forecast.

There are still significant uncertainties when the rainfall input is based on recorded observations. Rain gauges, for example, sample very small areas and there can be large gaps between observations (Catchlove et al., 2005). On the other hand, radar methods can be used to sample large areas but are unable to directly measure rainfall and requires a conversion from reflectivity into rainfall which in itself is not perfect (Catchlove et al., 2005).

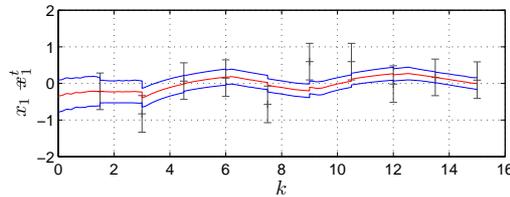
Our model uses data assimilation which combines observational data and numerical models to produce an analysis which is considered the best estimate of the current state of a dynamical system (Giannakopoulou, 2008). Our flood model is just using basic rainfall inputs. This section investigates the effects of wrong rainfall input and how well the data assimilation can compensate for this error.

The error in measured rainfall is accounted for by the imperfect observations in our model. The forecast rainfall error can take different forms: location error, magnitude error and timing error.

- Location error is when the rainfall is predicted in a different location to where the model is run. The model used in this project is 1-D and so error in forecast location is not relevant.
- Magnitude error refers to errors in the amount of rain forecasted. To test this, the ensemble members will be given a ‘false’ rainfall which differs slightly from the true rainfall. This will be created by adding small perturbations to a known rainfall.
- Timing error refers to uncertainty in the time when an event starts, finishes or how long it lasts. To test this, the ‘false’ rainfall will have will have factors which modify its start, end and frequency.



(a) True regular rainfall (red), incorrect rainfall (blue) (b) A case using the true rainfall and ten observations. Colours and symbols as in Fig. 11.



(c) [A case using the incorrect rainfall input and ten observations. Colours and symbols as in Fig. 11.

Figure 15: Truth and incorrect regular rainfall.

We now consider how well the state estimation scheme compensates for incorrect rainfall inputs.

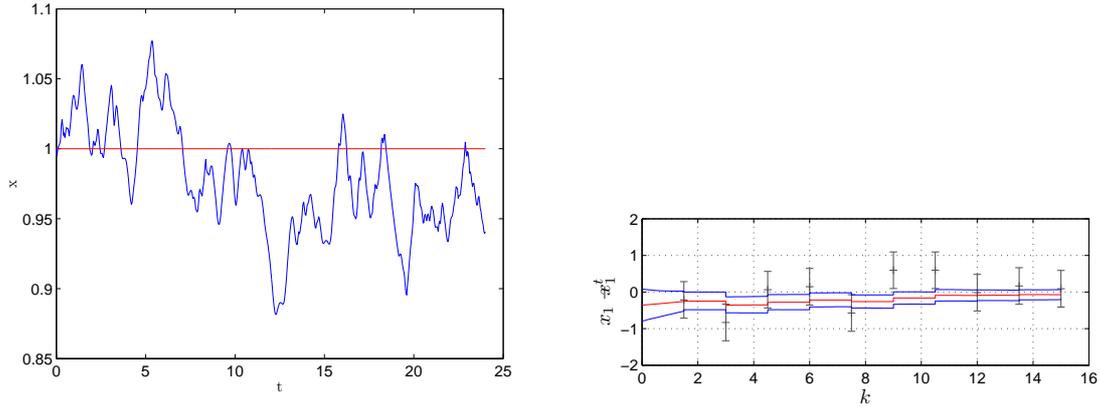
9.1 Rainfall magnitude uncertainty

9.1.1 Regular rainfall

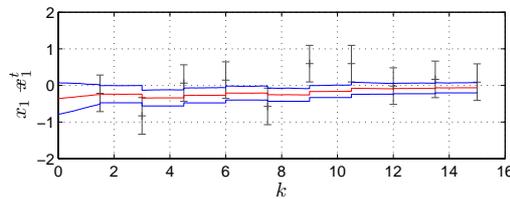
Small perturbations are added to regular rainfall to create a ‘false’ rainfall. In the period with no rainfall, the incorrect input also has no rainfall. The ensemble members are driven by inputs using either the ‘true’ or ‘false’ rainfall, but both use imperfect observations of the ‘true’ rainfall. The plots show the difference between the filter and the truth as in section 8. Figure 15 shows a set of results for the incorrect regular rainfall test.

Figure 15(a) shows the numerical solutions resulting from the perturbations used in this test. The solution with incorrect rainfall oscillates around the solution with true rainfall when it is raining and evolves parallel to it during the intervals with no rain.

Figures 15(b) and 15(c) compare the filter behaviour using the true and incorrect rainfall as inputs. After the first observation, the graphs are extremely similar. It is hard to tell in this experiment whether the similarity of behaviour is due to good filter performance, or simply due to the nature of the model dynamics, which ensure that the solutions rapidly decay during no-rain periods. Continuous rainfall does not have this decay to zero, so the next step is to consider that type of rainfall and see if the graphs match up in the same way.



(a) True continuous rainfall (red), incorrect rainfall (blue) (b) A case using the true continuous rainfall and ten observations. Colours and symbols as in Fig. 11.



(c) A case using the incorrect continuous rainfall and ten observations. Colours and symbols as in Fig. 11.

Figure 16: Continuous Rainfall and Incorrect continuous rainfall with errors up to 20% of the true value.

9.1.2 Continuous rainfall

In this example we use a continuous rainfall for the ‘true’ rainfall inputs where $r = 1$ over the whole interval and perturbations are added to create a ‘false’ rainfall. Much like the random rainfall, added perturbations are generated by a string of random numbers and 0.1 was added or subtracted from the previous rainfall when the random numbers were in the correct range (see section 2.1 for these ranges).

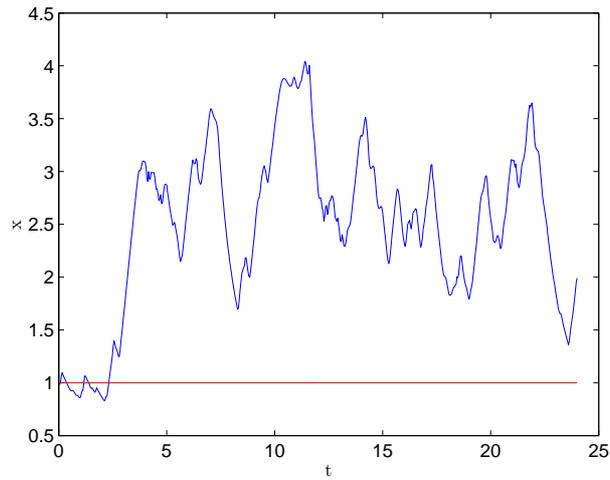
The example in figure 16 shows similar results to the example used in the regular rainfall scenario, although this time there are no drought periods. Again the graphs of the ensemble with the incorrect rainfall and the ensemble with the correct rainfall are similar. The truth is nearly always inside the ensemble mean \pm ensemble standard deviation range. So again the data assimilation scheme has made up for an incorrect rainfall input using observations of the truth.

So far the errors in the measured have been realistic, within 20% of the true value. Figure 17 shows the results when the errors are up to 400%. The spread of ensemble members with the incorrect random rainfall remains nearly constant and the truth is often included within the ensemble mean \pm standard deviation uncertainty band. We can conclude that the state estimation scheme compensates well for magnitude errors in rainfall inputs, even when these are rather large.

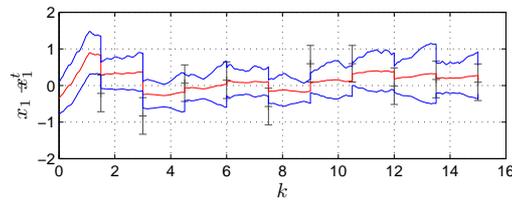
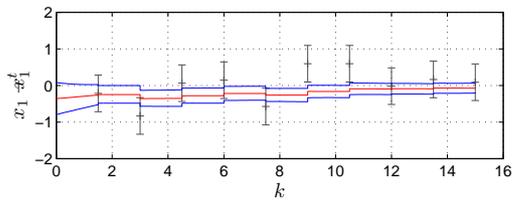
9.2 Rainfall timing uncertainty

9.2.1 Uncertainty in the rainfall event start time

For these experiments we used regular rainfall for the truth. A ‘false’ rainfall was created by adding a factor to make the ‘drought’ step happen before or after the same step in the true rainfall.



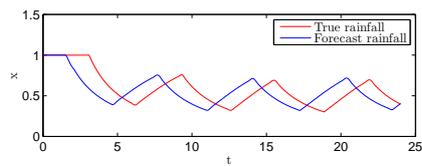
(a) True continuous rainfall in red, blue shows the incorrect rainfall



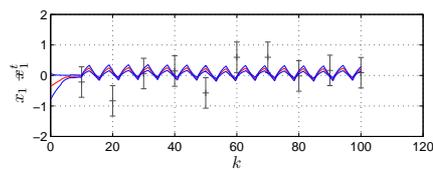
(b) A case using the true continuous rainfall and ten observations. Colours and symbols as in Fig. 11.

(c) A case using the incorrect continuous rainfall and ten observations. Colours and symbols as in Fig. 11.

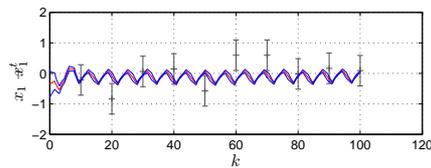
Figure 17: Continuous Rainfall and incorrect rainfall with errors up to 400% of the true value



(a) Truth and early start rainfall scenarios

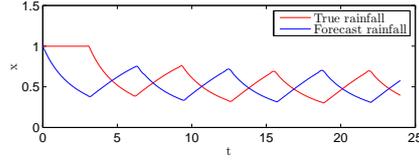


(b) A case using the true regular rainfall and ten observations. Colours and symbols as in Fig. 11.

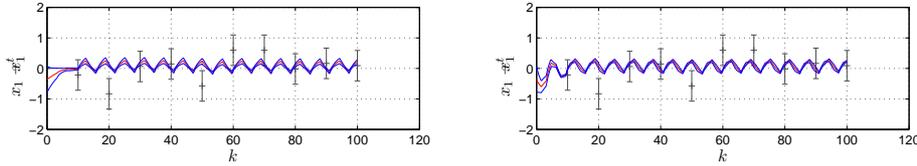


(c) A case using the early start false rainfall and ten observations. Colours and symbols as in Fig. 11.

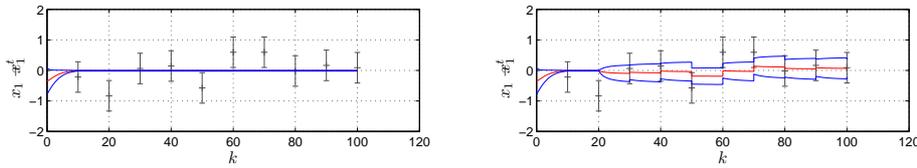
Figure 18: Comparison between true regular rainfall and incorrect rainfall with a phase shift of $\frac{\pi}{2}$ time units early.



(a) Truth and early start rainfall scenarios



(b) A case using the true regular rainfall and ten observations. Colours and symbols as in Fig. 11. (c) A case using the early start false rainfall and ten observations. Colours and symbols as in Fig. 11.

Figure 19: Comparison between true regular rainfall and incorrect rainfall with a phase shift of π time units early

(a) A case driven by the true continuous rainfall. Colours and symbols as in Fig. 11. (b) A case driven by the incorrect rainfall. Colours and symbols as in Fig. 11.

Figure 20: Forecast rainfall stops after 20 time units, true rainfall continues

Figure 18 show results from the scenario when the incorrect rainfall drought phase starts $\frac{\pi}{2}$ units of time early. In figure 18(c), the ensemble members are driven by the incorrect rainfall. It can be seen in this figure that the first observation adjusts the plot back to almost the same phase oscillation as seen in figure 18(b), which depicts the ensemble mean and standard deviation when the ensemble members are driven by the true rainfall. The oscillation in 18(c) is negatively biased relative to the truth, whereas the filter using the true rainfall inputs (figure 18(b)) is positively biased. Since in both cases the standard derivation is very small, the band of the ensemble mean \pm the ensemble standard deviation is close to the ensemble mean.

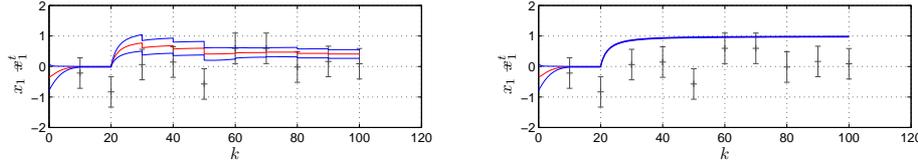
Figure 19 shows when the incorrect rainfall starts π units of time early. The results are similar to the previous experiment, the first observations fixes the oscillation on track. The difference in the means this time, after the first interval, is a lot closer, oscillating around 2%.

Further experiments were carried out using other values of phase shift including starting the rainfall later rather than earlier. They produced qualitatively similar results.

9.2.2 Uncertainty in the rainfall event finish time

To simplify this set of experiments, continuous rainfall is used and three different experiments were carried out.

Figure 20 shows results from an experiment when the forecast rain stops after 20 time units while the true rainfall continues. The observations are imperfect observations generated from the true rainfall.



(a) A case driven by the true rainfall. Colours and symbols as in Fig. 11. (b) A case driven by the incorrect rainfall. Colours and symbols as in Fig. 11.

Figure 21: Incorrect rainfall modelled as continuous, true rainfall stops after 20 time units

In figure 20(b), we see that the ensemble increases its spread after the rainfall stops (when the driving rainfall inputs differ from the truth). The truth is still included in the ensemble mean \pm ensemble standard deviation range. The observations appear to limit the spread to about 0.4 either side of the truth and the mean of the ensemble is close to the truth throughout most of the interval. The assimilation of observations from the true solution compensates well for the incorrect inputs. This is especially remarkable since the true rainfall continued so long after the rainfall was predicted to stop.

The next experiment investigates a case where the true rainfall stops after 20 time units, whereas the incorrect rainfall continues. Figure 21(a) shows the ensemble members with the true rainfall inputs. After the rainfall stops, the trajectories of the ensemble members move away from the truth. When the ensemble members are given the incorrect rainfall inputs, (figure 21(b)), the ensemble members still move away from the truth, but unlike the case with the true rainfall, the ensemble spread does not increase. The filter is overconfident and does not allow correction by the observations.

10 Model Uncertainty

Models are a gross simplification of reality (Leahy et al., 2007) since it is not possible to create a model which includes every process and every factor affecting the system. As stated in section 2, the model used in this project is a simplified 1-D model which only considers rainfall input and river flow. Since we are not using real data and we are generating our observations from model runs we may consider our model structure as a perfect representation of reality.

The model parameters also present significant uncertainty. However, these errors tend to decrease with time as more data is available to calibrate the model parameters (Leahy et al., 2007). Model parameters generally have a physical meaning and are determined by the topography of the land as well as soil and land cover (Giannakopoulou, 2008). In section 5.1, the parameters a and b were shown to be functions of k and n , where k is the storage rate coefficient and n is the store exponent.

In the experiments in this section, the ensemble is forecast using parameters a and b which differ slightly from their true values, which have been fixed at $a = 3(\frac{1}{340})^{\frac{1}{3}}$ and $b = \frac{2}{3}$. Errors in parameters a and b will first be considered separately using a percentage error of their true value. Then they will be changed together by changing the value of n , a factor which defines them both (see equation (12)).

Considering parameter a first, figure 22 shows the graphs of regular rainfall with one observation. Figure 22(a) illustrates the ensemble solution with the correct parameter value. When the incorrect a is smaller than its true value the spread of the ensemble stays larger for longer, always including the truth in the ensemble mean \pm standard derivation band. As time goes on the spread decreases and the graphs converge to the true solution. When the incorrect value of a is $\pm 15\%$ of its true value, the graphs look similar to the case as when the true value of a is used on the ensemble. When the incorrect a is larger than the true value of a , the ensemble spread decreases faster, collapsing to a single line. The collapse happens earlier the larger the errors in a used. As a result the truth is longer included in the ensemble standard derivation band, only at rare points in the time line.

Figure 23 illustrates how increasing the frequency of observations adjusts the filters with the wrong

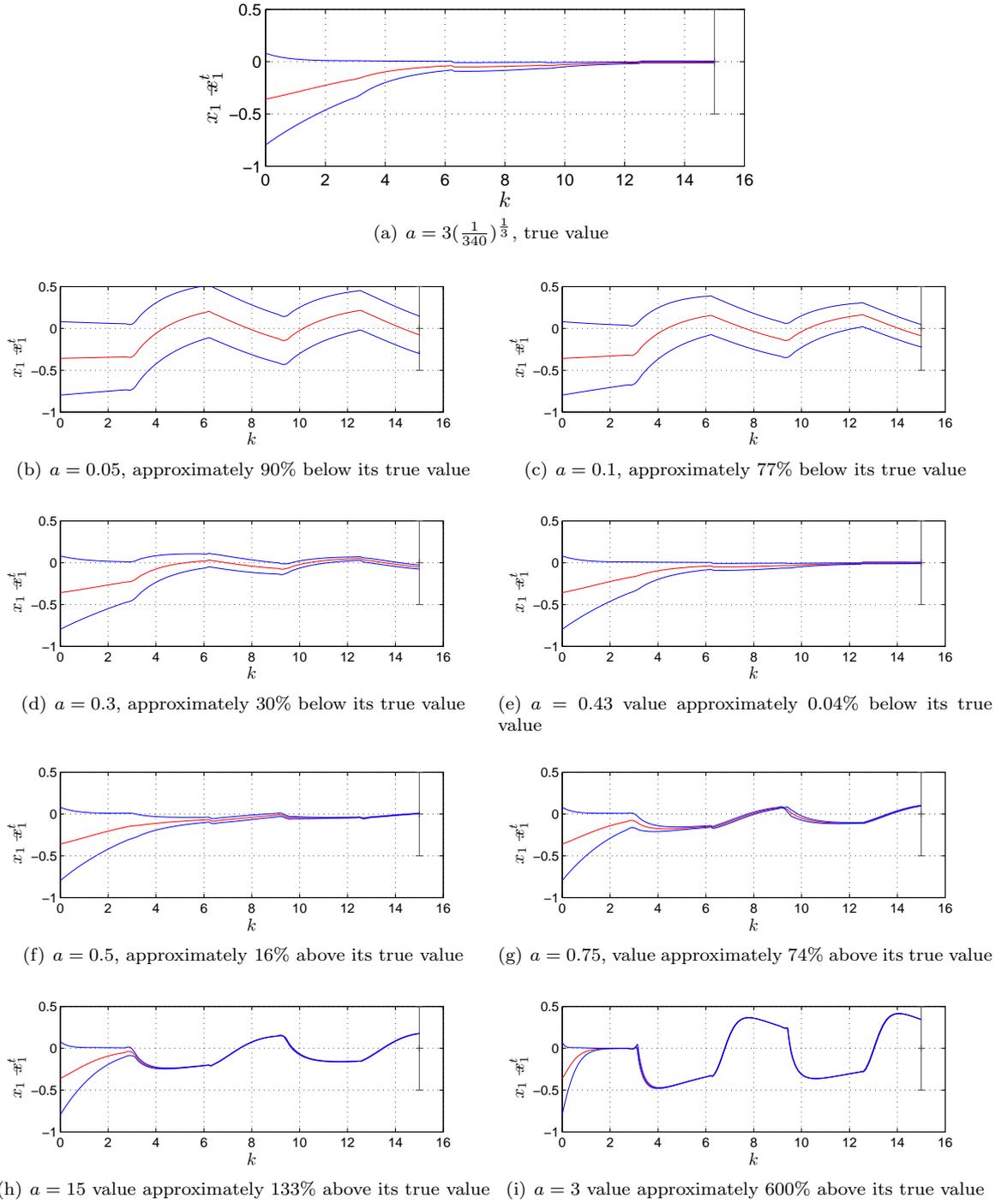


Figure 22: Filter solutions for one observation for a range of values of a . Colours and symbols as in Fig. 11.

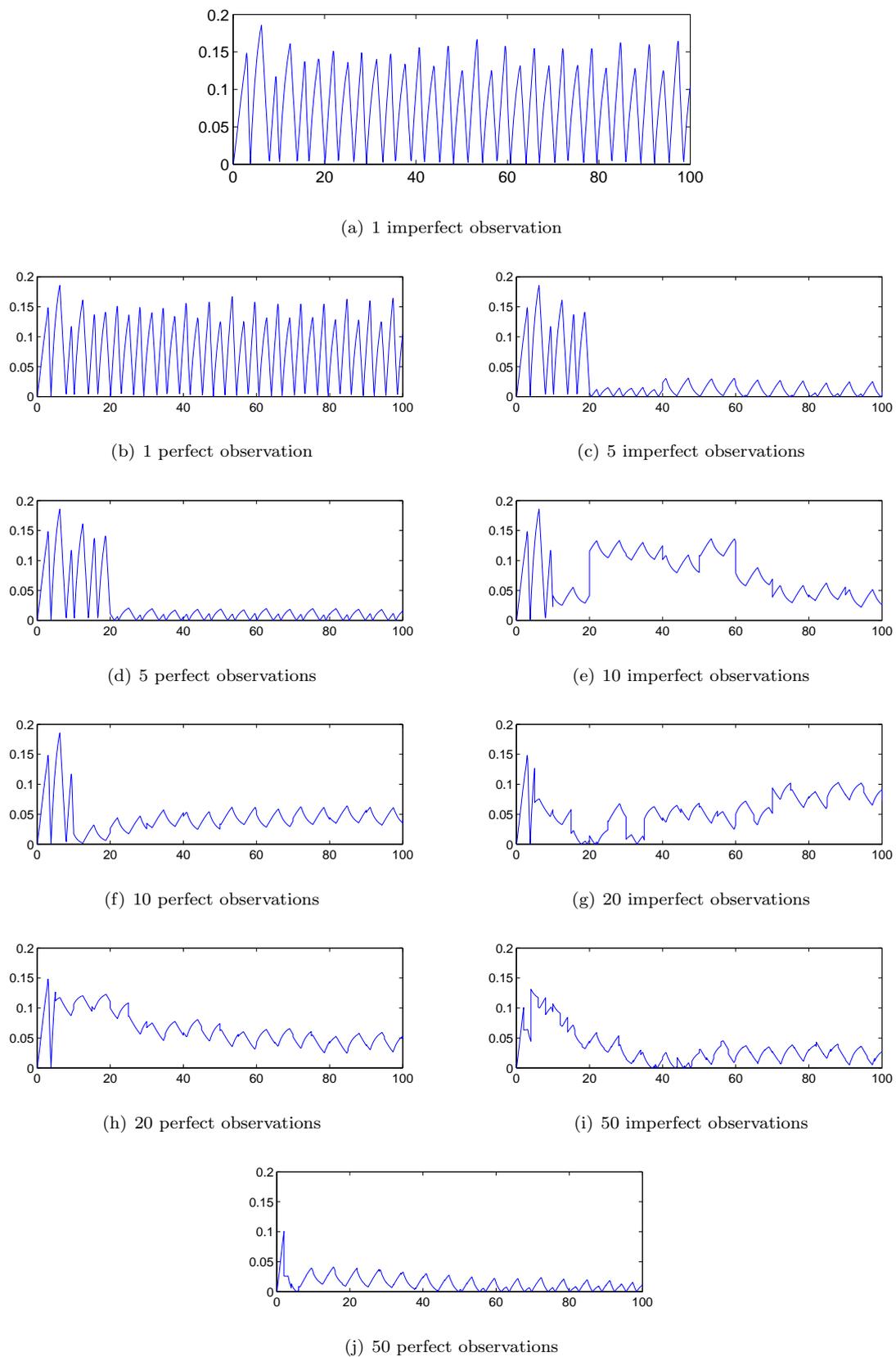


Figure 23: Plots of absolute differences in the mean value of the filter using the true value of the parameter a and the filter taking a as 25% of its true value.

parameter values towards the filter using the true value of a . Fixing the incorrect a to be 25% its true value, both perfect and imperfect observations are used. The absolute difference between the means decreases significantly as the number of observations increases.

Changing parameter b behaves differently from changing parameter a . Smaller values than the truth collapse the filter whereas bigger values spread out the ensemble (not shown). To understand why this is we need to consider equation (1) again.

$$\frac{dx}{dt} = a(r - x)x^b.$$

Parameter a is a constant which largely determines the magnitude of the solution. When the ensemble members are run with a different a value to the true value, their solutions will be of a different magnitude, and so the spread of the solution is larger. The ensemble members are more sensitive to the incorrect values of a than incorrect values of b .

Now let us consider the effects of changing n . Since n is a physical factors, the store exponent, which defines both a and b , it makes sense trying to see how errors in this value affects the results. The other physical factor k , the storage rate coefficient, only defines parameters a , so when looking at errors in a , we also indirectly covered the errors in k .

Looking at in equations in (12) again,

$$a = nk^{1/n}, \quad b = \frac{n-1}{n},$$

we can see that changing n will change a to a greater effect than on b ; the gradient of a with respect to n is steeper than the gradient of b with respect to n . As a result when the ensemble filter was run using parameters a and b calculated using the same errors in n , the results were similar to when the ensemble members were run using an incorrect value of a (not shown).

11 Conclusion

11.1 Summary and Discussion

This project investigated a simple 1-D flood model and applied an Ensemble Transform Kilter Filter (ETKF) to it, to consider the effects of varying the assimilation parameters; ensemble size, observation and background error covariances, and observation frequency. It also considered the main sources of uncertainty in flood models and how the ETKF may compensate for these.

It was found that increasing the number of ensemble members got better results, but only to a certain extent: our model only needs ten ensemble members to obtain its best result. Since our model was a simple 1-D model the state space was of dimension $n = 1$. The observation space dimension m was also equal to one. Thus the ensemble size N was always larger than both m and n . In more complicated models, the observation space and state space dimensions will be much larger than N , so the results might be different. Nevertheless, we would still expect the results to improve as the ensemble size increases.

The effects of changing the observation and background error covariances were seen to linked. Both error covariances worked best in a realistic range of between 1-25% the amplitude of the model solution. The best choice of initial background error covariance was equal to or slightly larger than the observation error covariance value.

The last assimilation parameter considered was the observation frequency. This experiment was less conclusive, however it was seen that more observations often gave better results, and the imperfect (noisy) observations helped to reduce ensemble collapse.

Section 10 considered rainfall input uncertainty and how the filter compensated for these errors. It was found that the state estimation scheme compensated well for magnitude errors in rainfall inputs. Increased observation frequency gave better results. We also considered uncertainty in rainfall timing. The filter compensated well for errors when the rainfall started earlier or later than predicted, with the first observation assimilated largely correcting the rainfall error. The earlier the first observation, the better the ultimate state estimate. However the state estimation scheme compensated poorly when the rainfall event finished before the predicted rainfall.

In section 11, we considered the effect of using the wrong model parameters in the assimilation. Parameters a and b were changed. The state estimation scheme compensates well for underestimations in the value of a and overestimations in the value of b . In the opposite cases the filter does not compensate as well, though works better with incorrect values of b . Underestimations in the value of a causes the spread of the ensemble members to increase, which can compensate for filter divergence. Deliberate use of the wrong parameter a could be considered in order to increase the ensemble spread and give better results could be considered.

11.2 Further Work

The main problem seen with the filtering experiments was ensemble collapse (also known as filter divergence) where the ensemble was overconfident and the ensemble spread was too small. This is a common problem seen in the ensemble filtering literature and it is often corrected by adding an inflation step to the system (Hamill et al., 2001). The next step with these experiments would be to implement such a measure and see how the results in these experiments were affected.

Acknowledgements

This work was funded under the UK Natural Environmental Research Council (NERC) Research Experience Placement (REP) programme.

References

- Bishop, C., Etherton, B., and Majumdar, S., 2001: Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Mon. Wea. Rev.*, **129**, 420–436.
- Catchlove, R., Leahy, C., Seed, A., and T.Malone, 2005: An investigation into the use of real-time radar rainfall and comparison with point gauge estimates for flood forecasting. *29th Hydrology and Water Resources Symposium, Canberra*, 1–6.
- Dormand, J., and Prince, P., 1980: A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, **6**, 19–26.
- Evensen, G., 2003: The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, **53**, 343–367.
- Gelb, A., Ed., 1974: *Applied Optimal Estimation*. The M.I.T. Press.
- Giannakopoulou, E.-M., 2008: *Flood Prediction and Uncertainty*. Master's thesis, University of Reading, Reading, UK.
- Grewal, M., 2001: *Kalman Filtering: theory and practice using MATLAB*. John Wiley and Sons, second edn.
- Hamill, T. M., Whitaker, J. S., and Snyder, C., 2001: Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Mon. Wea. Rev.*, **129**, 2776–2790.

- Leahy, C., Srikanthan, S., G.Amirthanathan, and Sooriyakumaran, S., 2007: Objective assessment and communication of uncertainty in flood warnings. *5th Flood Management Conference Warnamboll, 9 - 12 October 2007*, 1–6.
- Livingston, D., 2005: *Aspects of the Ensemble Kalman Filter*. Master's thesis, University of Reading, Reading, UK.
- Livingston, D., Dance, S., and Nichols, N., 2008: Unbiased ensemble square root filters. *Physica D*, **237(8)**, 1021–1028.
- MathWorks, 2010a: Ode45 documentation. <http://www.mathworks.com/help/techdoc/ref/ode23.html>.
- MathWorks, 2010b: Rand documentation. <http://www.mathworks.com/help/techdoc/ref/rand.html>.
- Moore, R., 2007: The PDM rainfall-runoff model. *Hydrology and Earth System Sciences*, **11(1)**, 483–499.
- Moore, R., and Bell, V., 2002: Incorporation of groundwater losses and well level data in rainfall runoff models illustrated using the PDM. *Hydrology and Earth System Sciences*, **6(1)**, 25–38.
- Moore, R., and Weiss, G., 1980: *Real-Time Forecasting/Control of Water Resource Systems*, chap. Real-Time Parameter Estimation of a Nonlinear Catchment Model using Extended Kalman Filters, 83–92. Pergamon Press.
- Petrie, R. E., and Dance, S., 2010: Ensemble-based data assimilation and the localisation problem. *Weather*, **65(3)**, 65–69.
- Thacker, N., and Lacey, A., 1998: Tutorial: The kalman filter. Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester.
- The Environment Agency, 2010: Flood. <http://www.environment-agency.gov.uk/homeandleisure/floods/default.aspx>.
- The Pitt Review, 2008: *Learning lessons from the 2007 floods*. HMSO, UK.
- Vos, F., Rodriguez, J., Below, R., and Guha-Sapir, D., 2010: *Annual Disaster Statistical Review 2009: The Numbers and Trends*. Centre for Research on the Epidemiology of Disasters (CRED), Brussels, Available from http://www.cred.be/sites/default/files/ADSR_2009.pdf.
- Welch, G., and Bishop, G., 2006: An introduction to the Kalman filter. Tech. Rep. 95-041, Department of Computer Science, University of North Carolina, Chapel Hill, NC 27599-3175.