**Title**
Forecasting high-dimensional state-spaces in the presence of model error



**Author**
Tran, Linda Ngoc

**Publication Date**
2016

Peer reviewed|Thesis/dissertation

**Forecasting high-dimensional state-spaces in the presence of model error**

by

Linda Ngoc Tran

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Assistant Professor Cari G. Kaufman, Co-chair
Professor Deborah A. Nolan, Co-chair
Professor Michael I. Jordan
Professor Pieter J. Abbeel

Spring 2016

**Forecasting high-dimensional state-spaces in the presence of model error**

**Abstract**

Forecasting high-dimensional state-spaces
in the presence of model error

by

Linda Ngoc Tran

Doctor of Philosophy in Statistics

University of California, Berkeley

Assistant Professor Cari G. Kaufman, Co-chair

Professor Deborah A. Nolan, Co-chair

Mathematical models are often used to forecast interesting scientific phenomena. These models may provide a good approximation for the system being studied, but they are rarely perfect. Whether due to scientific misunderstandings or the necessity of numerically approximating the solution to the model, the mathematical models fail to exactly replicate the true dynamics of the system. This "model error" leads to forecast errors and degradation of forecast skill. The primary goal of this dissertation is to develop methods to correct a mathematical model with observed data, improving the accuracy of forecasts generated by the model. Correcting for model error is particularly difficult when the system is high-dimensional and when the state transition model is nonlinear. The methods developed in this dissertation are robust to both of these difficulties.

The problem of estimating model error may be phrased as a parameter estimation problem within a filtering context. Many parameter estimation methods have been developed for the particle filter (PF), a filter with strong theoretical underpinnings and well understood large sample properties. Unfortunately, the PF fails when the state dimension is large. Instead, scientists who filter high-dimensional state-spaces typically use the ensemble Kalman filter (EnKF). This algorithm is effective in practice, but its large sample properties remain to be proven and methods for parameter estimation exist only in special cases. We make a newfound connection between these two algorithms: the EnKF is a regularized PF with additional approximations that have yet to be fully justified from a theoretical standpoint. Building on this connection, we introduce a novel algorithm that combines the best features of the EnKF

and the PF. The EnKF is used to filter the high-dimensional state while an auxiliary PF is used to estimate the low-dimensional parameters; we call this filter the EnKF-APF. Using the Lorenz 2005 system, we demonstrate that the true parameter values are better captured with the EnKF-APF algorithm than with the EnKF.

Having developed a robust method for parameter estimation within a filtering context, we return to the question of model error. Following previous work in the literature, we focus on a linear correction to the mathematical model. If the state has dimension $d$, the number of parameters in the linear correction is on the order of $d^2$. The EnKF-APF algorithm does not allow for the estimation of such high-dimensional parameters, and even if it did, an enormous amount of data would be required to provide good estimates. To make the problem tractable, we assume the problem of interest exhibits a large degree of spatial structure and exploit this special structure to reduce the dimensionality of the linear correction. We propose to correct the model error with a low-rank linear correction, inspired by methods used to model multivariate spatial processes in the geostatistics literature. We demonstrate our method on the Lorenz 2005 system by applying the EnKF-APF algorithm to estimate the parameters of the low-rank linear correction in both a batch and online manner. Although the model error in this test is nonlinear, we demonstrate that the proposed low-rank linear correction provides better forecasts.

To my husband David,
for there wouldn't be a dissertation
without his love and support.

# Contents

# List of Algorithms

# List of Figures

# Acknowledgements

First and foremost, I thank my advisor Cari Kaufman for her support and mentorship throughout my graduate career. I am extremely grateful to Deb Nolan for stepping in during her sabbatical to provide support in finishing my dissertation. Many ideas in this dissertation were inspired by discussions and comments from Doug Nychka, Jeff Anderson, and Josh Hacker at the National Center for Atmospheric Research (NCAR); I thank them for their hospitality and generosity during my visits. I thank both of my committee members, Mike Jordan and Pieter Abbeel, for their helpful comments and connecting me to parts of the literature related to my research.

I thank my fellow graduate students, who have fondly become both my friends and colleagues, for their technical and emotional support. I am especially grateful for the support from my academic siblings, Yu-Jay Huoh and Wayne Tai Lee. I thank the administrative staff for keeping everything running so smoothly that I rarely noticed anything was amiss, especially Chris Paciorek and Ryan Lovett for systems support and La Shana Porlaris for everything related to being a grad student.

I thank my family and friends for their love and support. Savet Hong always seemed to know what I needed before I knew that I needed it, especially during the roughest parts of grad school. I especially thank my husband, David Stein: I wouldn't have finished this dissertation without his love and support. Not only was he my rock, he helped me understand the nuances behind the mathematical models and numerical integration methods used in my demonstrations.

Lastly, I am indebted to the taxpayers of America for generously funding my graduate career with two fellowships: the UC Berkeley Chancellor's Fellowship and the National Science Foundation (NSF) Graduate Research Fellowship. I promise to become a productive member of society and pay you back in the form of taxes. In the same vein, I am obligated to acknowledge

my NSF fellowship officially. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1106400. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

# 1 Introduction

Mathematical models are often used to forecast interesting scientific phenomena. While these models are good approximations of the systems being studied, they sometimes miss dynamics of the system—a problem we call model error—and consequently contribute to forecast error. The main goal in this dissertation is to use measurements of the scientific phenomena to correct model error. To better motivate the issues, we introduce state-space models.

## 1.1 State-space models: an introduction

Suppose we want to learn about the *state* $X_t$ at time $t$. Here, we use the standard convention of denoting random variables with upper-case letters and fixed values with lower-case letters. We have that the state $X_t$ evolves in time as follows:

$$x_t = l_t(x_{t-1}, \eta_t) \text{ with density } f(x_t \,|\, x_{t-1}). \tag{1.1a}$$

This evolution of the system is called the *state transition model*. The function $l_t : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$ describes how the state $x_t$ of the system propagates forward in time: it is a function of the previous state $x_{t-1}$ and *state disturbance* $\eta_t$, where $d_x$ is the dimension[1] of the state $x_t$ for all $t$. Typically, the state disturbance $\eta_t$ will be additive[2] with respect to the previous state, e.g., $x_t = m_t(x_{t-1}) + \eta_t$. The form in Equation (1.1a) is used for greater generality, allowing $\eta_t$ be multiplicative with the previous state, for example. The probability density of the state transition model is called the *state transition density* (or transition density, for short) and is denoted by $f(x_t \,|\, x_{t-1})$. The *initial state* $X_0$ is sometimes called the *initial condition(s)*. The states $\{X_t : t = 1, ..., T\}$ are assumed to

[1]The state dimensions can vary with time. Without loss of generality, we assume that the dimensions do not change over time.

[2]For many applications in this dissertation, the state disturbance is zero.

form a *Markov process*: the future state of the system is conditionally independent of the past states given the current state:

$$X_t \perp\!\!\!\perp X_{t'} \,|\, X_{t-1} \text{ for all } t' < t-1.$$

This assumption is called the *Markov assumption*. To help fix these ideas, we present an example.

**Example 1.1.1** (A simple temperature model)**.** Suppose we are interested in forecasting temperature at a particular location. A simple state transition model is to assume that the current day's temperature is a function of the previous day's temperature:

$$x_t = x_{t-1} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, u_t^2),$$

where $\eta_t$ is a random variable representing our theory that temperature increases or decreases relative to the temperature from the previous day $x_{t-1}$ with a certain magnitude (its distribution is called a *prior*). Here $\mathcal{N}(\mu, \sigma^2)$ denotes a normal random variable with mean $\mu$ and variance $\sigma^2$. The normal prior says that we believe that the temperature can increase or decrease with equal probability (indicated by the mean zero) and with an increase or decrease by an amount less than $2u_t$ with 95% probability (indicated by the magnitude of the variance). Notice that our model satisfies the Markov assumption: the current day's temperature only depends on the previous day's temperature and is not a function of other days' temperatures. The transition density is:

$$f(x_t \,|\, x_{t-1}) = \phi(x_t; x_{t-1}, u_t^2),$$

where $\phi(x; \mu, \sigma^2)$ is the probability density of a normal random variable, with mean $\mu$ and variance $\sigma^2$, evaluated at $x$. ◁

Usually, the state cannot be measured directly[3] and, even if it can be, the measurement has error. Let $Y_t$ be the *measurement* at time $t$ with the following model:

$$y_t = o_t(x_t, \epsilon_t) \text{ with density } g(y_t \,|\, x_t). \tag{1.1b}$$

This is called the *measurement model*. The function $o_t : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ describes the mapping between the state and what is observed: it is a function of the

[3]For this reason, the state is sometimes called a *latent* or *hidden variable*.

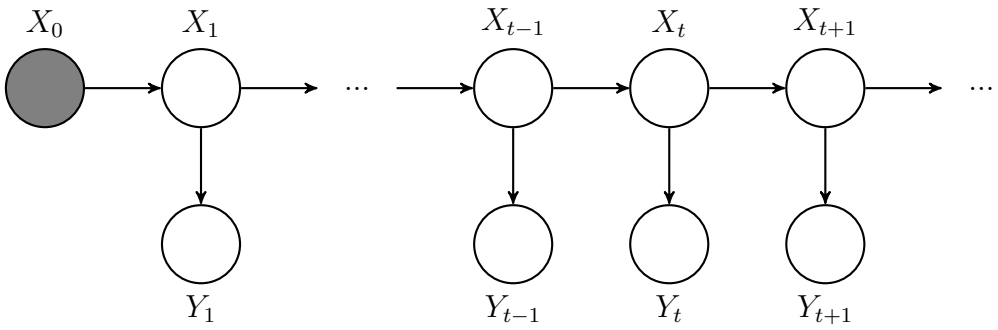



Figure 1.1: *State-space model with no measurements collected: a graphical model.* The nodes $\{X_t : t \geq 1\}$ represent the states. The node $X_0$ is the initial condition and is filled in to represent that it is observed. The nodes $\{Y_t : t \geq 1\}$ represent the measurements and are not filled in because they have not yet been collected.

current state $x_t$ and *measurement error* $\epsilon_t$, where $d_y$ is the dimension[4] of the measurement $y_t$ for all $t$. The probability density of the measurement model is called the *measurement density* and denoted by $g(y_t \,|\, x_t)$. The measurements depend only on the state, i.e., the measurements are conditionally independent of everything else given the state:

$$Y_t \perp\!\!\!\perp (X_{t'}, Y_{t'}) \,|\, X_t \text{ for all } t' \neq t;$$

we call this the *measurement independence assumption*. The set of equations in Equation (1.1) forms a *state-space model*. Figure 1.1 is a graphical model that describes the conditional independence relationships in the state-space model we've just outlined.

**Example 1.1.1** (A simple temperature model, continued)**.** The temperature can be measured with a thermometer but not without measurement error:

$$y_t = x_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, v_t^2),$$

where $v_t^2$ quantifies the accuracy of the thermometer. Then, the observation density is:

$$g(y_t \,|\, x_t) = \phi(y_t; x_t, v_t^2).$$

◁

[4]As with the state, the dimensions of the measurements can vary in time. Without loss of generality, we assume that the dimensions do not change over time.

**Example 1.1.2** (A simple precipitation model)**.** Consider modeling a slightly more complicated phenomenon: precipitation. Like temperature, a simple model of precipitation is that today's precipitation depends on yesterday's precipitation. However, unlike temperature, precipitation cannot take any value on the real line. Two measurements of precipitation are total inches rained rounded to the nearest tenth:

$$y_t \sim \text{Pois}(\lambda_t),$$

or more simply as whether or not it rained:

$$y_t \sim \text{Bern}(p_t),$$

where $\text{Pois}(\lambda)$ is a Poisson random variable with mean $\lambda$ and $\text{Bern}(p)$ is a Bernoulli random variable with mean $p$. These are measurement models and, unfortunately, do not describe how precipitation evolves over time.

A latent process can be used to describe an abstract notion of precipitation—a notion that is measured on a continuous scale and can be specified to be temporally correlated. For example, a latent process for total inches rained is

$$\log(\lambda_t) = \log(\lambda_{t-1}) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, u_t^2),$$

and for whether or not it rained is

$$\log\left(\frac{p_t}{1-p_t}\right) = \log\left(\frac{p_{t-1}}{1-p_{t-1}}\right) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, u_t^2),$$

where $\eta_t$ is a random variable that describes our theory that the latent state of precipitation increases or decreases relative to the previous latent state, similar to the simple temperature model of Example 1.1.1.

Let's define the state-space model. For total inches rained, let $x_t \equiv \log(\lambda_t)$. Then, the state-space model is defined as

$$\begin{aligned} x_t &= x_{t-1} + \eta_t, \\ y_t &\sim \text{Pois}(\exp(x_t)), \end{aligned}$$

with state transition and observation densities

$$\begin{aligned} f(x_t \,|\, x_{t-1}) &= \phi(x_t; x_{t-1}, u_t^2), \\ g(y_t \,|\, x_t) &= \frac{\exp(x_t y_t) \exp\exp(x_t)}{y_t!}, \end{aligned}$$

respectively, where $x!$ is the factorial of $x$. The state-space model for whether or not it rained can be similarly defined. ◁

Though the temperature and precipitation models discussed are simple and easy to understand, it is difficult to believe that weather dynamics are simply an unbiased random walk. Weather is a complex phenomenon that depends on many factors, such as the amount of sun that reaches the atmosphere or the amount of aerosols and carbon dioxides in the system, and often these relationships are nonlinear. Furthermore, these simple models cannot be used to describe global, or even local, dynamics: weather is temporally *and* spatially related, e.g., the temperature at one location is highly correlated with the temperature at nearby locations.

Scientists and engineers from a wide variety of fields, including but not limited to the atmospheric sciences, often describe and test their understanding of scientific phenomena with mathematical models. These mathematical models describe how scientific phenomena (the state) evolve over time and, in turn, can be used to predict (or, *forecast*) the state at a future timepoint. Even though many mathematical models are deterministic, these deterministic models may not be predictable—a quality called *chaos*. A system is called chaotic when two trajectories that started infinitesimally close to each other diverge quickly over a short amount of time.

The consequence of chaos is that these mathematical models cannot forecast phenomenon far in the future. When initializing mathematical models with two different, but close, initial conditions that represent reality, chaos will drive the two trajectories to two very different states. There needs to be a mechanism to correct the trajectories predicted by the mathematical models with reality and that mechanism is to *update* these imperfect forecasts with measurements. Measurements quantify the scientific phenomena that scientists or engineers are interested in capturing from reality, such as measuring temperature using thermometers or measuring rainfall with a ruler. Not only are these measurements subject to measurement error, it is impossible to measure desired quantities at every spatial location and timepoint that is desired. For example, the National Oceanic and Atmospheric Administration (NOAA) ingests meteorological data at approximately 36,000 locations in the contiguous United States (Meteorological Assimilation Data Ingest System (MADIS), 2014)—that is, each station covers more than 80 square miles if the stations were evenly distributed[5]! This iterative procedure of *forecasting* with a mathematical model and *updating* as an observation is collected is called *filtering* (or, *data assimilation* within the atmospheric science community).

The Kalman filter (Bucy and Kalman, 1961; Kalman, 1960) is the first

[5]Value derived from dividing the area of the contiguous United States by the number of stations.

major contribution to the filtering literature: it is used for linear state transitions and measurement models with additive Gaussian errors. Since then, many filtering algorithms have been introduced to solve a wide variety of state-space models. In particular, to overcome some of the strict assumptions of linearity and Gaussian errors, linear approximations have been proposed to extend the usefulness of the Kalman filter. However, these approximations can introduce large errors that further contribute to the chaos problem when filtering. Thanks to the pioneering work of Gordon et al. (1993) on particle filters (PFs), the field of sequential Monte Carlo (SMC) was developed in response: these methods employ importance sampling to approximate interested quantities of the states. Though there are many filtering methods to choose from, filtering remains difficult for high-dimensional systems, such as the weather models that atmospheric scientists are interested in studying. For example, a computationally infeasible amount of samples is required by SMC methods to properly capture the dynamics of the system.

Not only is filtering with high-dimensional state-spaces difficult, we are seldom able to perfectly capture the true dynamics of a system, especially complex systems such as the ones governing the atmosphere. We call this inability to capture the true dynamics of a system *model error*. Sometimes, model error is due to imperfect knowledge of the system. Other times, it comes from introducing approximations that are not necessarily believed to be true but are required for computational reasons. Model error is a common problem that can further contribute to the issues posed by chaos. For example, the inability to capture certain dynamics of a system can lead to two very different forecasts. In fact, model error is the problem we initially sought to improve that led to our other contributions to the filtering literature that are described in this dissertation.

Before discussing filtering and model error, we introduce a few state-space models. Although the algorithms discussed and developed in this dissertation can be applied to any model that satisfy the constraints and assumptions of the state-space models outlined, we focus on atmospheric science applications. For this reason, we introduce state transition and measurement models often used in the atmospheric science literature to motivate the filtering algorithms introduced in this dissertation. As we discuss these models, we motivate some of the difficulties of filtering with atmospheric models. In Section 1.4, we summarize these difficulties and introduce how we address them in the dissertation.

## 1.2 State transition models

### 1.2.1 Examples

#### Lorenz 1963

Edward Lorenz is widely known across many disciplines as one of the pioneers of chaos theory. His most well-known contribution is the development of a simple chaotic model (Lorenz, 1963) often used in textbooks across many disciplines to illustrate and gain intuition for chaos. The model was derived from a more complicated model of convection, a scientific phenomenon describing heat flow. Many atmospheric models include convection, such as those that describe air or ocean circulation.

The Lorenz model is a set of ordinary differential equations that describes the rate of change of the state of the system $(x(t), y(t), z(t))$ at any continuous timepoint $t$:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} \sigma[y(t) - x(t)] \\ x(t)[\rho - z(t)] - y(t) \\ x(t)y(t) - \beta z(t) \end{bmatrix} \tag{1.2}$$

where the dot on top of $x$ represents the first-order time-derivative, e.g., $\dot{x}(t) \equiv \frac{dx(t)}{dt}$ and the notation $x(t)$ is to distinguish between the continuous (with $t$ in parenthesis) and discrete (with $t$ as a subscript) states of the system, explained later in Section 1.2.2. The parameters of the system are $\rho, \sigma$, and $\beta$ and their default values are:

$$\rho = 28, \sigma = 10, \text{ and } \beta = 8/3.$$

The system is illustrated in Figures 1.2 to 1.4. Figure 1.5 plots the same trajectory from Figure 1.4 and another trajectory with the same parameters and very similar initial conditions: the initial condition of the new trajectory is the same as the other's initial conditions, but rounded to the 10th digit. Even though the initial conditions are similar, notice how the system diverges over time—this is chaos. It is an example of a very realistic problem of round-off error: two very similar initial conditions that differ only by round-off error lead to two very different predictions, illustrating the need to update the forecasts with measurements for the system to better represent reality.

Though the model is not realistic, we, as many others, like this model for a few reasons. The model is simple and can quickly run on any computer, yet has interesting dynamics. Furthermore, the three-dimensional nature of

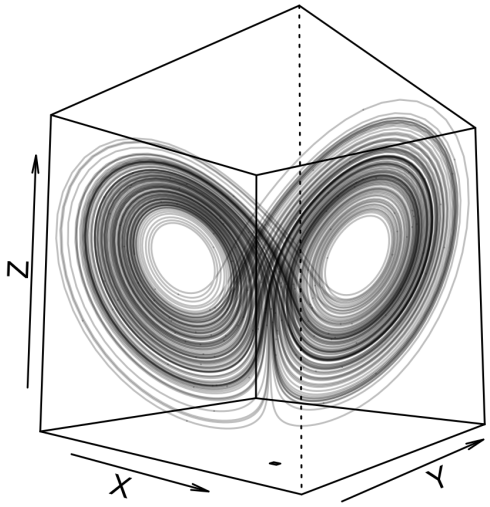



Figure 1.2: *Lorenz 1963: three-dimensional illustration.* Trajectory is generated by numerically integrating Equation (1.2) with the default parameter values.

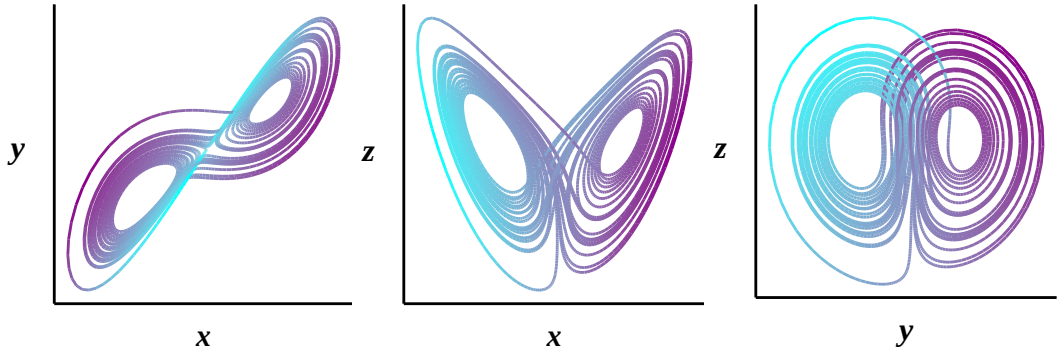



Figure 1.3: *Lorenz 1963: two-dimensional illustrations.* The trajectory is generated by numerically integrating Equation (1.2) with the default parameter values. The colors represent the missing axis: blue indicates lower values and purple indicates higher values.

the model allows for simpler visualization to motivate and explain some of the intuition behind the methods we introduce.

After the introduction of Lorenz (1963), Lorenz developed a few more models. (After introducing Lorenz's model for the first time, we refer to each of his models by his surname followed by the year of the paper, e.g., Lorenz 1963 model.) Like the Lorenz 1963 model, Lorenz (1984) models a three-dimensional state space that reflects dynamics that are present in global atmospheric circulation models (called Hadley circulation). Lorenz then developed models to more realistically reflect an atmospheric model. The first was introduced in Lorenz (1996) (and evaluated further in Lorenz and Emanuel (1998)) and two more models in Lorenz (2005).

Both the Lorenz 1996 and 2005 models are systems of equations that reflect weather dynamics, such as temperature or vorticity, at locations on an equally-spaced mathematical grid on a latitude band of the globe, where the latitude band is assumed to be perfectly spherical. Each equation in the system describes the rate of change of the state at one location on the mathematical grid. Figure 1.6 illustrates a mathematical grid on a latitude band. *Resolution* describes the number of locations on the grid: higher resolution means more locations. Since its introduction, the Lorenz 1996 model is widely used in the atmospheric sciences to demonstrate filtering methods.

### Lorenz 2005

Recognizing that weather models are subject to model error, Lorenz (2005) specifically developed two models (Models II and III)—both inspired by the Lorenz 1996 model[6]—for scientists to test their ideas and methods on model error. Model II captures the large-scale dynamics of Model III, but omits the small-scale dynamics.

Define a new summation operator:

$$\sum_{i=1}^{n}{}' x_i = \frac{x_1}{2} + \sum_{i=2}^{n-1} x_i + \frac{x_n}{2}$$

[6]Model I in Lorenz (2005) is, in fact, the Lorenz 1996 model.

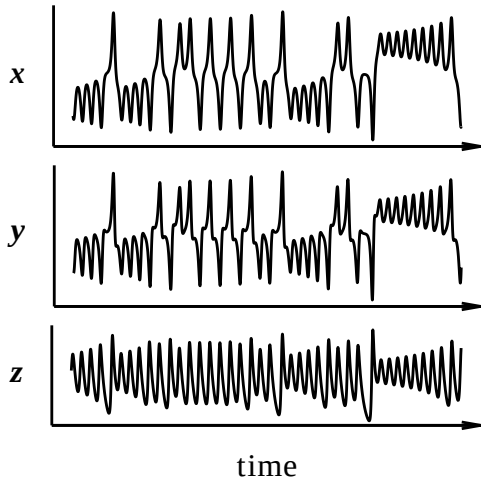



Figure 1.4: *Lorenz 1963: one-dimensional illustrations.* The trajectory is generated by numerically integrating Equation (1.2) with the default parameter values.

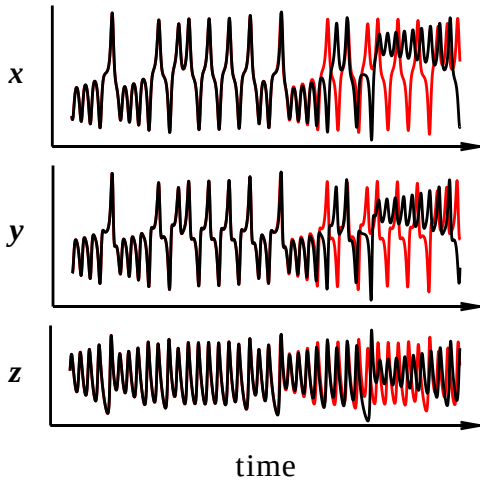



Figure 1.5: *Lorenz 1963: an example of chaos.* The black trajectory is the same as in Figure 1.4. The red trajectory has the same parameter values as the black trajectory but with different initial conditions: the red's initial condition is the black's initial condition rounded to the 10th digit.

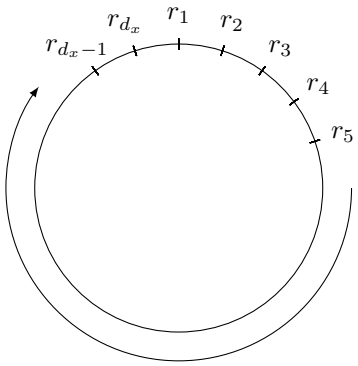



Figure 1.6: *Mathematical grid on a latitude band.* The circle represents a top-down view of a latitude band. Let $r_i$ be the location of the state corresponding to the $i$th element of the $d_x$-vector $x_n$. The locations $\{r_i : 1 \leq i \leq d_x\}$ in this illustration are equispaced on the latitude band.

and the Lorenz inner product function[7]:

$$[x,y]_{K,n} = \begin{cases} \frac{1}{K^2} \sum_{j=-\frac{K}{2}}^{\frac{K}{2}} \sum_{i=-\frac{K}{2}}^{\frac{K}{2}} \left(-x_{n-2K-i}y_{n-K-j} + x_{n-K+j-i}y_{n+K+j}\right), & \text{if } K \text{ is even,} \\ \frac{1}{K^2} \sum_{j=-\frac{K-1}{2}}^{\frac{K-1}{2},} \sum_{i=-\frac{K-1}{2}}^{\frac{K-1}{2},} \left(-x_{n-2K-i}y_{n-K-j} + x_{n-K+j-i}y_{n+K+j}\right), & \text{if } K \text{ is odd.} \end{cases} \tag{1.3}$$

Figure 1.7 illustrates the Lorenz inner product function. Then, the two Lorenz 2005 models are as follows:

- **Model II**: Each component of the state $x(t) \equiv (x_1(t), ..., x_N(t))$ is modeled with an ordinary differential equation:

$$\dot{x}_n(t) = [x(t), x(t)]_{K,n} - x_n(t) + F, \tag{1.4}$$

  with parameters $N, K$, and $F$. The parameter $N$ controls the resolution of the mathematical grid on the latitude band, i.e., the dimensions of the state and the parameter $F$ is a forcing. The parameter $K$ controls the wavenumber of the dynamical system, where a higher value indicates more spatial correlation.

- **Model III**: Each component of the state $z(t) \equiv (z_1(t), ..., z_N(t))$ is modeled with an ordinary differential equation:

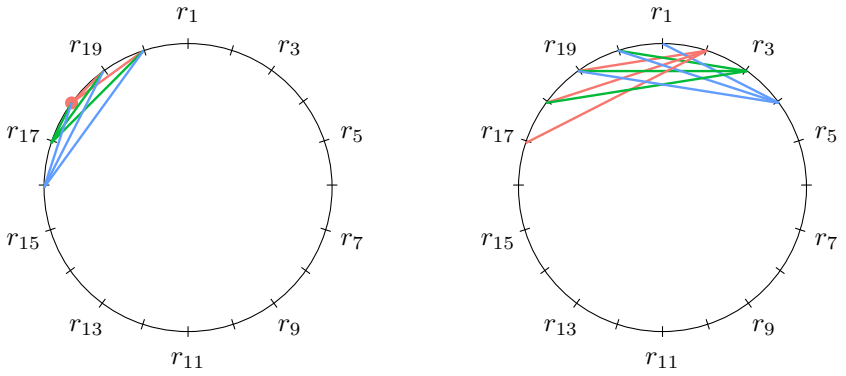
$$\dot{z}_n(t) = [x(t), x(t)]_{K,n} + b^2[y(t), y(t)]_{1,n} + c[y(t), x(t)]_{1,n} - x_n(t) - by_n(t) + F, \tag{1.5a}$$

  with

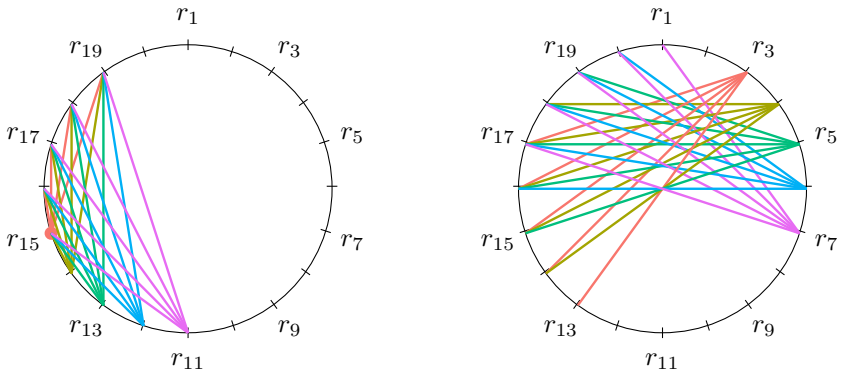
$$x_n(t) = \sum_{i=-I}^{I} (\alpha - \beta|i|) z_{n+i}(t), \qquad y_n(t) = z_n(t) - x_n(t), \tag{1.5b}$$

$$\alpha = \frac{3I^2+3}{2I^3+4I}, \qquad \beta = \frac{2I^2+1}{I^4+2I^2}, \tag{1.5c}$$

[7] In this dissertation, brackets $[\cdot]$ are also used to group mathematical terms, represent matrices/vectors, and indicate the inputs of a function, similar in use to parentheses. This is distinctly different from brackets with two subscripts separated by a comma $[\cdot,\cdot]_{K,n}$, representing the function stated here.




(a) $K = 2$




(b) $K = 4$

Figure 1.7: *Lorenz 2005: Lorenz inner product function.* The circle from Figure 1.6 is used to illustrate the Lorenz inner product in Equation (1.3) with the left and right circles representing the first and second products in the sums. The parameter $n$ is set to 1. The lines represent the product between the $x$ and $y$ located at the indicated spatial locations $r$. When $x$ and $y$ from the same spatial locations $r$ are being multiplied together, the line is replaced by a dot. The colors represent the $x$ and $y$ components in the left and right circles, respectively.

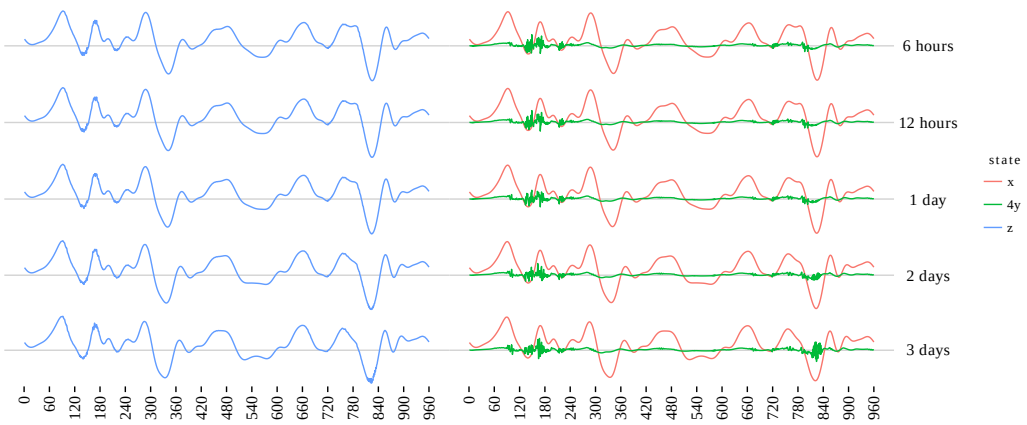



Figure 1.8: *Lorenz 2005-III: large- versus small-scale dynamics.* Model III is numerically integrated with the default parameter values. The left figure illustrates the state $\{z_n : 1 \leq n \leq 960\}$; its largest magnitude is 15. The right figure illustrates the $x_n$ and $y_n$ components of the state $z_n$. The $y_n$ component is multiplied by 4 to better illustrate the small-scale dynamics. The middle line is zero. The $x$-axis is the index of equispaced locations on a latitude band.

and parameters $N, K, F, I, b$, and $c$. The parameters $N, K$, and $F$ have similar effects as in Model II. The $x_n(t)$ and $y_n(t)$ components represent the large- and small-scale dynamics of the system, respectively, and the relationship between them are controlled by the parameters $I, b$, and $c$. Furthermore, notice that Model III reduces to Model II when $I = 1$. Figure 1.8 illustrates the large- versus small-scale dynamics of this system.

In his paper, Lorenz used the following parameter values, where applicable, for both models:

$$N = 960, K = 32, F = 15,$$
$$I = 12, b = 10, c = 2.5;$$

we call these the default parameters for both systems. Figure 1.9 illustrates the effect of the small-scale dynamics on forecasts. In addition to referring to these models as the Lorenz 2005 systems, we refer to the specific models with a hyphenation, i.e., Lorenz 2005-II and Lorenz 2005-III for Models II and III, respectively.

### 1.2.2 Continuous state transition models

In their present forms, the Lorenz models cannot be directly used as a state transition model. We have presented mathematical models on *the rate of change* of the state of the system and have not yet described the model of the

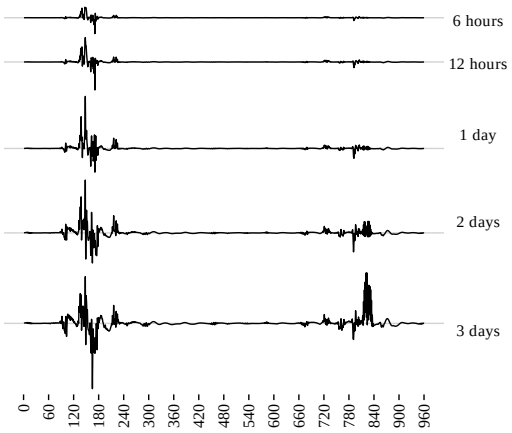



Figure 1.9: *Lorenz 2005: the effect of small-scale dynamics on forecasts.* Both Model II and Model III are numerically integrated with default parameter values and the same initial conditions. Each line represents the difference in forecasts between Model II and Model III after the amount of time indicated to the right of the plot, where the numerical timestep $dt = 0.001$ represents one hour. The $x$-axis is the index of equispaced locations on a latitude band. The gray line represents no difference. Above the gray line, Model II predicts higher values than Model III. The magnitudes are relative to each other with the largest percent difference of 30%.

state of the system when measurements are collected. Although the equations provide models for the system at any continuous timepoint $t$, the reality is that measurements are collected at discrete timepoints $t_n$. To update forecasts with the collected measurements, the state transition model needs to be integrated to a similar temporal resolution as the measurements. To distinguish between continuous and discrete time, we adopt the convention of using parentheses to describe the state that can be evaluated at any continuous timepoint $t$, e.g., $x(t)$, and subscripts to describe the state at a discrete timepoint $t_n$, e.g., $x_{t_n}$.

Let $x(t)$ be the state of the system at the continuous timepoint $t$ and $W$ be the model of the first time-derivative of the state of the system:

$$\dot{x}(t) = W(x(t); \theta)$$

where $\theta$ is the parameter of the model and the dot on top of $x$ represents the first time-derivative, i.e., $\dot{x}(t) \equiv \frac{dx(t)}{dt}$. Because the model describes the system at any continuous timepoint $t$, we call this the *continuous state transition model*. To help fix this idea, we use the Lorenz systems introduced in the last section to present examples of continuous state transition models.

**Example 1.2.1** (Lorenz 1963: continuous state transition model)**.** Lorenz 1963 is a model of the state $(x(t), y(t), z(t))$ and its continuous state transition model $W$ is defined by the right-hand side of Equation (1.2) with the parameter

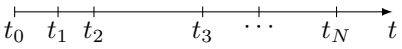



Figure 1.10: *Discrete versus continuous time.* The line represents continuous time. The tick marks at $t_n$ for each $n = 1, ..., N$ represent the timepoints at which measurements are collected with the exception of $t_0$, where $t_0$ represents the timepoint of the initial condition. In this illustration, there are $N$ discrete timepoints.

$\theta = (\rho, \sigma, \beta)$, i.e.,

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = W\left( \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}; \theta \right) = \begin{bmatrix} \sigma[y(t) - x(t)] \\ x(t)[\rho - z(t)] - y(t) \\ x(t)y(t) - \beta z(t) \end{bmatrix}. \tag{1.6}$$

◁

**Example 1.2.2** (Lorenz 2005: continuous state transition model)**.** Let $d_x$ be the number of states on the mathematical grid, corresponding to the parameter $N$ in Lorenz's original formulation of the model. With Lorenz 2005-II, the state $x(t) \equiv (x_1(t), ..., x_{d_x}(t))$ has a continuous state transition model $W$ as follows:

$$\dot{x}(t) = W(x(t); \theta) = \begin{bmatrix} [x(t), x(t)]_{K,1} \\ \vdots \\ [x(t), x(t)]_{K,d_x} \end{bmatrix} - x(t) + F, \tag{1.7}$$

where $\theta = (K, F)$. ◁

Figure 1.10 illustrates continuous versus discrete time: the line represents continuous time, but the system can only be observed at snapshots in time when measurements are collected, i.e., at discrete timepoints $t_n$. In the atmospheric sciences, the time between discrete timepoints is called the *forecast lead time* and is denoted by $\Delta t_n \equiv t_n - t_{n-1}$. Chaotic models are approximately linear for small forecast lead times and the nonlinearities of the models increase as the forecast lead time increases[8]. As the system becomes more nonlinear, updating with data becomes even more critical so that forecasts better match reality. Furthermore, nonlinearity increases the necessity of complex filtering algorithms—ones that have less strict assumptions than the Kalman filter. We focus on reasonably large forecast lead times, so for the purposes of this dissertation, a chaotic model is synonymous with a nonlinear model.

[8]The use of the Lyapunov exponent provides one way to quantify what we mean by "small" versus "large" forecast lead times. Its description is beyond the scope of this dissertation. For more information, we refer the reader to Cvitanović et al. (2015).

The continuous state transition model must be integrated to obtain the state at the timepoints when measurements are collected. Specifically, the state transition model at the discrete timepoint $t_n$ is:

$$x_{t_n} = m_{t_n}(x_{t_{n-1}}) + \eta_{t_n}, \quad \eta_t \sim \mathcal{N}(0, U_{t_n}),$$

where $\eta_t$ is the state disturbance term with mean zero and $d_x \times d_x$ variance matrix $U_{t_n}$. The state transition density is $f(x_{t_n} \mid x_{t_{n-1}}) = \phi[x_{t_n}; m_{t_n}(x_{t_{n-1}}); U_{t_n}]$. As before, $\mathcal{N}(\cdot, \cdot)$ denotes a normally distributed random variable and $\phi(\cdot; \cdot, \cdot)$ is its density but generalized to multivariate vectors. The function $m_{t_n}$ represents the integration of the state from time $t_{n-1}$ to $t_n$ using the continuous state transition model $W$; we call it the *discrete state transition model*. If the continuous state transition model can be analytically integrated, then the discrete state transition model is as follows:

$$m_{t_n}(x_{t_{n-1}}; \theta) = x_{t_{n-1}} + \int_{t_{n-1}}^{t_n} \dot{x}(\tau) d\tau.$$

Many interesting mathematical models, including the Lorenz systems, cannot be analytically integrated and must be numerically integrated instead. Numerical integrators are not perfect: they have *numerical error* and the error is a function of the numerical timestep $dt$. Smaller numerical error is achieved with smaller $dt$, but also increases computational time. Denote the numerical integrator by $\mathcal{I}(\cdot)$. Then, the discrete state transition model is as follows:

$$m_{t_n}(x_{t_{n-1}}; \theta) = \mathcal{I}_W(x_{t_{n-1}}; \theta) + O(e(dt)),$$

where $e$ is a function mapping the order of the integrator's error, e.g., $e(x) = x$ for forward Euler and $e(x) = x^4$ for fourth-order Runge Kutta, and the subscript $W$ on the numerical integrator $\mathcal{I}$ is to remind the reader that the integrator depends on the continuous state transition model $W$. Unless otherwise stated, we use second-order Runge Kutta with $dt = 0.01$ to integrate the Lorenz 1963 model and fourth-order Runge Kutta with $dt = 0.001$ for the Lorenz 2005 model. For both models, each $dt$ corresponds to one hour.

Many atmospheric science applications have a deterministic state transition model, i.e., the state disturbance $\eta_t$ is zero, and consequently the state transition density is degenerate. Even when the density is not degenerate, the first moment, i.e., the discrete state transition model, generally does not have an analytic form due to the continuous state transition model. Both issues add to the difficulty of filtering with atmospheric models. Despite these issues, sampling from the density is easy, thus we are partial to sampling algorithms for filtering.

For many parts of the dissertation, there is no need to distinguish between the continuous and discrete state transition models and thus we drop the subscript $n$ from $t_n$ for clarity, e.g., call the discrete state $x_t$ instead of $x_{t_n}$. Without loss of generality, the applications in the dissertation have measurements with the same forecast lead time for all $n, n' \geq 1$, i.e., $\Delta t_n = \Delta t_{n'}$. In reality, the forecast lead time depends on how often measurements are collected.

## 1.3 Measurement models

There are a wide variety of measurement models, as simple as the temperature model from Example 1.1.1 or as complicated as the precipitation model from Example 1.1.2. In the atmospheric sciences, measurement models tend to be a linear interpolation from the mathematical grid to the spatial resolution of the measurements, usually with Gaussian errors. For this reason, we focus on this particular measurement model in this dissertation.

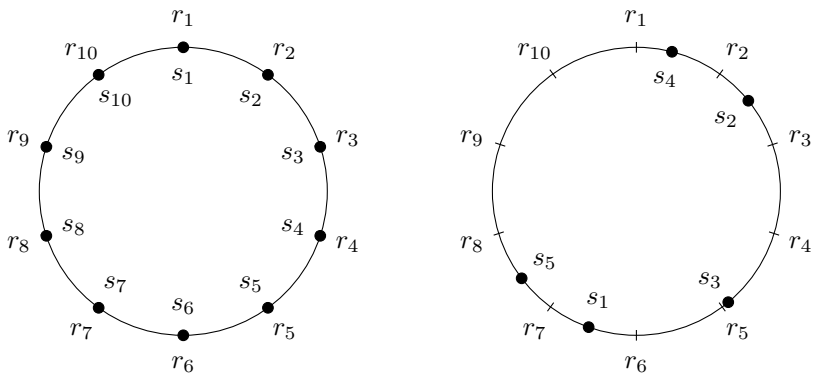
Suppose that measurements of the systems described in Section 1.2.1 are collected at discrete timepoints $t_n$. Let $y_{t_n}$ be the vector of observations at timepoint $t_n$. Then, a linear measurement model is

$$y_{t_n} = H_{t_n} x_{t_n} + \epsilon_{t_n}, \quad \epsilon_{t_n} \sim \mathcal{N}(0, V_{t_n}),$$

where $\epsilon_{t_n}$ represents the measurement error with mean zero and $d_y \times d_y$ variance matrix $V_{t_n}$. The measurement density is then $g(y_{t_n} \,|\, x_{t_n}) = \phi[y_{t_n}; H_{t_n} x_{t_n}, V_{t_n}]$. If measurements of the state variable $x_{t_n}$ are exactly collected at locations on the mathematical grid, $H_{t_n}$ is simply the $d_x \times d_x$ identity matrix $I_{d_x}$. We call this system *fully observed*. This is rarely the case—measurements are collected where equipment is placed, which has no bearing on the mathematical grid used to numerically integrate the state transition model. In this case, $H_{t_n}$ is a $d_y \times d_x$ matrix that linearly interpolates the state to the measurements. In other words, $H_{t_n}$ is a weighting matrix that places higher weights to parts of the state that are spatially closer to the measurement. We call this system *partially observed*.

**Example 1.3.1** (Lorenz 1963: measurement models)**.** If the state in the Lorenz 1963 model is fully observed, then all parts of the state, i.e., the $x$-, $y$-, and $z$-directions, are observed. If only the $x$- and $y$-directions are observed, then the system is partially observed and the linear mapping is:

$$H_{t_n} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$




(a) *Fully observed system.* (b) *Partially observed system.*

Figure 1.11: *Mathematical grid on a latitude band with measurement locations.* Similar to Figure 1.6, let $r_i$ and $s_j$ be the locations of the state and measurement of the $i$th and $j$th elements, respectively, of the $d_x$-vector $x_n$ and $d_y$-vector $y_n$. In this figure, the state has size $d_x = 10$ and the measurement has sizes $d_y = 10$ in the fully observed system (a) and $d_y = 5$ in the partially observed system (b). The tick marks represent locations of the state and the dots represent the locations of the measurements.

If the location at which the measurement is collected is exactly halfway between the locations of the $x$- and $y$-directions by some distance metric, then this system is also partially observed with the following linear mapping:

$$H_{t_n} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}.$$

◁

**Example 1.3.2** (Lorenz 2005: measurement models)**.** With the same latitude band from before, we illustrate fully observed and partially observed systems in Figure 1.11. In a fully observed system, similar to Figure 1.11a, $H_{t_n}$ is the identity matrix $I_{d_x}$. In a partially observed system, similar to Figure 1.11b, each row of $H_{t_n}$ sums to one with higher weights places on parts of the state that are closer to the measurement. ◁

## 1.4 Problems and outline

In this chapter, we introduced state-space models with examples of state transition and measurement models often used in the atmospheric sciences. We also highlighted several difficulties when filtering with atmospheric models; we summarize them here:

(1) Atmospheric models are chaotic. For the purposes of this dissertation, a chaotic model is synonymous with a nonlinear model.

(2) States are high-dimensional. For example, imagine forecasting temperature for the contiguous United States. We probably want a finer resolution than 80 square miles for our forecasts, so the dimensions of the state can be on the order of 10,000s or even 100,000s.

(3) There is generally no analytic expression for the state transition density $f$, but it is easy to sample from it.

(4) Mathematical models are subject to *model error*: the models are not perfect because they are only an approximation of the true system.

In **Part I**, we describe filtering in more detail. We first describe the optimal filtering densities and show that they are easily derived when both the state transition and measurement models are linear and have additive Gaussian noise, i.e., the assumptions of the state-space model for the Kalman filter to be applicable. When nonlinearities are introduced to the models, the optimal filtering densities rarely have an analytic expression. We review filtering algorithms to deal with the nonlinearities, paying particular attention to two sampling algorithms: the ensemble Kalman filter (EnKF) and a PF introduced by Gordon et al. (1993), the bootstrap filter (BF). We draw a novel connection between the EnKF and the BF. This connection facilitates the ability to combine the best qualities of both filters in our proposed methodology for parameter estimation with high-dimensional state space models.

Many methods with the PF have been developed to estimate parameters, but atmospheric scientists rarely use PFs in their applications due to the computational infeasibility of applying PFs to high-dimensional state-spaces. On the other hand, methods with the EnKF have been developed to estimate specific parameters but these methods cannot be generalized to estimate any parameter. Since many atmospheric scientists use the EnKF, we develop an algorithm in **Part II** to estimate any parameter in state-space models where EnKF can be appropriately applied. Furthermore, the algorithm can be easily added to existing implementations of EnKF, such as the Data Assimilation Research Testbed (DART)[9].

As mentioned earlier, we originally embarked on this journey to improve model error in atmospheric models. In **Part III**, we address the issue by introducing a low-rank linear correction to the continuous state transition

[9]DART is developed at the National Center for Atmospheric Research (NCAR). More information about DART can be found at `https://www.image.ucar.edu/DAReS/DART/`.

model so that it is applicable to high-dimensional state-spaces. When model error is stated in this way, it is a special case of parameter estimation with state-space models. We apply the parameter estimation algorithm developed in Part II to improve model error and thus forecasts.

# Part I

# Filtering

This page
intentionally
left blank

# 2 Filtering algorithms: a selected review

## 2.1 Optimal filtering

Filtering is an iterative procedure to forecast and then update the forecast with collected measurements. Though their distributions do not always have an analytic expression, we describe how the optimal filtering densities are derived in this section. Interested readers can refer to Doucet et al. (2001) and Crisan and Doucet (2002) for a similar treatment.

Since the main goal of filtering is to forecast, we first discuss the optimal forecast distribution. Let $y_{1:t}$ be the set of measurements collected up to time $t$, where $y_{m:n} \equiv (y_m, ..., y_n)$. Figure 2.1 is similar to the graphical model from Figure 1.1 but with nodes filled in to represent that the measurements $y_{1:t}$ are observed. By the definition of conditional probability, the forecast distribution

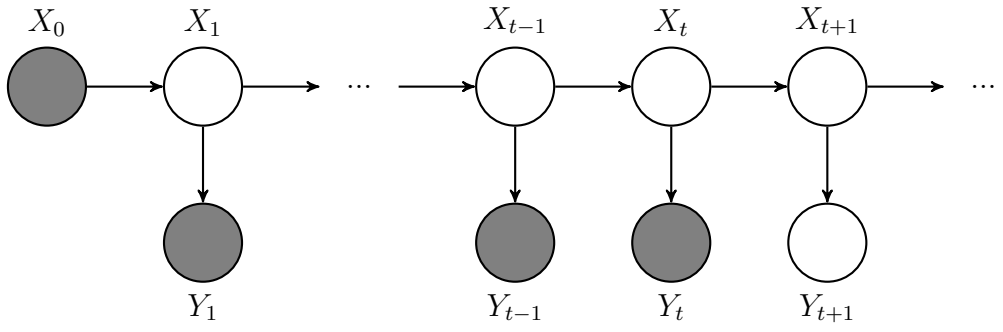



Figure 2.1: *State-space model with measurements collected up to time t: a graphical model.* Observed values are represented by filled circles.

is:

$$p(x_{t+1} \mid y_{1:t}, x_0) = \int p(x_{t+1} \mid x_t, y_{1:t}, x_0) p(x_t \mid y_{1:t}, x_0) dx_t.$$

By the Markov assumption, the state $X_{t+1}$ is conditionally independent of the past states and their measurements ($X_{t'}$ and $Y_{t'}$ for $t' < t$) given its previous state $X_t$, thus $p(x_{t+1} \mid x_t, y_{1:t}, x_0) = f(x_{t+1} \mid x_t)$ and

$$p(x_{t+1} \mid y_{1:t}, x_0) = \int f(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}, x_0) dx_t. \tag{2.1a}$$

The forecast random variable at time $t+1$ is denoted as $X_{t+1}^f \stackrel{d}{=} X_{t+1} \mid y_{1:t}, x_0$, where "$\stackrel{d}{=}$" denotes equal in distribution. The first two moments of $X_{t+1}^f$ are denoted as $\mu_{t+1}^f$ and $\Sigma_{t+1}^f$, where the superscript "$f$" is to remind the reader that they are moments of the $f$orecast distribution.

Notice that the forecast distribution requires the posterior distribution of the previous state, $p(x_t \mid y_{1:t}, x_0)$: called the *analysis* distribution. An application of Bayes' theorem provides the analysis distribution:

$$p(x_t \mid y_{1:t}, x_0) = \frac{g(y_t \mid x_t) p(x_t \mid y_{1:t-1}, x_0)}{p(y_t \mid y_{1:t-1}, x_0)}. \tag{2.1b}$$

The analysis random variable at time $t$ is denoted as $X_t^a \stackrel{d}{=} X_t \mid y_{1:t}, x_0$ and its mean and variance are $\mu_t^a$ and $\Sigma_t^a$, respectively. Again, the superscript "$a$" is to remind the reader that they are moments of the $a$nalysis distribution. The procedure to find the analysis distribution at time $t$ is called *updating* and is performed only after observing $y_t$. The analysis distribution is a function of the measurement density, the forecast distribution of the previous time, and the *predictive likelihood*:

$$p(y_t \mid y_{1:t-1}, x_0) = \int p(y_t \mid x_t, y_{1:t-1}, x_0) p(x_t \mid y_{1:t-1}, x_0) dx_t.$$

Since the measurement is conditionally independent of the previous measurements given its state, $p(y_t \mid x_t, y_{1:t-1}, x_0) = g(y_t \mid x_t)$ and thus the predictive likelihood is

$$p(y_t \mid y_{1:t-1}, x_0) = \int g(y_t \mid x_t) p(x_t \mid y_{1:t-1}, x_0) dx_t. \tag{2.2}$$

Many of the filtering algorithms need not calculate or sample from the predictive likelihood, but it is an important component to derive or estimate the likelihood, a crucial component for parameter estimation.

The set of equations in Equation (2.1) is called the *optimal filtering densities*. There are rarely analytic solutions for the densities. The rest of the chapter is dedicated to reviewing algorithms that derive or approximate the filtering densities in Equation (2.1). In preparation for Part II on parameter estimation, we also derive or approximate the predictive likelihood of Equation (2.2) when discussing the algorithms.

Section 2.2 discusses the Kalman filter, an algorithm to analytically solve the filtering densities when both the state transition and measurement models are linear and have additive and Gaussian errors. Section 2.3 discusses two extensions of the Kalman filter that loosen the requirement of linearity required in the Kalman filter: the extended Kalman filter (EKF) and the ensemble Kalman filter (EnKF). These extensions still require that the state disturbances and measurement errors be additive and Gaussian and thus cannot be used to solve for general state-space models. On the other hand, particle filters (PFs) are a class of sequential Monte Carlo (SMC) algorithms to approximately sample from the optimal filtering densities for the general state space model of Equation (1.1). In Section 2.4, we discuss the simplest PF algorithm: the bootstrap filter (BF). We end the chapter with a short discussion comparing the EnKF and PFs.

We first end this section by defining some notational conveniences and a note about the derivation of the algorithms.

### 2.1.1 Notation

The filtering densities up to time $t$ are all conditionally dependent on the initial condition $x_0$ and the measurements that have been collected, e.g., either $y_{1:t-1}$ or $y_{1:t}$. For notational succinctness, let $Y_0 = X_0$. For example, the succinct notation $y_{0:t} = \{x_0, y_{1:t}\}$ and the filtering densities $p(x_t \,|\, y_{1:t-1}, x_0)$, $p(x_t \,|\, y_{1:t}, x_0)$, and $p(y_t \,|\, y_{1:t-1}, x_0)$ are denoted as $p(x_t \,|\, y_{0:t-1})$, $p(x_t \,|\, y_{0:t})$, and $p(y_t \,|\, y_{0:t-1})$, respectively.

To distinguish the optimal filtering densities from their estimators in the following sections, we use the following shorthand when convenient:

$$\pi_{s:t}(z) \equiv p(z \,|\, y_{s:t}),$$

where $z \in \{x_1, x_2, ..., y_1, y_2, ...\}$ and $s, t \in \{1, 2, ...\}$. For example, the forecast density, analysis density, and predictive likelihood from Equation (2.1) are

denoted as

$$\begin{aligned}\pi_{0:t}(x_{t+1}) &= p(x_{t+1} \mid y_{0:t}),\\ \pi_{0:t}(x_t) &= p(x_t \mid y_{0:t}),\\ \pi_{0:t-1}(y_t) &= p(y_t \mid y_{0:t-1}),\end{aligned}$$

respectively. The distribution of the initial state $X_0$ is denoted as $\pi_0$. For notational succinctness, the initial state is considered an *a*nalysis random variable when no measurements $y_t$ have been collected, thus define $\pi_{1:0}(x_0) \equiv p(x_0) = \pi_0(x_0)$ and $\pi_{1:0}(x_1) \equiv p(x_1 \mid x_0)$. Let the first two moments of the initial condition be denoted as $\mu_0^a = \mathbb{E}[X_0]$ and $\Sigma_0^a = \mathbb{V}\text{ar}[X_0]$.

When discussing estimators constructed with Monte Carlo samples, such as the EnKF and the PF, the estimators are denoted with a superscript $M$ (e.g., $\pi_{s:t}^M$) to indicate the sample size $M$ to construct the estimator. The sampling algorithms are initialized with simple random samples $x_0^{a(m)} \sim \pi_0$ for $m = 1, ..., M$. The superscript "$(m)$" denotes the sample number, e.g., $x_t^{f(m)}$ and $x_t^{a(m)}$ denote the $m$th sample from the forecast and analysis distributions at time $t$, respectively. Let $\{x^{(m)}\}_{m=1}^M$ be a shorthand for a sample of size $M$, i.e., $\{x^{(1)}, ..., x^{(M)}\}$. Furthermore, simple random samples cannot always be generated, thus each sample may have a nonuniform weight associated with it. A sample $x^{(m)}$ with weight $w^{(m)}$ may be denoted as the sample $(x^{(m)}, w^{(m)})$. If the sample does not include a weight, it is assumed to be a simple random sample is generated with uniform weights $w^{(m)} = 1/M$ for $m = 1, ..., M$.

### 2.1.2 Derivation of the estimators: a remark

Due to the inductive nature of the filtering densities, in each following section, we derive estimators of the filtering densities by beginning with an estimator of the analysis distribution at time $t-1$ and plugging that particular estimator into the filtering densities at time $t$ to derive their estimators. This is true for all filtering algorithms presented with one exception: the Kalman filter. The state-space model under the assumptions of the Kalman filter has analytic expressions for the filtering densities.

## 2.2 Kalman filter

The state-space model under the assumptions of the Kalman filter is one of the rare models with analytic solutions to the optimal filtering densities (see

Kalman (1960) and Bucy and Kalman (1961) for the original papers and Section 13.2 of Bishop (2006) for a textbook review). We briefly derive the forecast and analysis distributions because they will be important later in the discussion of extensions of the Kalman filter.

Consider the following state-space model:

$$\begin{aligned} x_t &= M_t x_{t-1} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, U_t), \\ y_t &= H_t x_t + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, V_t), \end{aligned}$$

where $M_t \in \mathbb{R}^{d_x \times d_x}$ is a $d_x \times d_x$ matrix, $H_t \in \mathbb{R}^{d_y \times d_x}$ is a $d_y \times d_x$ matrix. The transition and observation densities are thus

$$\begin{aligned} f(x_t \,|\, x_{t-1}) &= \phi(x_t; M_t x_{t-1}, U_t), \\ g(y_t \,|\, x_t) &= \phi(y_t; H_t x_t, V_t), \end{aligned}$$

respectively. The initial state is also normally distributed with mean $\mu_0^a$ and variance $\Sigma_0^a$, i.e., $X_0 \sim \mathcal{N}(\mu_0^a, \Sigma_0^a)$.

Suppose the analysis random variable at time $t-1$ is normally distributed with mean $\mu_{t-1}^a$ and variance $\Sigma_{t-1}^a$: $X_{t-1} \,|\, y_{0:t-1} \sim \mathcal{N}(\mu_{t-1}^a, \Sigma_{t-1}^a)$. Since the convolution of two Gaussian random variables is also a Gaussian random variable, the forecast random variable at time $t$ is normally distributed with mean and variance equal to

$$\begin{aligned} \mu_t^f &= M_t \mu_t^a, \\ \Sigma_t^f &= U_t + M_t \Sigma_{t-1}^a M_t^T, \end{aligned}$$

respectively. Therefore, the forecast distribution at time $t$ is

$$\pi_{0:t-1}(x_t) = \phi(x_t; \mu_t^f, \Sigma_t^f). \tag{2.3}$$

Because normal random variables are defined by their first two moments, the following joint distribution is derived using the laws of total expectation, variance, and covariance:

$$\begin{bmatrix} X_t \\ Y_t \end{bmatrix} \,\Bigg|\, y_{0:t-1} \sim \mathcal{N}\left( \begin{bmatrix} \mu_t^f \\ H_t \mu_t^f \end{bmatrix}, \begin{bmatrix} \Sigma_t^f & \Sigma_t^f H_t^T \\ H_t \Sigma_t^f & V_t + H_t \Sigma_t^f H_t^T \end{bmatrix} \right). \tag{2.4}$$

After observing $y_t$, the predictive likelihood is

$$\pi_{0:t}(y_t) = \phi(y_t; H_t \mu_t^f, V_t + H_t \Sigma_t^f H_t^T), \tag{2.5}$$

obtained by marginalizing $X_t, Y_t \,|\, y_{0:t-1}$ over $X_t$. Plugging in Equations (2.3) and (2.5) into the optimal analysis distribution of Equation (2.1b), we have

$$\pi_{0:t}(x_t) = \frac{\phi(y_t; H_t x_t, V_t)\phi(x_t; \mu_t^f, \Sigma_t^f)}{\phi(y_t; H_t \mu_t^f, V_t + H_t \Sigma_t^f H_t^T)}. \tag{2.6}$$

The first two moments of the analysis random variable $X_t \,|\, y_{0:t}$ (and thus the analysis distribution) are calculated using the conditioning techniques of multivariate normal random variables, derived using Schur complements:

$$\begin{aligned}
\mu_t^a &\equiv \mathbb{E}(X_t \,|\, y_{0:t}) \\
&= \mathbb{E}(X_t \,|\, y_{0:t-1}) + \mathbb{C}\mathrm{ov}(X_t, Y_t \,|\, y_{0:t-1})\, \mathbb{V}\mathrm{ar}(Y_t \,|\, y_{0:t-1})^{-1}[y_t - \mathbb{E}(Y_t \,|\, y_{0:t-1})], \\
\Sigma_t^a &\equiv \mathbb{V}\mathrm{ar}(X_t \,|\, y_{0:t}) \\
&= \mathbb{V}\mathrm{ar}(X_t \,|\, y_{0:t-1}) - \mathbb{C}\mathrm{ov}(X_t, Y_t \,|\, y_{0:t-1})\, \mathbb{V}\mathrm{ar}(Y_t \,|\, y_{0:t-1})^{-1}\, \mathbb{C}\mathrm{ov}(Y_t, X_t \,|\, y_{0:t-1})),
\end{aligned}$$

thus $X_t \,|\, y_{1:t} \sim \mathcal{N}(\mu_t^a, \Sigma_t^a)$ and

$$\pi_{1:t}(x_t) = \phi(x_t; \mu_t^a, \Sigma_t^a) \tag{2.7}$$

with mean and variance equal to

$$\begin{aligned}
\mu_t^a &= \mu_t^f + \Sigma_t^f H_t^T (V_t + H_t \Sigma_t^f H_t^T)^{-1}(y_t - H_t \mu_t^f), \\
\Sigma_t^a &= \Sigma_t^f - \underbrace{\Sigma_t^f H_t^T (V_t + H_t \Sigma_t^f H_t^T)^{-1}}_{\text{Kalman gain}} H_t \Sigma_t^f,
\end{aligned}$$

respectively. The term above the underbrace is called the *Kalman gain* and is denoted as

$$K_t = \mathbb{C}\mathrm{ov}(X_t, Y_t \,|\, y_{0:t-1})\, \mathbb{V}\mathrm{ar}(Y_t \,|\, y_{0:t-1})^{-1} = \Sigma_t^f H_t^T (V_t + H_t \Sigma_t^f H_t^T)^{-1}. \tag{2.8}$$

The Kalman gain is a crucial component to update the state in the Kalman filter and its extensions. By induction, the filtering densities in Equations (2.3), (2.5), and (2.7) are the optimal filtering densities for all $t$.

## 2.3 Extensions of the Kalman filter

In this section, we discuss two extensions of the Kalman filter that loosen the requirement of linearity. Consider the following state-space model:

$$\begin{aligned}
x_t &= m_t(x_{t-1}) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, U_t), \\
y_t &= h_t(x_t) + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, V_t),
\end{aligned}$$

where $m_t : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$ and $h_t : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ are nonlinear mappings of the state. The state transition and observation densities are then:

$$\begin{aligned}
f(x_t \,|\, x_{t-1}) &= \phi(x_t; m_t(x_{t-1}), U_t), \\
g(y_t \,|\, x_{t-1}) &= \phi(y_t; h_t(x_t), V_t).
\end{aligned}$$

The initial state is also distributed normally with mean $\mu_0^a$ and variance $\Sigma_0^a$, i.e., $X_0 \sim \mathcal{N}(\mu_0^a, \Sigma_0^a)$. Notice that the state-space model is the same as the model under the Kalman filter, except that the first moments of both the state transition and measurement models are nonlinear with respect to the state because of the mappings $m_t$ and $h_t$, respectively.

Even if both the analysis random variable $X_{t-1}^a \overset{d}{=} X_{t-1} \,|\, y_{0:t-1}$ and the forecast random variable $X_t^f \overset{d}{=} X_t \,|\, y_{0:t-1}$ are normally distributed, $m_t(X_{t-1}^a)$ and $h_t(X_t^f)$ need not be normally distributed and therefore the distributions of $X_t \,|\, y_{0:t-1}$ and $X_t \,|\, y_{0:t}$, respectively, generally do not have analytic forms. The idea behind the extensions in this section is to approximate $m_t(X_{t-1}^a)$ and $h_t(X_t^f)$ with normal random variables, thus allowing the distributions of $X_t \,|\, y_{0:t-1}$ and $X_t \,|\, y_{0:t}$ (and hence the optimal filtering densities) to be approximated with normal distributions as well. Since normally distributed random variables are defined by their first two moments, these algorithms outline how to approximate their first two moments.

### 2.3.1 Extended Kalman filter

The extended Kalman filter circumvents the nonlinearity issue by linearizing the mappings around the current mean and variance estimates. Specifically, if $m_t$ and $h_t$ are sufficiently differentiable, their Taylor expansions are used to approximate the first two moments of their distributions. In this section, we outline the first-order extended Kalman filter (EKF), which uses only the first-order approximation of the Taylor expansion in the estimation of the means and variances. Interested readers can read more about extended Kalman filters in Einicke (2012).

Suppose the mean and variance of $X_{t-1}^a$ are approximated by $\tilde{\mu}_{t-1}^a$ and $\tilde{\Sigma}_{t-1}^a$, respectively. Further suppose $m_t$ is at least twice differentiable. Then, by the multivariate Taylor's theorem,

$$m_t(x) = m_t(\mu) + D^{(1)}m_t(\mu)(x-\mu) + O[(x-\mu)^T D^{(2)}m_t(\mu)(x-\mu)],$$

where $D^{(1)}m_t(\mu)$ and $D^{(2)}m_t(\mu)$ are the Jacobian and Hessian of $m_t$ evaluated at $\mu$, respectively. Let $\tilde{M}_t \equiv D^{(1)}m_t(\tilde{\mu}_{t-1}^a)$. Then, the mean and variance of $X_t^f$ are approximated as

$$\begin{aligned} \mu_t^f &\approx m_t(\tilde{\mu}_{t-1}^a) &&\equiv \tilde{\mu}_t^f, \\ \Sigma_t^f &\approx U_t + \tilde{M}_t \tilde{\Sigma}_t^a \tilde{M}_t^T &&\equiv \tilde{\Sigma}_t^f, \end{aligned}$$

respectively. Notice that the truncation error of $\tilde{\mu}_t^f$ and $\tilde{\Sigma}_t^f$ are on the order of the variance and fourth central moment of $X_{t-1}^a$, respectively, scaled by the

Hessian evaluated at $\mu^a_{t-1}$. The forecast distribution is then approximated as

$$\pi_{0:t-1}(x_t) \approx \phi(x_t; \tilde{\mu}^f_t, \tilde{\Sigma}^f_t). \tag{2.9}$$

Similarly, suppose $h_t$ is at least twice differentiable. Let $\tilde{H}_t$ be the Jacobian of $h_t$ evaluated at $\tilde{\mu}^f_t$. After observing $y_t$, the $\mu^f_t, \Sigma^f_t$, and $H_t$ terms that appear in the calculation of the Kalman filter's predictive likelihood and analysis distribution (Equations (2.5) and (2.7), respectively) are replaced by the approximations calculated here. Specifically, the predictive likelihood is approximated as

$$\pi_{0:t-1}(y_t) \approx \phi(y_t; \tilde{H}^T_t \tilde{\mu}^f_t, V_t + \tilde{H}_t \tilde{\Sigma}^f_t \tilde{H}^T_t), \tag{2.10}$$

and the analysis mean and variance are approximated as:

$$\begin{aligned} \mu^a_t &\approx \tilde{\mu}^f_t + \tilde{\Sigma}^f_t \tilde{H}^T_t (V_t + \tilde{H}_t \tilde{\Sigma}^f_t \tilde{H}^T_t)^{-1} [y_t - h_t(\tilde{\mu}^f_t)] \equiv \tilde{\mu}^a_t, \\ \Sigma^a_t &\approx \tilde{\Sigma}^f_t - \tilde{\Sigma}^f_t \tilde{H}^T_t (V_t + \tilde{H}_t \tilde{\Sigma}^f_t \tilde{H}^T_t)^{-1} \tilde{H}_t \tilde{\Sigma}^f_t \quad \equiv \tilde{\Sigma}^a_t, \end{aligned}$$

thus providing an approximation to the analysis distribution:

$$\pi_{0:t}(x_t) \approx \phi(x_t; \tilde{\mu}^a_t, \tilde{\Sigma}^a_t). \tag{2.11}$$

Notice that three approximations are made to derive the filtering densities for each time $t$. Two approximations come from the Taylor expansions of $m_t$ and $h_t$ used to derive the first two moments of the forecast and analysis distributions, respectively. The last approximation comes from the estimator of the analysis distribution at time $t-1$ used to derive the filtering densities at time $t$. This last approximation contains all the errors accumulated from the Taylor expansions up to time $t-1$ and are then propagated into estimations of future forecast and analysis distributions. The accumulation of errors is a common theme in all filtering algorithms with the exception of the Kalman filter.

With the EKF, the accumulation of errors can lead to unbounded error growth (Evensen, 1994, 2003). This error is part of a filtering problem called *filter divergence*, where the filter is no longer able to capture the true dynamics of the system (Evensen, 1992; Gottwald and Majda, 2013; Harlim and Majda, 2010; Miller et al., 1994). One solution is to use a higher-ordered EKF, which uses higher order terms in the Taylor expansion of $m_t$ and $h_t$ and thus provides better approximations to the means and covariances. An alternative (but partial) solution is to replace the mean and variance of the forecast with their sample estimates; this is the idea behind the EnKF in the next section. It

is only a partial solution because the measurement model is required to be linear in the state, like the Kalman filter: $h_t(x_t) = H_t x_t$. Although solutions have been developed to overcome this requirement, we will not discuss them in this dissertation. For more information, interested readers can refer to Evensen (2009, Section A1), Livings et al. (2008), Luo and Hoteit (2014), and references therein.

### 2.3.2 Ensemble Kalman filter

Evensen (1994) developed the EnKF in response to the filter divergence issues of the EKF. The EnKF uses Monte Carlo samples instead of a Taylor approximation to estimate the first two moments of the forecast distribution. In this literature, each sample is called an *ensemble member* and the set of samples is called the *ensemble*. For a textbook review of the EnKF, interested readers can refer to Evensen (2009).

Consider the following state-space model:

$$\begin{aligned} x_t &= m_t(x_{t-1}) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, U_t), \\ y_t &= H_t x_t + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, V_t), \end{aligned} \tag{2.12}$$

where $m_t : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$ is a nonlinear mapping of the state. Like the state-space model under the Kalman filter, the measurement is linear in the state with a $d_y \times d_x$ matrix $H_t$ that maps the state to the measurements. The state transition and measurement densities are:

$$\begin{aligned} f(x_t \,|\, x_{t-1}) &= \phi(x_t; m_t(x_{t-1}), U_t), \\ g(y_t \,|\, x_{t-1}) &= \phi(y_t; H_t x_t, V_t), \end{aligned}$$

respectively.

Suppose we have random samples from the approximate analysis distribution, $\{x_{t-1}^{a(m)}\}_{m=1}^M \dot{\sim} X_{t-1} \,|\, y_{0:t-1}$, where $X \dot{\sim} Y$ means that $X$ is approximately distributed as $Y$. Samples cannot be obtained from the optimal analysis distribution, so samples are generated from its estimator instead. At time $t$, samples from the forecast distribution are generated as follows:

$$x_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)}) = \phi(x_t; m_t(x_{t-1}^{a(m)}), U_t), \tag{2.13}$$

for each $m = 1, ..., M$. The sample mean and variance are calculated from the samples $\{x_t^{f(m)}\}_{m=1}^M$ and are denoted by $\hat{\mu}_t^f$ and $\hat{\Sigma}_t^f$, respectively. After observing $y_t$, samples from the analysis distribution are generated as a function of the forecast samples as follows:

$$\tilde{x}_t^{a(m)} = x_t^{f(m)} + \hat{\Sigma}_t^f H_t^T (V_t + H_t \hat{\Sigma}_t^f H_t^T)^{-1} [y_t - H_t x_t^{f(m)}]. \tag{2.14}$$

The sample mean and covariance from samples $\{\tilde{x}_t^{a(m)}\}_{m=1}^M$ should equal

$$\hat{\mu}_t^a = \hat{\mu}_t^f + \hat{\Sigma}_t^f H_t^T (V_t + H_t \hat{\Sigma}_t^f H_t^T)^{-1} [y_t - H_t \hat{\mu}_t^f], \tag{2.15a}$$

$$\hat{\Sigma}_t^a = \hat{\Sigma}_t^f - \hat{\Sigma}_t^f H_t^T (V_t + H_t \hat{\Sigma}_t^f H_t^T)^{-1} H_t \hat{\Sigma}_t^f, \tag{2.15b}$$

respectively.

Both Houtekamer and Mitchell (1998) and Burgers et al. (1998) discovered that the sample mean calculated from the samples $\{\tilde{x}_t^{a(m)}\}_{m=1}^M$ equals to $\hat{\mu}_t^a$, but the sample covariance did not equal to $\hat{\Sigma}_t^a$. To generate samples with a sample covariance that match $\hat{\Sigma}_t^a$, they both independently proposed to replace $y_t$ in Equation (2.14) with a closely related perturbed version $y_t^{(m)} = y_t + \epsilon_t^{(m)}$ where $\epsilon_t^{(m)} \sim \mathcal{N}(0, V_t)$, leading to the generation of analysis samples:

$$x_t^{a(m)} = x_t^{f(m)} + \hat{\Sigma}_t^f H_t^T (V_t + H_t \hat{\Sigma}_t^f H_t^T)^{-1} [y_t^{(m)} - H_t x_t^{f(m)}]. \tag{2.16}$$

This algorithm is called *EnKF with perturbed observations*. An alternative procedure is to deterministically scale $\{x_t^{f(m)}\}_{m=1}^M$ so that the resulting samples $\{x_t^{a(m)}\}_{m=1}^M$ have the proper mean and variance via

$$x_t^{a(m)} = \hat{\mu}_t^a + \hat{A}_t [x_t^{f(m)} - \hat{\mu}_t^f],$$

where $\hat{A}_t$ has the following property: $\hat{\Sigma}_t^a = \hat{A}_t \hat{\Sigma}_t^f \hat{A}_t$. There is no unique $\hat{A}_t$. Tippett et al. (2003) summarizes different algorithms to find $\hat{A}_t$; this set of algorithms is called *deterministic square-root filters* (or *square-root filters*, for short). Though rarely used in practice, we call the original formulation of the EnKF developed by Evensen (2003) with resulting samples

$$x_t^{a(m)} = \tilde{x}_t^{a(m)}$$

the *original EnKF* to distinguish it from the two other EnKF algorithms.

Due to the EnKF's close connection with the EKF, the optimal filtering densities of Equation (2.1) and predictive likelihood of Equation (2.2) are approximated similarly to the extended Kalman filter: instead of approximating the moments with their Taylor expansions, the moments are replaced with sample moments. Specifically, the forecast and analysis distributions are approximated as

$$\pi_{0:t-1}(x_t) \approx \phi(x_t; \hat{\mu}_t^f, \hat{\Sigma}_t^f), \tag{2.17a}$$

$$\pi_{0:t}(x_t) \approx \phi(x_t; \hat{\mu}_t^a, \hat{\Sigma}_t^a), \tag{2.17b}$$

**Inputs**: $\{x_{t-1}^{a(m)}\}_{m=1}^{M} \dot{\sim} p(x_{t-1} \,|\, y_{0:t-1})$
**Output**:
- $\{x_t^{f(m)}\}_{m=1}^{M} \dot{\sim} p(x_t \,|\, y_{0:t-1})$
- $\{x_t^{a(m)}\}_{m=1}^{M} \dot{\sim} p(x_t \,|\, y_{0:t})$

1. Forecast step: Sample $x_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)})$ for each $m = 1, ..., M$.
2. Observe $y_t$.
3. If performing inflation, calculate sample mean $\hat{\mu}_t^f$ from forecast samples $\{x_t^{f(m)}\}_{m=1}^{M}$, choose or estimate[a] inflation factor $\lambda_t \geq 1$, and set

$$\tilde{x}_t^{f(m)} = \hat{\mu}_t^f + \lambda_t(x_t^{f(m)} - \hat{\mu}_t^f).$$

   Otherwise, set $\tilde{x}_t^{f(m)} = x_t^{f(m)}$.
4. Update step:
   a) Calculate the Kalman gain

$$\hat{K}_t = \hat{\Sigma}_t^f H_t^T (V_t + H_t \hat{\Sigma}_t^f H_t^T)^{-1},$$

      where $\hat{\Sigma}_t^f$ is the sample covariance calculated from $\{x_t^{f(m)}\}_{m=1}^{M}$, including tapering if performing localization.
   b) Perform initial update:

$$\tilde{x}_t^{a(m)} = x_t^{f(m)} + \hat{K}_t[y_t - H_t \tilde{x}_t^{f(m)}] \text{ for each } m = 1, ..., M.$$

   c) Perform final update:
      - With original EnKF, set $x_t^{a(m)} = \tilde{x}_t^{a(m)}$ for each $m = 1, ..., M$.
      - With perturbed observations[b], sample $\epsilon_t^{(m)} \sim \mathcal{N}(0, V_t)$ and calculate

$$x_t^{a(m)} = \tilde{x}_t^{a(m)} + \hat{K}_t \epsilon_t^{(m)} \text{ for each } m = 1, ..., M.$$

      - With square-root filters[c], calculate the sample mean of $\{\tilde{x}_t^{a(m)}\}_{m=1}^{M}$, denoted as $\hat{\mu}_t^a$. Find $\hat{A}_t$ such that $\hat{\Sigma}_t^a = \hat{A}_t \hat{\Sigma}_t^f \hat{A}_t$, where $\hat{\Sigma}_t^a$ is the sample covariance calculated with $\{\tilde{x}_t^{a(m)}\}_{m=1}^{M}$. Then,

$$x_t^{a(m)} = \hat{\mu}_t^a + \hat{A}_t[\tilde{x}_t^{f(m)} - \hat{\mu}_t^f] \text{ for all } m = 1, ..., M.$$

[a] See Anderson (2007, 2009) for more information on how to apply adaptive inflation.
[b] In practice, this step is combined with Step 4b for computational efficiency, but we outline the algorithm in this way to contrast the various EnKF updating algorithms.
[c] See Tippett et al. (2003) for more information on finding such an $\hat{A}_t$.

Algorithm 2.1: *Ensemble Kalman filter.*

respectively (cf. Equations (2.9) and (2.11)). Furthermore, the predictive likelihood is approximated as

$$\pi_{0:t-1}(y_t) \approx \phi(y_t; H_t^T \hat{\mu}_t^f, V_t + H_t \hat{\Sigma}_t^f H_t^T) \tag{2.17c}$$

(cf. Equation (2.10)).

Even though the development of the EnKF was inspired by the EKF, unlike the EKF, the sample mean $\hat{\mu}_t^a$ and variance $\hat{\Sigma}_t^a$ are not used to forecast the next set of samples. In practice, the samples $\{x_t^{a(m)}\}_{m=1}^M$ are used in the next step of the algorithm to generate samples from the forecast distribution at time $t+1$. Therefore, the errors associated with the generation of the analysis samples are propagated to the next step of the algorithm, also leading to the possibility of filter divergence. To overcome filter divergence, the following two adjustments are highly recommended by EnKF experts:

- **Variance inflation**. Deterministically scale the forecast samples $\{x_t^{f(m)}\}_{m=1}^M$ to increase its sample variance via:

$$\tilde{x}_t^{f(m)} = \hat{\mu}_t^f + \lambda_t (x_t^{f(m)} - \hat{\mu}_t^f), \tag{2.18}$$

  where $\lambda_t > 1$ is chosen by the user (Hamill et al., 2001) or estimated with data (Anderson, 2007, 2009) (called *adaptive inflation* in the literature).

- **Localization**. Taper the sample forecast covariance matrix via:

$$\hat{\Sigma}_t^f \circ T(d),$$

  where "$\circ$" denotes the Schur product and $T(d)$ is a tapering covariance matrix that sets covariances to 0 when the distance between two spatial locations of the state are greater than $d$. We refer the reader to Bickel and Levina (2008) for more information on tapering and to Furrer and Bengtsson (2007) for more information on tapering in the context of EnKF.

In practice, the above two adjustments are crucial to prevent particle collapse when applying the EnKF to high-dimensional state spaces. For this reason, we include the two adjustments in Algorithm 2.1, which summarizes the EnKF algorithm.

## 2.4 Bootstrap filter: a particle filter

Particle filters (PFs) are a class of algorithms to approximate the filtering densities of the general state space model outlined in Equation (1.1). This

state-space model need not have additive and Gaussian errors as with the extensions of the Kalman filter. The idea behind these methods is similar to the EnKF: a Monte Carlo sample is used to estimate the optimal filtering distributions (or point estimates of them). The samples are called *particles* instead of ensemble members. The sampling mechanism behind the particle filter is importance sampling applied sequentially in time, hence its other name in the literature: *sequential Monte Carlo (SMC)* or *sequential importance sampling and resampling (SISR)*.

In this section, we present the simplest particle filter: the bootstrap filter. Most presentations of the BF present it as a SISR algorithm, where the prior transition density $f(x_t \mid x_{t-1})$ is used as the proposal distribution for the importance sampling[1]. Instead, our presentation follows the slight modification of the bootstrap filter presented in Crisan and Doucet (2002, Section III-A). This particular presentation is often used in the probabilistic analysis of particle filters (Cappé et al., 2006; Del Moral, 2004, 2013), where the bootstrap filter is presented as approximating the optimal filtering densities in Equation (2.1) with empirical densities[2]. The reason for choosing this particular presentation over its usual presentation is to make clear how the EnKF is connected to the bootstrap filter: EnKF uses a different estimator to the optimal filtering densities than the bootstrap filter; we elaborate on this further in Section 3.1.

Suppose we have random samples from an approximation of the analysis distribution, $\{x_{t-1}^{a(m)}\}_{m=1}^{M} \sim X_{t-1} \mid y_{0:t-1}$. The empirical density

$$\pi_{0:t-1}^{M}(x_{t-1}) \equiv \frac{1}{M} \sum_{m=1}^{M} \delta_{x_{t-1}^{a(m)}}(x_{t-1})$$

is an approximation of $\pi_{0:t-1}(x_{t-1})$, the optimal density of $X_{t-1}^{a}$; the superscript $M$ is to remind the reader that $\pi_{0:t-1}^{M}$ is an estimator of $\pi_{0:t-1}$ and that it depends on the sample size $M$. Plug this approximation into the optimal forecast distribution of Equation (2.1a) to generate forecast particles at time $t$:

$$x_t^{f(m)} \sim \int f(x_t \mid x_{t-1}) \pi_{0:t-1}^{M}(x_{t-1}) dx_{t-1} = \frac{1}{M} \sum_{m=1}^{M} f(x_t \mid x_{t-1}^{a(m)}),$$

[1]For a more general overview of particle filtering, we refer interested readers to Cappé et al. (2007), Doucet and Johansen (2011), and references therein.

[2]The approximations are referred to as *mean field approximations*.

for each $m = 1, ..., M$. In the usual presentation of the bootstrap filter[3], $x_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)})$ for each $m = 1, ..., M$. Note that these are not samples from $\pi_{0:t-1}(x_t)$; they are samples from its estimator, since an estimator of the analysis distribution $\pi_{0:t-1}^M(x_{t-1})$ is used to sample from the forecast distribution instead of the optimal analysis distribution $\pi_{0:t-1}(x_{t-1})$. These samples are used to provide an estimator to the forecast distribution:

$$\pi_{0:t-1}^M(x_t) = \frac{1}{M} \sum_{m=1}^{M} \delta_{x_t^{f(m)}}(x_t). \tag{2.19}$$

After observing $y_t$, the state is updated using this estimator of the forecast distribution.

Recall that the predictive likelihood is required in the calculation of the analysis distribution. Plugging the approximation of the forecast distribution ($\pi_{0:t-1}^M(x_t)$) into the optimal predictive likelihood of Equation (2.2) provides an approximation to the predictive likelihood:

$$\pi_{0:t-1}^M(y_t) \equiv \int g(y_t \,|\, x_t) \pi_{0:t-1}^M(x_t) dx_t = \frac{1}{M} \sum_{m=1}^{M} g(y_t \,|\, x_t^{f(m)}) = \frac{1}{M} \sum_{m=1}^{M} l_t^{(m)}, \tag{2.20}$$

where $l_t^{(m)} \equiv g(y_t \,|\, x_t^{f(m)})$. Now, the analysis distribution is approximated by plugging the estimators of the forecast distribution and the predictive likelihood into the optimal analysis density of Equation (2.1b):

$$\begin{aligned} \tilde{\pi}_{0:t}^M(x_t) &\equiv \frac{g(y_t \,|\, x_t) \pi_{0:t-1}^M(x_t)}{\pi_{0:t-1}^M(y_t)} \\ &= \sum_{m=1}^{M} \frac{g(y_t \,|\, x_t^{f(m)})}{\sum_{n=1}^{M} l_t^{(n)}} \delta_{x_t^{f(m)}}(x_t) \\ &= \sum_{m=1}^{M} \frac{l_t^{(m)}}{\sum_{n=1}^{M} l_t^{(n)}} \delta_{x_t^{f(m)}}(x_t). \end{aligned}$$

Let $\tilde{x}_t^{a(m)} = x_t^{f(m)}$ and $\tilde{w}_t^{(m)} = l_t^{(m)} / \sum_{n=1}^{M} l_t^{(n)}$ for each $m = 1, ..., M$. The weights $\{\tilde{w}_t^{(m)}\}_{m=1}^M$ are called *importance weights*. Then,

$$\tilde{\pi}_{0:t}^M(x_t) = \sum_{m=1}^{M} \tilde{w}_t^{(m)} \delta_{\tilde{x}_t^{a(m)}}(x_t) \tag{2.21}$$

[3]This technicality is not important in the mathematical analysis of SISR algorithms, because SISR is considered to be an equivalent interpretation of Feynman-Kac mean field Interacting Particle System models. See Del Moral (2013) for a comprehensive review.

is an estimator of the analysis distribution $\pi_{0:t}(x_t)$. We make a few remarks about the update step:

- Technically, the predictive likelihood need not be calculated to sample from the analysis distribution. Only the measurement density needs to be evaluated with the forecast samples, i.e., $l_t^{(m)} = g(y_t \mid x_t^{f(m)})$, which are normalized in the calculation of the importance weights without evaluating the estimator of the predictive likelihood.
- The particles $x_t^{f(i)}$ from the forecast step and $\tilde{x}_t^{a(i)}$ from the analysis step do not differ; only their weights have changed.

With high probability, samples with low importance weights will become irrelevant at future filtering steps, causing *particle collapse* (or *particle degeneracy*), where all samples but one have an importance weight of zero. Resampling helps remove those samples with low probability mass. There are many resampling algorithms proposed in the literature; see Section 2.4.1 for a selected review. Let $\{x_t^{a(m)}\}_{m=1}^M$ be a simple random sample from $\tilde{\pi}_{0:t}^M(x_t)$. These samples provide another estimator of the analysis distribution:

$$\pi_{0:t}^M(x_t) \equiv \frac{1}{M} \sum_{m=1}^{M} \delta_{x_t^{a(m)}}(x_t).$$

Since resampling increases the Monte Carlo variance, the resampling step is, in practice, only performed when the effective sample size

$$ESS = \left( \sum_{m=1}^{M} (\tilde{w}_t^{(m)})^2 \right)^{-1}$$

is below some threshold[4]. Either estimator of the analysis distribution, $\tilde{\pi}_{0:t}^M(x_t)$ or $\pi_{0:t}^M(x_t)$, can be used in the next filtering step, repeating the algorithm outlined in this section. Algorithm 2.2 summarizes the BF algorithm, where the user chooses whether or not to resample after every time step. Without loss of generality, we assume that resampling is performed at each time $t$ in our presentation.

[4]For more information on effective sample sizes, refer to Doucet and Johansen (2011, Section 3.5) and references therein.

**Inputs**: $\{x_{t-1}^{a(m)}\}_{m=1}^{M} \dot{\sim} p(x_{t-1} \,|\, y_{0:t-1})$
**Output**:
- $\{x_t^{f(m)}\}_{m=1}^{M} \dot{\sim} p(x_t \,|\, y_{0:t-1})$
- $\{x_t^{a(m)}\}_{m=1}^{M} \dot{\sim} p(x_t \,|\, y_{0:t})$

1. Forecast step: Sample $x_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)})$ for each $m = 1, ..., M$.

2. Observe $y_t$.

3. Update step:

   a) Calculate

   $$l_t^{(m)} = w_{t-1}^{(m)} g(y_t \,|\, x_t^{f(m)})$$

   for each $m = 1, ..., M$.

   b) Calculate

   $$\tilde{x}_t^{a(m)} = x_t^{f(m)},$$
   $$\tilde{w}_t^{(m)} = l_t^{(m)} / \textstyle\sum_{n=1}^{M} l_t^{(n)}$$

   for each $m = 1, ..., M$.

4. Resample step: If resampling,

   $$j_m = i \text{ with probability } \tilde{w}_t^{(i)},$$
   $$x_t^{a(m)} = \tilde{x}_t^{(j_m)},$$
   $$w_t^{(m)} = 1/M,$$

   for each $m = 1, ..., M$. Otherwise,

   $$x_t^{a(m)} = \tilde{x}_t^{(m)},$$
   $$w_t^{(m)} = \tilde{w}_t^{(m)},$$

   for each $m = 1, ..., M$.

Algorithm 2.2: *Bootstrap filter.*

### 2.4.1 Resampling algorithms

In this section, we discuss resampling broadly, outside of the context of filtering. The goal in this section is to obtain a simple random sample $\{x_m\}_{m=1}^M$ from $p(x)$ when we have a random, but not necessarily simple, sample $\{(\tilde{x}_m, \tilde{w}_m)\}_{m=1}^M$ from the density $p(x)$. Each sample $\tilde{x}_m$ has a weight $\tilde{w}_m$ for each $m = 1, ..., M$ and $\sum_{m=1}^M \tilde{w}_m = 1$. These samples form the empirical density estimator $\tilde{p}(x) \equiv \sum_{m=1}^M \tilde{w}_m \delta_{\tilde{x}_m}(x)$. Resampling is performed by sampling indices with weights $\{\tilde{w}_m\}_{m=1}^M$:

$$j_m = i \text{ with probability } \tilde{w}_i \tag{2.22}$$

for each $m = 1, ..., M$ and using the resampled indices $\{j_m\}_{m=1}^M$ to obtain simple random samples $\{x_m\}_{m=1}^M$ from $\tilde{p}(x)$:

$$x_m = \tilde{x}_{j_m} \text{ with probability } w_m = 1/M.$$

We discuss a few resampling algorithms to sample the indices $\{j_m\}_{m=1}^M$ as per Equation (2.22):

- **Multinomial sampling**: Sampling indices as in Equation (2.22) is performed using an inverse transform sampling algorithm. Sample auxiliary random variables

  $$u_m \sim \mathrm{U}(0, 1) \text{ for each } m = 1, ..., M,$$

  where $\mathrm{U}(a, b)$ is the continuous uniform random variable on the interval $[a, b]$. These auxiliary random variables are used to select indices as follows:

  $$j_m = \sum_{n=1}^M n \mathbb{I}\left[u_n \in \left[\sum_{i=1}^{n-1} \tilde{w}_i, \sum_{i=1}^n \tilde{w}_i\right)\right] \text{ for each } m = 1, ..., M, \tag{2.23}$$

  where $\mathbb{I}$ is the indicator function (i.e., $\mathbb{I}(x \in A) = 1$ when $x \in A$ and zero otherwise) and $\sum_{i=1}^0 \tilde{w}_i \equiv 0$.

- **Stratified sampling**: Sample $\tilde{u}_m \sim \mathrm{U}(0, 1)$ for each $m = 1, ..., M$. Calculate auxiliary random variables

  $$u_m = \frac{(m-1) + \tilde{u}_m}{M} \quad \text{for each } m = 1, ..., M$$

  and then select indices according to Equation (2.23).

- **Systematic sampling**: Sample $\tilde{u} \sim U(0,1)$. Calculate

$$u_m = \frac{(m-1)+\tilde{u}}{M} \quad \text{for each } m = 1, ..., M$$

  and then select indices according to Equation (2.23).

According to Doucet and Johansen (2011, Section 3.5), the sampling algorithms in decreasing order of popularity and efficiency are: systematic sampling, residual sampling (not discussed), and multinomial sampling. For a more thorough discussion of resampling in the context of PFs, we refer the reader to Doucet and Johansen (2011, Sections 3.4 and 3.5), Douc et al. (2005), and Hol et al. (2006). We conclude this section with a theorem that will be important in a later discussion on the connection between the EnKF and the BF.

**Theorem 2.4.1.** *If $\tilde{w}_m = 1/M$ for all $m = 1, ..., M$, stratified and systematic random sampling returns the simple random samples $x_m = \tilde{x}_m$.*

*Proof.* By Equation (2.23),

$$j_m = \sum_{n=1}^{M} n\mathbb{I}\left[u_n \in \left[\tfrac{n-1}{M}, \tfrac{n}{M}\right)\right] \text{ for each } m = 1, ..., M.$$

With stratified or systematic sampling, $\int_{(n-1)/M}^{n/M} p(u_k) = 1$ if $k = n$ and zero otherwise, therefore $\mathbb{I}[u_n \in [\frac{n-1}{M}, \frac{n}{M})] = 1$ if $n = m$ and zero otherwise. Therefore, $j_m = m$ and $x_m = \tilde{x}_m$. □

## 2.5 Ensemble Kalman filter and particle filter: large sample asymptotics versus practicalities

For all algorithms but the Kalman filter, if the derivation of the filtering densities began with the true analysis distribution at time $t-1$ instead of its estimator, it is straightforward to analyze the convergence of the forecast and analysis distributions for a particular time $t$ with large sample asymptotics. However, a nontrivial caveat of each algorithm is that estimators are inductively being used to approximate the filtering densities. For example, both the EnKF and BF use samples from an *estimator* of the analysis distribution

at time $t-1$ to then derive *estimators* of the filtering densities at time $t$. Inductively applying estimators complicates analyses of large sample asymptotics. Even though the error may be small at a particular time, the errors can accumulate and eventually lead to non-convergence of the estimators at some future timepoint (see Section IV-A in Crisan and Doucet (2002) for an example).

Many of the large asymptotic results with the EnKF are analyzed for a particular time $t$ assuming that one has the true analysis distribution from the previous timepoint $t-1$ (Furrer and Bengtsson, 2007; Lei et al., 2010). Very few analyze the full algorithm with the accumulation of errors from inductively plugging in estimators to form new estimators at future timepoints. The few that we have seen are with linear $m_t$, i.e., $m_t(x_{t-1}) = M_t x_{t-1}$, thus simplifying to the state-space model of the Kalman filter (Le Gland et al., 2009; Mandel et al., 2009). In fact, Le Gland et al. (2009) mention that the current formulation of the EnKF needs to account for importance weights, like the BF, for large asymptotic analyses to apply for nonlinear $m_t$. In addition, Lei et al. (2010) conclude that the EnKF is not robust to any deviation from the Gaussianity assumption of the analysis distribution. In practice, however, the EnKF is widely used to filter atmospheric models, especially for high-dimensional state-spaces, despite not having large asymptotic results for the full EnKF algorithm.

On the other hand, the PF has an abundance of large asymptotic results on the weak convergence of the filtering densities[5], i.e., $\pi^M_{0:t-1}(x_t) \to \pi_{0:t-1}(x_t)$, $\tilde{\pi}^M_{0:t}(x_t) \to \pi_{0:t}(x_t)$, and $\pi^M_{0:t}(x_t) \to \pi_{0:t}(x_t)$ when the number of particles $M \to \infty$, but the analyses have mostly been proven for fixed state dimensions. In fact, Bengtsson et al. (2008) concludes that the PF is impractical for high-dimensional state-spaces due to the inevitability of particle collapse, further deterring many atmospheric scientists from applying the PF to their applications of interest (Snyder et al., 2008).

[5]For a short review, see Crisan and Doucet (2002). For textbook reviews, see Del Moral (2004) and Del Moral (2013).

# 3 Ensemble Kalman filter: its connection to the particle filter

Consider the state-space model under the EnKF from Equation (2.12), repeated here for convenience:

$$\begin{aligned} x_t &= m_t(x_{t-1}) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, U_t), \\ y_t &= H_t x_t + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, V_t). \end{aligned}$$

The state transition and measurement densities are:

$$\begin{aligned} f(x_t \,|\, x_{t-1}) &= \phi(x_t; m_t(x_{t-1}), U_t), \\ g(y_t \,|\, x_{t-1}) &= \phi(y_t; H_t x_t, V_t), \end{aligned}$$

respectively. In this chapter, we present a novel interpretation of the EnKF that clearly highlights the relationship between the EnKF and the PF. The relationship is helpful in the next part of the dissertation on parameter estimation with the EnKF, where we propose a hybrid approach of applying the EnKF to filter the states and the PF to filter parameters.

To highlight the similarities between the EnKF and PF, we present the EnKF in a different manner than its usual presentation, i.e., the one previously presented in Section 2.3.2. Instead, Section 3.1 presents the algorithm similar to the presentation of the BF in Section 2.4. Throughout the discussion, we remark on the similarities and differences between the EnKF and BF, which are summarized in Section 3.1.1. In particular, we point out three main differences: the choice of estimator for the forecast distribution and two approximations introduced in the derivation of estimators to the analysis distribution. The choice of estimator has previously appeared in the kernel density estimation (KDE) literature and we discuss the connection in Section 3.2.

Not only does the re-interpretation aid the development of novel parameter estimation methodology, it connects the EnKF to the PF and its abundant

large sample asymptotic results. We hope that this connection will facilitate exploration to understand the practical effectiveness of the EnKF in filtering high-dimensional state-spaces. Section 3.2 is the beginning of this exploration. In that section, we present the EnKF with only the first approximation, which uses a different estimator for the forecast distribution than the empirical density estimator as with the BF. It turns out that this particular filter is a special case of a pre-regularized particle filter with Gaussian kernels. Special cases of this particular filter have been applied to high-dimensional state-spaces with poor results. From this, we conjecture that the practicality of the EnKF lies in the other approximations, which we discuss in Section 3.3 and leave for future exploration. Sections 3.2 and 3.3 may be omitted without affecting the rest of the dissertation.

## 3.1 Ensemble Kalman filter: a re-interpretation

Like the BF, we begin with simple random samples from the approximate analysis distribution, $\{x_{t-1}^{a(m)}\}_{m=1}^{M} \sim p(x_{t-1} \,|\, y_{0:t-1})$, that form the empirical density estimator

$$\psi_{0:t-1}^{M}(x_{t-1}) \equiv \frac{1}{M} \sum_{m=1}^{M} \delta_{x_{t-1}^{a(m)}}(x_{t-1}).$$

This estimator is then used to generate samples from the forecast distribution:

$$x_t^{f(m)} \sim \int f(x_t \,|\, x_{t-1}) \psi_{0:t-1}^{M}(x_{t-1}) dx_{t-1} = \frac{1}{M} \sum_{m=1}^{M} \phi(x_t; m_t(x_{t-1}^{a(m)}), U_t).$$

In the usual presentation of the BF, this last sampling step is instead $x_t^{f(m)} \sim \phi(x_t; m_t(x_{t-1}^{a(m)}), U_t)$, which exactly matches the sampling mechanism in the original presentation of EnKF (cf. Equation (2.13)). Up to this point, the EnKF and the BF do not differ: both generate forecast samples $\{x_t^{f(m)}\}_{m=1}^{M}$ in the same way. The difference begins with the choice of estimator to approximate the forecast distribution, which consequently changes the updating algorithm.

Recall that the BF uses the empirical density as an estimator of the forecast distribution (see Equation (2.19)). Instead of the empirical density, the EnKF

approximates the forecast distribution with a mixture of Gaussian densities:

$$\psi^M_{0:t-1}(x_t) \equiv \frac{1}{M}\sum_{m=1}^{M}\phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f). \tag{3.1}$$

Each Gaussian density is centered at the forecast samples $\{x_t^{f(m)}\}_{m=1}^M$ and has a variance equal to the sample variance calculated from the forecast samples, denoted by $\hat{\Sigma}_t^f$. In the KDE literature, a mixture of Gaussian densities, such as the estimator above, is proposed as an alternative estimator to the empirical density estimator for probability densities. We make this connection and further elaborate in Section 3.2. We introduce terminology from the KDE literature to facilitate the rest of our discussion. The Gaussian in the mixture is a particular choice of a *kernel density* (or, *kernel* for short). The mean and variance of the Gaussian, e.g., $x_t^{f(m)}$ and $\hat{\Sigma}_t^f$, respectively, are called the *center* and *bandwidth* of the kernel.

After observing $y_t$, the state is updated using the estimator of the forecast distribution in Equation (2.17a). Plugging the estimator of the forecast distribution ($\psi^M_{0:t-1}$) into the optimal predictive likelihood of Equation (2.2) provides an estimator to the predictive likelihood:

$$\begin{aligned}
\psi^M_{0:t-1}(y_t) &\equiv \int g(y_t \mid x_t)\psi^M_{0:t-1}(x_t)dx_t \\
&= \frac{1}{M}\sum_{m=1}^{M}\int \phi(y_t; H_t x_t, V_t)\phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f)dx_t \\
&= \frac{1}{M}\sum_{m=1}^{M}\phi(y_t; H_t x_t^{f(m)}, V_t + H_t\hat{\Sigma}_t^f H_t^T) \qquad (3.2)\\
&= \frac{1}{M}\sum_{m=1}^{M} l_t^{(m)}, \qquad (3.3)
\end{aligned}$$

where $l_t^{(m)} = \phi(y_t; H_t x_t^{f(m)}, V_t + H_t\hat{\Sigma}_t^f H_t^T)$. The summand in Equation (3.2) should remind the reader of the predictive likelihood of the Kalman filter (see Equation (2.5)). The estimator of the analysis distribution is then derived by plugging approximations of the forecast distribution and the predictive likelihood ($\psi^M_{0:t-1}(x_t)$ and $\psi^M_{0:t-1}(y_t)$, respectively) into the optimal analysis

density in Equation (2.1b):

$$\begin{aligned}
\vartheta_{0:t}^M(x_t) &\equiv \frac{\phi(y_t; H_t x_t, V_t)\psi_{0:t-1}^M(x_t)}{\psi_{0:t-1}^M(y_t)} \\
&= \frac{\frac{1}{M}\sum_{m=1}^M \phi(y_t; H_t x_t, V_t)\phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f)}{\frac{1}{M}\sum_{n=1}^M l_t^{(n)}} \\
&= \sum_{m=1}^M \frac{l_t^{(m)}}{\sum_{n=1}^M l_t^{(n)}} \frac{\phi(y_t; H_t x_t, V_t)\phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f)}{l_t^{(m)}} \\
&= \sum_{m=1}^M \frac{l_t^{(m)}}{\sum_{n=1}^M l_t^{(n)}} \underbrace{\frac{\phi(y_t; H_t x_t, V_t)\phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f)}{\phi(y_t; H_t x_t^{f(m)}, V_t + H_t\hat{\Sigma}_t^f H_t^T)}}
\end{aligned}$$

The term denoted by the underbrace should remind the reader of the analysis distribution of the Kalman filter (see Equation (2.6)). Applying the conditioning technique of multivariate normal random variables to the term above the underbrace, an analytic form for the mixture distribution is derived:

$$\vartheta_{0:t}^M(x_t) = \sum_{m=1}^M \tilde{w}_t^{(m)} \phi(x_t; \tilde{x}_t^{a(m)}, \hat{\Sigma}_t^a) \tag{3.4}$$

with

$$\tilde{x}_t^{a(m)} = x_t^{f(m)} + \hat{\Sigma}_t^f H_t^T (V_t + H_t\hat{\Sigma}_t^f H_t^T)^{-1}(y_t - H_t x_t^{f(m)}), \tag{3.5a}$$

$$\hat{\Sigma}_t^a = \hat{\Sigma}_t^f - \hat{\Sigma}_t^f H_t^T (V_t + H_t\hat{\Sigma}_t^f H_t^T)^{-1} H_t \hat{\Sigma}_t^f, \tag{3.5b}$$

and importance weights $\tilde{w}_t^{(m)} = l_t^{(m)} / \sum_{n=1}^M l_t^{(n)}$ for each $m = 1, ..., M$. Notice that each kernel in the estimator of the forecast distribution is updated with the measurement $y_t$ and each kernel's weight is updated with the measurement density evaluated at the measurement $y_t$ consequently changing the uniform weight to a nonuniform weight $\tilde{w}_t^{(m)}$. In other words, there are two components to the update: the kernel itself and its weight. This particular approximation, however, is not used in the EnKF. Two approximations are introduced: (1) the importance weights are approximated to be uniform and (2) the bandwidth $\hat{\Sigma}_t^a$ is replaced by another bandwidth depending on the EnKF algorithm.

We discuss the first approximation. The importance weights are approximated to be uniform:

$$\tilde{w}_t^{(m)} \approx 1/M, \tag{3.6}$$

providing another estimator to the analysis distribution ($\pi_{0:t}(x_t)$):

$$\tilde{\vartheta}^M_{0:t}(x_t) \equiv \frac{1}{M}\sum_{m=1}^{M} \phi(x_t; \tilde{x}_t^{a(m)}, \hat{\Sigma}_t^a). \tag{3.7}$$

We make a few remarks about this estimator:

- Notice that this approximation effectively removes one of the two modes to update the state: the weight.
- Contrast this estimator to the estimator of the bootstrap filter in Equation (2.21). After a similar step in the bootstrap filter, the estimator is a mixture of point masses centered at the analysis samples $\tilde{x}_t^{a(m)}$ that do *not* differ from the forecast samples $x_t^{f(m)}$; only their weights differ. With the EnKF, the samples $x_t^{f(m)}$ and $\tilde{x}_t^{a(m)}$ *do* differ because of the Kalman gain term, i.e., $K_t \equiv \hat{\Sigma}_t^f H_t^T (V_t + H_t \hat{\Sigma}_t^f H_t^T)^{-1}$, but their weights do not change because of the approximation made in Equation (3.6). Therefore, the updating mechanism of the EnKF solely relies on the Kalman gain when it should additionally depend on the importance weight. On the other hand, the updating mechanism of the BF only relies on the importance weights.
- Compare each kernel's moments $\tilde{x}_t^{a(m)}$ and $\hat{\Sigma}_t^a$ to the analysis samples derived in the initial exposition of EnKF (Section 2.3.2). Specifically, compare Equation (2.14) with (3.5a) and Equation (2.15b) with (3.5b). Both pairs of equations are equal.

At a similar point in the BF algorithm, the user chooses whether or not to resample. If the user chooses not to resample, the estimator with nonuniform weights, $\tilde{\pi}^M_{0:t}(x_t)$, is used to sample from the forecast distribution of the next timestep; otherwise, the estimator with uniform weights, $\pi^M_{0:t}(x_t)$, is used. The same principle does not apply to the EnKF estimator $\tilde{\vartheta}^M_{0:t}(x_t)$: the convolution of the nonlinear mapping of the state transition model, i.e., $m_{t+1}$, with a Gaussian density does not have an analytic expression. Specifically, when using $\tilde{\vartheta}^M_{0:t}(x_{t-1})$ to generate samples from the forecast distribution at time $t+1$, we have

$$\begin{aligned} x_{t+1}^{f(m)} &\sim \int f(x_{t+1} \,|\, x_t) \tilde{\vartheta}^M_{0:t}(x_{t-1}) dx_t \\ &= \frac{1}{M}\sum_{m=1}^{M} \int \phi(x_{t+1}; m_t(x_t), U_{t+1}) \phi(x_t; \tilde{x}_t^{a(m)}, \hat{\Sigma}_t^a) dx_t. \end{aligned}$$

The integral in each summand generally does not have an analytic expression. A solution is to approximate the integral with a Monte Carlo sample, which is equivalent to resampling from $\tilde{\vartheta}^M_{0:t}(x_{t-1})$ to form an empirical density estimator of the analysis distribution.

Both the original EnKF and EnKF with perturbed observations have a step that is equivalent to a resampling step, using stratified or systematic sampling. Since the weights on each Gaussian in the estimator $\tilde{\vartheta}^M_{0:t}(x_t)$ are uniform, Theorem 2.4.1 implies that stratified/systematic sampling generates one sample from each Gaussian density in the mixture. However, instead of resampling from each Gaussian density with a variance of $\hat{\Sigma}^a_t$, the following approximations to $\hat{\Sigma}^a_t$ are made:

- **Original EnKF**: The bandwidth is approximated as $\hat{\Sigma}^a_t \approx 0$, thus the estimator of the analysis distribution is approximated as

$$\tilde{\vartheta}^M_{0:t}(x_t) \approx \tilde{\psi}^M_{0:t}(x_t) \equiv \frac{1}{M}\sum_{m=1}^{M} \delta_{\tilde{x}^{a(m)}_t}(x_t).$$

  Stratified or systematic random sampling is used to generate a simple random sample from $\tilde{\psi}^M_{0:t}(x_t)$, obtaining the analysis samples $x^{a(m)}_t = \tilde{x}^{a(m)}_t$ for each $m = 1, ..., M$.

- **EnKF with perturbed observations**: Recall that this particular method adds noise to the measurement $y_t$ (Equation (2.16)), generating analysis samples:

$$x^{a(m)}_t = x^{f(m)}_t + K_t(y_t - H_t x^{f(m)}_t) + K_t\epsilon^{(m)}_t,$$

  where $\epsilon^{(m)}_t \sim \mathcal{N}(0, V_t)$. This is equivalent to stratified or systematic resampling with uniform weights and approximating the analysis bandwidth $\hat{\Sigma}^a_t$ to be $\mathbb{C}\text{ov}(K_t\epsilon_t) = K_tV_tK^T_t$ (see Equation (2.16)). Therefore, the estimator of the analysis distribution is approximated as

$$\tilde{\vartheta}^M_{0:t}(x_t) \approx \tilde{\psi}^M_{0:t}(x_t) \equiv \frac{1}{M}\sum_{m=1}^{M} \phi(x_t; \tilde{x}^{a(m)}_t, K_tV_tK^T_t).$$

  Again, stratified or systematic random sampling is used to generate a simple random sample from $\tilde{\psi}^M_{0:t}(x_t)$, obtaining the analysis samples $x^{a(m)}_t \sim \mathcal{N}(\tilde{x}^{a(m)}_t, K_tV_tK^T_t)$ for each $m = 1, ..., M$.

Another estimator of the analysis distribution is constructed from analysis samples generated from either estimator:

$$\psi_{0:t}^{M}(x_t) = \frac{1}{M}\sum_{m=1}^{M}\delta_{x_t^{a(m)}}(x_t). \tag{3.8}$$

Since square-root filters deterministically scale the forecast samples $\{x_t^{f(m)}\}_{m=1}^{M}$, there is no analogous BF interpretation for square-root filters[1], but the same estimator $\psi_{0:t}^{M}(x_t)$ with point masses centered at the resulting analysis samples $\{x_t^{a(m)}\}_{m=1}^{M}$ is used to sample from the approximate forecast distribution of the next timestep[2]. As mentioned in Section 2.3.2, even though the EnKF algorithm was inspired by the EKF, the sample mean and variance $\hat{\mu}_t^a$ and $\hat{\Sigma}_t^a$ are not used when filtering the next timestep. Only the samples $\{x_t^{a(m)}\}_{m=1}^{M}$ are used and our re-interpretation makes it clear how.

### 3.1.1 Connection to the bootstrap filter: a summary

We summarize the similarities and differences between the EnKF and BF:

- **Forecast step**: Both the BF and EnKF generate forecast samples in the same way. However, the BF approximates the forecast distribution with an empirical density, whereas the EnKF approximates with mixture density of Gaussian densities.

- **Update step**: In the update step, the BF updates the weight of each forecast sample and does not change the forecast samples, whereas the EnKF approximates the weight to be uniform and moves the forecast samples towards the measurement. Similar to the forecast distribution, the BF uses an empirical density estimator and the EnKF uses a mixture density of Gaussian densities to approximate the analysis distribution.

- **Resample step**: Unlike the BF, the EnKF requires a resampling step and specifically stratified or systematic sampling is used. Instead of using the bandwidth $\hat{\Sigma}_t^a$, the EnKF approximates the bandwidth with a different

[1] The update step with the deterministic square-root filter is more closely related to the update step in sigma-point Kalman filters, where the samples are scaled to have the proper mean and variance (van der Merwe, 2004).

[2] We are not the first to recognize that the perturbed observations algorithm use both $\tilde{\psi}_{0:t}^{M}(x_t)$ and $\psi_{0:t}^{M}(x_t)$ as estimators of the analysis distribution (Frei and Künsch, 2012, 2013). We independently discovered these estimators through our probabilistic re-interpretation of the EnKF.

| *(a) Bootstrap filter* | *(b) Ensemble Kalman filter with perturbed observations* |
|---|---|
| 1. Forecast step: Sample<br>$$x_t^{f(m)} \sim f(x_t \mid x_{t-1}^{a(m)})$$<br>for each $m = 1, ..., M$. | 1. Forecast step: Sample<br>$$x_t^{f(m)} \sim f(x_t \mid x_{t-1}^{a(m)})$$<br>for each $m = 1, ..., M$. |
| 2. Observe $y_t$. | 2. Observe $y_t$. |
| 3. Update step: Calculate weights<br>$$l_t^{(m)} = \phi(y_t; H_t x_t^{f(m)}, V_t)$$<br>for each $m = 1, ..., M$.<br><br>Set<br>$$\tilde{x}_t^{a(m)} = x_t^{f(m)},$$<br>$$\tilde{w}_t^{(m)} = l_t^{(m)} / \sum_{n=1}^{M} l_t^{(n)},$$<br>for each $m = 1, ..., M$. | 3. Update step: Calculate the Kalman gain:<br>$$\hat{K}_t = \hat{\Sigma}_t^f H_t^T (V_t + H_t \hat{\Sigma}_t^f H_t^T)^{-1},$$<br>where $\hat{\Sigma}_t^f$ is the (tapered) sample covariance calculated from $\{x_t^{f(m)}\}_{m=1}^M$.<br>Set<br>$$\tilde{x}_t^{a(m)} = x_t^{f(m)} + \hat{K}_t (y_t - H_t x_t^{f(m)}),$$<br>$$w_t^{(m)} = 1/M,$$<br>for each $m = 1, ..., M$. |
| 4. Resample step: Sample<br>$$j_m = i \text{ with probability } w_t^{(i)},$$<br>$$x_t^{a(m)} = \tilde{x}_t^{a(j_m)},$$<br>for each $m = 1, ..., M$. | 4. Resample step: Sample<br>$$x_t^{a(m)} \sim \mathcal{N}(\tilde{x}_t^{a(m)}, B_t^a)$$<br>for each $m = 1, ..., M$ with $B_t^a = K_t V_t K_t^T$. |

Algorithm 3.1: *Bootstrap filter versus ensemble Kalman filter: a comparison.* For each time $t$, these algorithms require samples $\{x_{t-1}^{a(m)}\}_{m=1}^M \dot{\sim} p(x_{t-1} \mid y_{0:t-1})$. To directly compare the two algorithms, resampling after every filtering step is required of the bootstrap filter and the variance inflation step has been removed from the ensemble Kalman filter.

bandwidth. Both the BF and EnKF approximate the analysis distribution with an empirical density after the resampling the particles.

Algorithm 3.1 compares the essential steps of both algorithms side-by-side; the presentation of the algorithms are slightly different from their initial expositions to allow for direct comparisons, i.e., requiring resampling after every filtering step in the BF and omitting the variance inflation step in the EnKF.

## 3.2 Pre-regularized particle filter with Gaussian kernels: the bridge between the ensemble Kalman filter and the bootstrap filter

The first difference between the BF and the EnKF is the estimators used to approximate the forecast distribution: the BF uses an empirical density estimator and the EnKF uses a KDE with Gaussian kernels. If the variance of a Gaussian density is taken to zero, it becomes a point mass centered at the mean. Therefore, the EnKF estimator for the forecast distribution is equivalent to the BF estimator when the bandwidth is taken to be zero instead of the sample variance. Considering how well the EnKF performs in practice for high-dimensional state-spaces, it seems beneficial to have nonzero covariances.

To better understand this benefit, we draw intuition from the KDE literature (see Silverman (1986) and Wand and Jones (1995) for textbook reviews). Instead of using an empirical density as an estimator of a probability density, KDE uses a mixture of kernels centered at independently and identically distributed (iid) samples $x_1, ..., x_n \sim p(x)$:

$$\hat{p}_B(x) \equiv \frac{1}{n}\sum_{i=1}^{n} |B|^{-1/2} K\left[B^{-1/2}(x - x_i)\right],$$

where $B$ is a smoothing parameter, often called a *bandwidth*. For $\hat{p}_B(x)$ to be an appropriate estimator of a probability density, the kernel $K$ must satisfy the following constraints: (1) $K(u) \geq 0$ for all $u$, (2) $\int K(u)du = 1$, and (3) $K(-u) = K(u)$. It is easy to show that the standard Gaussian density satisfies the constraints of a kernel.

As the bandwidths approach zero, kernel density estimators (KDEs) are well-known to be a biased, but asymptotically optimal, estimator of $p(x)$ (Wand, 1992). Optimality in the KDE literature is often defined in terms of asymptotic properties of the integrated mean squared error, $\mathbb{E}[\int(\hat{p}_B(x) - p(x))^2 dx]$. If $x_1, ..., x_n$ are iid samples from $\mathcal{N}(\mu, \Sigma)$, the asymptotic integrated mean squared error is minimized by the following bandwidth:

$$B_n = c_{n,d}\Sigma \qquad \text{with} \qquad c_{n,d} = \left(\frac{4}{n(d+2)}\right)^{\frac{2}{d+4}},$$

where $d$ is the dimension of $x_i$. Notice that the bandwidth is a function of the variance of the density, which is not known in reality. The sample variance $\hat{\Sigma}$

is a reasonable estimator and is in fact the suggested bandwidth, after scaling it by $c_{n,d}$, for unimodal densities (Wand and Jones, 1995). Furthermore, notice that the bandwidth is scaled by both the dimension of $x$ ($d$) and the sample size $n$ ($c_{n,d}$). In fact, $c_{n,d} \to 0$ as $n \to \infty$, thus $B_n \to 0$. Therefore, the bandwidths should be scaled by the sample size in such a way that the kernels shrink to point masses as the number of samples increases.

With this intuition, it seems reasonable to approximate the forecast distribution with a mixture distribution as the EnKF does. The bandwidth choice of the sample variance is also reasonable but with one caveat: the bandwidth should scale with the ensemble size, similar to the scalar $c_{n,d}$ as in the KDE literature. In fact, the idea of using KDEs with the PF has been proposed numerous times in the literature to overcome particle collapse; these filters are known as *regularized particle filters (RPFs)*. When Gordon et al. (1993) first introduced the BF, they suggested jittering the forecast samples when filtering state-space models with a deterministic state transition model to overcome particle degeneracy. Hürzeler and Künsch (1998) then suggested a more principled approach to jittering: use a KDE as an estimator of the forecast distribution, leading to the introduction of *pre-regularized particle filters* (pre-RPF)[3]. Therefore, using a mixture of Gaussian densities as an estimator to the forecast distribution, as the EnKF does, is a pre-RPF with Gaussian kernels.

The unique combination of applying pre-RPF with Gaussian kernels to linear measurement models with Gaussian errors is amenable to closed-form expressions to the estimators, which is not true for many applications of the RPF. Like the PF, the RPF was developed for general state-space models such as the one outlined in Equation (1.1), thus analytic forms could not be derived for the resulting estimators and rejection sampling is performed instead. In the special case that we have presented, an analytic solution to the estimator is easily derived, e.g., $\vartheta^M_{1:t}(x_t)$, and thus allows for straightforward and efficient sampling from the estimator, important for the resampling step.

In this section, we explore the pre-RPF with Gaussian kernels applied to the state-space model outlined at the beginning of the chapter. Like the EnKF, the forecast distribution is approximated with a mixture of Gaussian kernels with a bandwidth that scales by the number of particles. We first present the estimators of the optimal filtering densities, deferring the bandwidth choice to Section 3.2.1. The filter does not include the approximations made in the update and resampling steps of the EnKF. We make some hypotheses regarding the roles of these approximations in the success of the EnKF and discuss

[3]On the other hand, post-regularized particle filters use a KDE to approximate the *analysis* distribution.

**Inputs**: $\{x_{t-1}^{a(m)}\}_{m=1}^{M} \dot{\sim} p(x_{t-1} \,|\, y_{1:t-1})$
**Output**:
- $\{x_t^{f(m)}\}_{m=1}^{M} \dot{\sim} p(x_t \,|\, y_{1:t-1})$
- $\{x_t^{a(m)}\}_{m=1}^{M} \dot{\sim} p(x_t \,|\, y_{1:t})$

1. Forecast step: Sample $x_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)})$ for each $m = 1, ..., M$.

2. Update step:

   a) Calculate the Kalman gain:

   $$K_t = B_t^f H_t^T (V_t + H_t B_t^f H_t^T)^{-1}.$$

   and

   $$l_t^{(m)} = \phi(y_t; H_t x_t^{f(m)}, V_t + H_t B_t^f H_t^T)$$

   for each $m = 1, ..., M$.

   b) Calculate

   $$\tilde{x}_t^{a(m)} = x_t^{f(m)} + K_t[y_t - H_t x_t^{f(m)}],$$
   $$\tilde{w}_t^{(m)} = l_t^{(m)} / \textstyle\sum_{n=1}^{M} l_t^{(n)},$$

   for each $m = 1, ..., M$.

3. Resampling step:

   a) Calculate $B_t^a = B_t^f - K_t H_t B_t^f$.

   b) Resample

   $$j_m = i \text{ with probability } \tilde{w}_t^{(i)},$$
   $$x_t^{a(m)} \sim \mathcal{N}(\tilde{x}_t^{(j_m)}, B_t^a),$$

   for each $m = 1, ..., M$.

Algorithm 3.2: *Pre-regularized particle filter with Gaussian kernels.*

further exploration in Section 3.3. Since the derivation of the estimators to the optimal filtering densities is quite similar to the derivations in the last section on re-interpreting the EnKF, we directly state the estimators without deriving them; for more detailed derivations, refer to Section 3.1.

Suppose we have generated simple random samples from the approximate

forecast distribution: $\{x_t^{f(m)}\}_{m=1}^M \dot{\sim} X_t \,|\, y_{1:t-1}$. As with the EnKF, the forecast distribution is approximated by a KDE with Gaussian kernels centered at the forecast samples and forecast bandwidth $B_t^f$:

$$\nu_{1:t-1}^M(x_t) \equiv \frac{1}{M}\sum_{m=1}^M \phi(x_t; x_t^{f(m)}, B_t^f).$$

After observing $y_t$, this estimator is plugged into the optimal predictive likelihood and analysis distribution, deriving estimators to those densities. The estimator to the predictive likelihood is

$$\nu_{1:t-1}^M(y_t) \equiv \int g(y_t \,|\, x_t)\nu_{1:t-1}^M(x_t)dx_t = \frac{1}{M}\sum_{m=1}^M l_t^{(m)}$$

with $l_t^{(m)} \equiv \frac{1}{M}\phi(y_t; H_t x_t^{f(m)}, V_t + H_t B_t^f H_t^T)$. Then, an estimator of the analysis distribution is derived:

$$\tilde{\nu}_{1:t}^M(x_t) \equiv \frac{\phi(y_t; H_t x_t, V_t)\nu_{1:t-1}^M(x_t)}{\nu_{1:t-1}^M(y_t)} = \sum_{m=1}^M \tilde{w}_t^{(m)}\phi(x_t; \tilde{x}_t^{a(m)}, B_t^a), \tag{3.9}$$

with updated weights $\tilde{w}_t^{(m)} = l_t^{(m)} / \sum_{n=1}^M l_t^{(n)}$ and updated centers and analysis bandwidths as follows:

$$\tilde{x}_t^{a(m)} = x_t^{f(m)} + \hat{\Sigma}_t^f H_t^T (V_t + H_t B_t^f H_t^T)^{-1}(y_t - H_t x_t^{f(m)}), \tag{3.10a}$$

$$B_t^a = B_t^f - B_t^f H_t^T (V_t + H_t B_t^f H_t^T)^{-1} H_t B_t^f, \tag{3.10b}$$

for each $m = 1, ..., M$.

Like the EnKF, resampling must be performed to obtain an empirical density estimator required in the next forecast step. In particular, a simple random sample $\{x_t^{a(m)}\}_{m=1}^M \sim \tilde{\nu}_{1:t}^M(x_t)$ is generated via any resampling algorithm outlined in Section 2.4.1. These new samples provide another approximation to the analysis distribution:

$$\nu_{1:t}^M(x_t) = \frac{1}{M}\sum_{m=1}^M \delta_{x_t^{a(m)}}(x_t).$$

Algorithm 3.2 summarizes the algorithm. In Algorithm 3.3, we repeat Algorithm 3.1 and add the pre-RPF between the BF and EnKF algorithms to provide a side-by-side comparison of the three algorithms applied to the state-space model outlined at the beginning of this chapter. Figure 3.1 illustrates the comparisons.

*(a) Bootstrap filter*

1. Forecast step: Sample
$$x_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)})$$
for each $m = 1, ..., M$.
2. Observe $y_t$.
3. Update step: Calculate weights
$$l_t^{(m)} = \phi(y_t; H_t x_t^{f(m)}, V_t)$$
for each $m = 1, ..., M$.
4. Resample step: Sample
$$j_m = i \text{ with probability } \tilde{w}_t^{(i)},$$
$$x_t^{a(m)} = \tilde{x}_t^{a(j_m)},$$
for each $m = 1, ..., M$.

*(b) Pre-regularized particle filter with Gaussian kernels*

1. Forecast step: Sample
$$x_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)})$$
for each $m = 1, ..., M$.
2. Observe $y_t$.
3. Update step: Choose bandwidth matrix $B_t^f$ and calculate the Kalman gain:
$$K_t = B_t^f H_t^T (V_t + H_t B_t^f H_t^T)^{-1}.$$
Calculate weights
$$l_t^{(m)} = \phi(y_t; H_t x_t^{f(m)}, V_t + H_t B_t^f H_t^T)$$
for each $m = 1, ..., M$. Set
$$\tilde{x}_t^{a(m)} = x_t^{f(m)} + K_t(y_t - H_t x_t^{f(m)}),$$
$$\tilde{w}_t^{(m)} = l_t^{(m)} / \textstyle\sum_{n=1}^M l_t^{(n)},$$
for each $m = 1, ..., M$.
4. Resample step: Sample
$$j_m = i \text{ with probability } \tilde{w}_t^{(i)},$$
$$x_t^{a(m)} \sim \mathcal{N}(\tilde{x}_t^{a(j_m)}, B_t^a),$$
for each $m = 1, ..., M$ with $B_t^a = B_t^f - K_t H_t B_t^f$.

*(c) Ensemble Kalman filter with perturbed observations*

1. Forecast step: Sample
$$x_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)})$$
for each $m = 1, ..., M$.
2. Observe $y_t$.
3. Update step: Calculate the Kalman gain:
$$\hat{K}_t = \hat{\Sigma}_t^f H_t^T (V_t + H_t \hat{\Sigma}_t^f H_t^T)^{-1},$$
where $\hat{\Sigma}_t^f$ is the (tapered) sample covariance calculated from $\{x_t^{f(m)}\}_{m=1}^M$.
Set
$$\tilde{x}_t^{a(m)} = x_t^{f(m)} + \hat{K}_t(y_t - H_t x_t^{f(m)}),$$
$$w_t^{(m)} = 1/M,$$
for each $m = 1, ..., M$.
4. Resample step: Sample
$$x_t^{a(m)} \sim \mathcal{N}(\tilde{x}_t^{a(m)}, B_t^a)$$
for each $m = 1, ..., M$ with $B_t^a = \hat{K}_t V_t \hat{K}_t^T$.

Algorithm 3.3: *Bootstrap filter versus pre-regularized particle filter versus ensemble Kalman filter: a comparison.* This figure is the same as Algorithm 3.1 with the pre-RPF algorithm added between the bootstrap and ensemble Kalman filter for comparison. For each time $t$, these algorithms require samples $\{x_{t-1}^{a(m)}\}_{m=1}^M \sim p(x_{t-1} \,|\, y_{0:t-1})$. To directly compare the algorithms, resampling after every filtering step is required of the bootstrap filter and the variance inflation step has been removed from the ensemble Kalman filter.

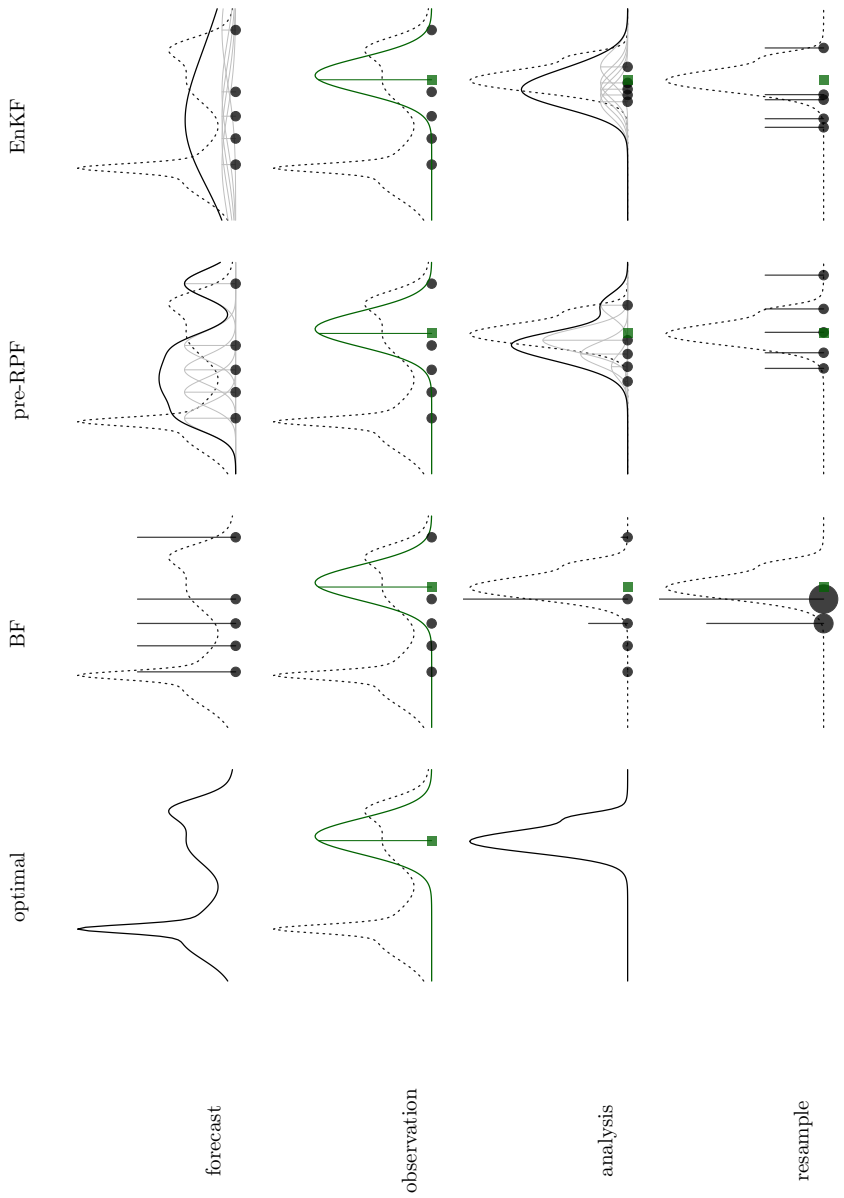



Figure 3.1: *Bootstrap filter versus pre-regularized particle filter versus ensemble Kalman filter: an illustration.* The "optimal" column represents the optimal filtering densities. The optimal forecast and analysis distributions are shown in the first and last two rows, respectively. The measurement density is represented by the green line with the measurement represented by the green square. Black points represent the forecast or analysis samples and the size of the black point represents the number of samples. The vertical line above the black points represents a point mass centered at a sample; its height represents its weight. The gray line above the black points represents the Gaussian kernels. The black lines in the "pre-RPF" and "EnKF" columns represent the estimators of the forecast or analysis distribution.

### 3.2.1 Bandwidth selection

The choice of the forecast bandwidth $B_t^f$ is crucial to the success of the RPF. If the bandwidth is too small, the RPF collapses to the PF and particle collapse becomes an issue—the very problem that the RPF was developed to avoid. If the bandwidth is too large, the estimator is a poor approximation of the optimal filtering densities. Since the development of the RPF, a few papers have analyzed large sample properties of the RPF with fixed bandwidths (Crisan and Miguez, 2014; Le Gland and Oudjane, 2004; Le Gland et al., 1998). The analyses suggest a forecast bandwidth of

$$B_t^f = M^{-1/(d_x+4)}\Sigma, \tag{3.11}$$

where $\Sigma$ is a fixed covariance matrix. Therefore, as with KDEs, the bandwidth is dependent on both the dimension of the state $x_t$ ($d_x$) and the sample size $M$. Unfortunately, because optimality is defined differently in the two literatures, the method to find an optimal bandwidth in the KDE literature cannot be applied to find an optimal forecast bandwidth $B_t^f$, even in specific cases.

## 3.3 Future exploration

In this section, we introduced a re-interpretation of the EnKF. Not only is the re-interpretation useful in the development of a new parameter estimation methodology that will be introduced in Part II, it clarifies the connection between the EnKF and the PF. We believe that the EnKF can be more rigorously studied by borrowing some of the theoretical underpinnings of the PF. In particular, the pre-RPF with Gaussian kernels is a step in that direction: like the EnKF, the pre-RPF also uses a mixture of Gaussian kernels as an estimator of the forecast distribution. We conclude this section with a few hypotheses that we think are critical to the practical success of the EnKF and are worth exploring more rigorously in the future.

Two special cases of the pre-RPF with Gaussian kernels have appeared in the atmospheric science literature: both propose two different forecast bandwidths $B_t^f$. The kernel filter (Anderson and Anderson, 1999) uses the (tapered) sample variance $\hat{\Sigma}_t^f$ calculated with the forecast samples $\{x_t^{f(m)}\}_{m=1}^M$ as the bandwidth $B_t^f$. The nonlinear ensemble filter (Bengtsson et al., 2003) uses a nearest neighbor approach to estimate a different bandwidth for each kernel in the mixture, again with the forecast samples $\{x_t^{f(m)}\}_{m=1}^M$. Neither scaled the covariance by the ensemble size. Both methods were demonstrated to perform better than the EnKF when applied to low-dimensional state-spaces ($d_x \leq 40$)

but had trouble with particle collapse when applied to higher-dimensional state-spaces (verbal discussion with authors). In our own explorations with the Lorenz 2005 system with $d_x = 960$, we found somewhat similar results: the pre-RPF with a simple fixed bandwidth (Equation (3.11) with $\Sigma = I_{d_x}$ and multiplied by a fixed scalar $c$) is just as effective as or better than the EnKF when filtering fully observed systems, but suffered from particle collapse when filtering partially observed systems.

Our explorations of the forecast bandwidth lead us to believe that the KDE of the forecast distribution is important but not critical to the success of the EnKF. Instead, we suspect the critical success comes from the combination of the variance inflation in the forecast step and the weight approximation in the update step. Variance inflation increases the variance of the ensemble to better explore the state space, but this is not enough to prevent particle collapse—it is actually the weight approximation introduced in the update step of the EnKF that prevents particle collapse. Recall that the EnKF estimator $\vartheta^M_{0:t}(x_t)$ from Equation (3.4)[4] has two modes to update the particles: the weight $\tilde{w}_t^{(m)}$ and the conditional update of the Gaussian kernel with the measurement[5] $y_t$. Since the EnKF approximates the weights $\tilde{w}_t^{(m)}$ to be uniform (see Equation (3.6)), the particles only gets updated through one of the two update modes: the conditional update. It seems that the weight approximation increases the effective sample size in exchange for the introduction of biasedness of the estimator. Without the weight approximation, however, particle collapse becomes inevitable in high-dimensional state-spaces, thus rendering variance inflation ineffective. Since large sample asymptotics of the EnKF with linear state transitions have been previously studied (Le Gland et al., 2009; Mandel et al., 2009), we have good reason to believe that the weight approximation may be appropriate for near-linear state transitions and therefore deviations from linearity should be studied in conjunction with variance inflation and, more importantly, the weight approximation. To our knowledge, no theoretical analyses of the EnKF acknowledges the weight approximation and many acknowledge, but leave out, variance inflation, even though it is must be added whenever the EnKF is applied to high-dimensional state-spaces; we believe these analyses are missing a crucial part of the puzzle.

For completeness, we discuss one last difference between the pre-RPF and the EnKF that is not as important in the practical success of the EnKF. Recall that the analysis bandwidth $B_t^a$ in the pre-RPF algorithm (see Equa-

[4]This estimator corresponds to the pre-RPF estimator $\tilde{\nu}^M_{0:t}(x_t)$ of Equation (3.9).

[5]This corresponds to the pre-RPF's Equation (3.10a).

tion (3.10b)) is

$$B_t^a = B_t^f - B_t^f H_t^T (V_t + H_t B_t^f H_t^T)^{-1} H_t B_t^f = (I - K_t H_t) B_t^f,$$

where the Kalman gain $K_t = B_t^f H_t^T (V_t + H_t B_t^f H_t^T)^{-1}$ is substituted into the second equality. When the bandwidth $B_t^f$ is substituted with the sample variance $\hat{\Sigma}_t^f$, the analysis bandwidth $B_t^a$ becomes

$$B_t^a = (I - K_t H_t) \hat{\Sigma}_t^f.$$

The EnKF with perturbed observations uses the following analysis bandwidth instead of $B_t^a$:

$$K_t V_t K_t^T = K_t (I - K_t^T H_t^T) \hat{\Sigma}_t^f,$$

where the Woodbury matrix identity is applied to obtain the equality. The difference between the two expressions is

$$B_t^a - K_t V_t K_t^T = [\underbrace{(I - K_t H_t) - K_t (I - K_t^T H_t^T)}] \hat{\Sigma}_t^f.$$

Therefore, the analysis step of the pre-RPF and EnKF are asymptotically equivalent if the expression above the underbrace is asymptotically zero. It may be interesting to explore the large sample asymptotics of that particular term.

# Part II

# Parameter estimation

This page
intentionally
left blank

# 4 Parameter estimation: a short review and discussion of issues

In the previous part of the dissertation, we examined filtering when the model and its parameters are known. Though the models may be reasonable for a variety of situations, there are often unknown parameters governing the systems. While one can use scientific expertise to set parameter values, it is better to let the data speak: the parameters can be estimated with collected measurements. Unfortunately, parameter estimation with state-space models is not simple: with the exception of the state-space model under the Kalman filter, the likelihood cannot be evaluated exactly and must be approximated with filtering algorithms. For this reason, parameter estimation methods that have been developed for state-space models are algorithm-specific. Many parameter estimation methods have been developed for the PF, but the methods developed with the EnKF estimate parameters with specific forms. The novel re-interpretation of the EnKF outlined in Section 3.1 facilitates connecting the abundant methods that have been developed for PFs to the EnKF. With computational expense in mind, we develop a methodology in this part of the dissertation for parameter estimation with the EnKF that estimate any parameter of applicable state-space models.

## 4.1 State-space models with unknown parameters

We introduce unknown parameters to the state-space model presented in the treatment of the EnKF (Equation (2.12)). In particular, we introduce an unknown state parameter $\theta_x \in \mathbb{R}^{d_{\theta_x}}$ to the state transition model:

$$x_t = m_t(x_{t-1}, \theta_x) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, U_t(\theta_x)) \tag{4.2a}$$

and an unknown measurement parameter $\theta_y \in \mathbb{R}^{d_{\theta_y}}$ to the measurement model:

$$y_t = H_t(\theta_y)x_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, V_t(\theta_y)). \tag{4.2b}$$

Notice that $m_t : \mathbb{R}^{d_x + d_{\theta_x}} \to \mathbb{R}^{d_x}$ is a nonlinear mapping of the previous state $x_{t-1}$ and state parameter $\theta_x$ and $H_t(\cdot)$ is a $d_y \times d_x$ matrix that depends on the parameter $\theta_y$. Furthermore, the variance of the state disturbance, $U_t(\cdot)$, and measurement noise, $V_t(\cdot)$, also depend on the parameters $\theta_x$ and $\theta_y$, respectively. Let these two parameters be denoted by $\theta = (\theta_x, \theta_y)$ with dimension $d_\theta = d_{\theta_x} + d_{\theta_y}$. The parameter $\theta$ is not time-varying and thus it is called a *static* parameter. Like the state-space model under the EnKF, the state transition and measurement densities are Gaussian but are now functions of state *and* parameter: the state transition density is

$$f(x_t \,|\, x_{t-1}, \theta_x) = \phi(x_t \,|\, m_t(x_{t-1}, \theta_x), U_t(\theta_x))$$

and the measurement density is

$$g(y_t \,|\, x_t, \theta_y) = \ \phi(y_t \,|\, H_t(\theta_y)x_t, V_t(\theta_x)).$$

The same assumptions outlined in Chapter 1 apply to this state-space model. The goal is to estimate the parameter $\theta$, but that cannot be done without filtering, as the reader will notice from the form of the likelihood.

A crucial component for parameter estimation is the *likelihood* of the measurements collected up to time $T$, i.e., $p(y_{1:T} \,|\, \theta)$. A recursive application of the axiom of conditional probability provides the following form of the likelihood:

$$p(y_{1:T} \,|\, \theta) = \prod_{t=1}^{T} p(y_t \,|\, y_{0:t-1}, \theta).$$

This shows that the likelihood is a function of the predictive likelihoods $p(y_t \,|\, y_{0:t-1}, \theta)$ up to time $T$ and is thus easily calculated as long as the predictive likelihood is easily evaluated. However, recall from the discussion of filtering algorithms in Part I that the predictive likelihood cannot be evaluated exactly with the exception of the Kalman filter and must be approximated instead. For example, the EKF approximates the predictive likelihood with Gaussian density (see Equation (2.10)) and both the PF and EnKF approximate the predictive likelihood with a Monte Carlo estimate (see Equations (2.20) and (3.3), respectively). Thus, the likelihood is subject to the approximation errors of the filtering algorithm, which adds to the difficulty of parameter estimation with state-space models. In particular, with Monte

Carlo sampling methods, such as the PF and the EnKF, the Monte Carlo variance of the likelihood approximation is reduced with more samples.

In the next section, we provide a brief review of parameter estimation algorithms, with a particular focus on those that can be applied with the PF and EnKF. As the reader will notice, the algorithms are often specific to the filtering algorithm, and our goal is to bridge the gap using the connection from Section 3.1.

## 4.2 Limitations of current parameter estimation methods with high-dimensional state-spaces

There are a host of algorithms to estimate static parameters with PFs. The methods span both maximum likelihood and Bayesian approaches, including online and offline approaches. However, as with filtering, many methods suffer from particle degeneracy, even for low-dimensional systems[1]. Therefore, PFs are impractical for atmospheric science applications, which often have high-dimensional state-spaces. On the other hand, though the EnKF has been demonstrated to be effective at filtering high-dimensional state-spaces, parameter estimation is less developed than in the PF literature. Since the introduction of the EnKF, parameter estimation with the EnKF is largely performed with state augmentation: the parameter is "augmented" to the state to create a new state variable that gets updated with new data (Anderson, 2001; Dee and da Silva, 1998; Yang and Delsole, 2009; Zupanski and Zupanski, 2006); this particular parameter estimation method is called *state augmentation*. In fact, Evensen (2009) dedicates a chapter to parameter estimation using state augmentation with the EnKF.

While state augmentation has worked well in many situations, it is not sufficient to deal with all parameters. Stroud and Bengtsson (2007) first recognized this issue when estimating the scale parameter of the measurement noise, i.e., the parameter $\sigma$ in $\epsilon_t \sim \mathcal{N}(0, \sigma V_t)$. The authors stressed that state augmentation cannot be used to estimate this parameter because EnKF is a linear updating rule: the update of parameters under state augmentation relies on the correlation between the state and parameters, which can be problematic when the states and parameters are not linearly correlated (or near linear). Both Delsole and Yang (2010) and Frei and Künsch (2012) further elaborated

[1]For a comprehensive review of parameter estimation with PFs and the particle degeneracy issues, we refer the reader to Kantas et al. (2015).

on this limitation of state augmentation and developed algorithms to estimate parameters other than $\sigma$. However, both algorithms cannot be generally applied to estimate any parameter $\theta$ in the state-space model of Equation (4.2): the algorithm developed by DelSole and Hou estimates a parameter that controls a Wiener process in the continuous state-transition model and the algorithm developed by Frei and Künsch estimates measurement noise parameter $\rho$ of $\epsilon_t \sim \mathcal{N}(0, V_t(\rho))$ but cannot estimate other parameters, such as parameters of the state transition model or other parameters governing the measurement model.

Given that the EnKF is effective at filtering many high-dimensional applications, it is desirable to borrow the abundant parameter estimation techniques from the PF literature. At first glance, many of the parameter estimation techniques seem to apply to the EnKF because of its similarities with PFs (as outlined in Section 3.1) with the added benefit of that particle degeneracy is less of an issue with the EnKF than with PFs. In fact, this already happens: state augmentation is a specific case of *artificial evolution of parameters*, a parameter estimation technique used with PFs to overcome particle degeneracy issues. However, upon closer inspection, there are very few parameter estimation algorithms from the PF literature that can be directly applied to the EnKF for a couple of practical reasons.

First, there is often not a closed-form expression for the first moment of the state transition density. As mentioned in Chapter 1, the EnKF is often applied to state-space models with mathematical models as state transition models, thus the first moment and the derivatives of the state transition density are analytically unattainable. Approximations can be made, but that introduces error into the already difficult task estimating the likelihood, especially for high-dimensional state-spaces. For this reason, many parameter estimation techniques that require the state transition density are eliminated. This includes many of the maximum likelihood approaches that require evaluations of the gradient of the state transition density, such as EM algorithms, and many of the more efficient Markov chain Monte Carlo (MCMC) algorithms developed for PFs (Andrieu et al., 2010), such as particle Gibbs sampling. To our knowledge, this leaves two classes of parameter estimation algorithms: particle marginal Metropolis-Hastings (PMMH) and methods that use artificial evolution of parameters.

Many atmospheric scientists mainly filter with the EnKF as opposed to PFs because of its ability to filter high-dimensional state-spaces. Therefore, any parameter estimation method applied with EnKF must be cognizant of the computational expense. For this reason, this eliminates PMMH as a reasonable parameter estimation technique, leaving artificial evolution of parameters

as the best parameter estimation technique that can be borrowed from the PF literature. However, as mentioned earlier in the discussion of state augmentation, the technique of artificial evolution of parameters with the EnKF cannot generally be applied to estimate any parameter, an issue we expound upon in Section 4.3. Because of these shortcomings, we develop a combined EnKF and PF approach in Chapter 5 that uses the EnKF to filter the often high-dimensional states and the PF to filter the lower-dimensional static parameters. The algorithm can easily be plugged into parameter estimation algorithms from the PF literature that use artificial evolution of parameters. Artificial evolution of parameters is used in both Bayesian inference (Kitagawa, 1998; Liu and West, 2001) and estimation of a maximum likelihood estimator (MLE) (Ionides et al., 2006, 2015).

Before proposing our algorithm, we review the method of artificial evolution of parameters, elaborate on its inability to estimate any static parameters when applied with the EnKF, and demonstrate its limitations on the Lorenz 1963 system.

## 4.3 Artificial evolution of parameters

With artificial evolution of parameters, the parameter $\theta$ is replaced with a closely related time-varying parameter $\theta_t \equiv (\theta_{t,x}, \theta_{t,y})$ and the state transition and measurement densities are replaced with closely related densities

$$f(x_t \,|\, x_{t-1}, \theta_{t,x}) \text{ and } g(y_t \,|\, x_t, \theta_{t,y})$$

with $\theta_t = (\theta_{t,x}, \theta_{t,y})$. The parameter $\Theta_t$ is a random walk as follows

$$\theta_t = \theta_{t-1} + \zeta_t \text{ with density } q(\theta_t \,|\, \theta_{t-1}) \tag{4.3a}$$

$$\mathbb{V}\text{ar}(\zeta_0) = \tau^2 \Psi \text{ and } \mathbb{V}\text{ar}(\zeta_t) = \sigma^2 \Psi \text{ for } t > 0 \tag{4.3b}$$

where $\tau^2$ is a positive scalar and $\Psi$ is a $d_\theta \times d_\theta$ matrix, both pre-specified by the user, and $\theta_0$ is an initial guess of the parameter. Of course, the static parameter is not truly time-varying, hence the "artificial" descriptor in its name[2]. In fact, $\sigma^2$ is zero in state augmentation—the idea being that the parameter is static, thus should be a priori constant in time. When artificial evolution of parameters is used to estimate an MLE (Ionides et al., 2006, 2015), $\sigma^2$ is a scalar that exponentially decreases over multiple iterations of the filter. If artificial evolution of parameters is used directly, $\sigma^2$ is a scalar chosen by

[2]In fact, Kantas et al. (2015) mention that the artificial perturbation may introduce biases into the estimation of parameters that is difficult to quantify in real-world examples.

the user. The density is called the *perturbation density* and is denoted by $q(\theta_t \,|\, \theta_{t-1})$. Typically the density $q$ is a multivariate normal distribution, but any distribution that satisfies the above constraints can be used, such as the truncated normal distribution. Alternatively, parameters can be transformed to allow the usage of a multivariate normal distribution.

Since the time-varying parameters are also assumed to follow a Markov process like the state, we can obtain an equivalent reparametrization of the state-space model by creating a new state variable $Z_t$ that contains both the state $X_t$ and parameter $\theta_t$:

$$z_t = \begin{bmatrix} x_t \\ \theta_t \end{bmatrix};$$

this is called "augmenting" the state. This leads to the augmented state-space model:

$$\begin{aligned} z_t &= \begin{bmatrix} m(x_{t-1}, \theta_t) \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \eta_t \\ \zeta_t \end{bmatrix}, \\ y_t &= \tilde{H}_t(\theta_t) z_t + \epsilon_t, \end{aligned} \tag{4.4}$$

where $\tilde{H}_t(\theta_t) = \begin{bmatrix} H_t(\theta_t) & 0_{d_\theta} \end{bmatrix}$, $0_{d_\theta}$ is the zero $d_\theta$-vector, and $\eta_t$, $\zeta_t$, and $\epsilon_t$ are distributed as before. The assumptions behind the state-space model is the same as the state-space model introduced in the previous part of the dissertation, hence the same graphical model in Figure 1.1 applies after replacing the nodes with $X_t$ with $Z_t$. For this reason, many developments in the previous part of the dissertation are directly applicable to the augmented state-space model—a nice feature called *plug-and-play* by Ionides et al. (2006, 2011) that is maintained in our proposed methodology. For completeness, we provide two graphical models for this state space model in Figure 4.1.

For the augmented state-space model to be complete, we derive the state transition and measurement densities of the model, which are used to obtain the optimal filtering densities and predictive likelihood. By the axiom of conditional probability, the state transition density is

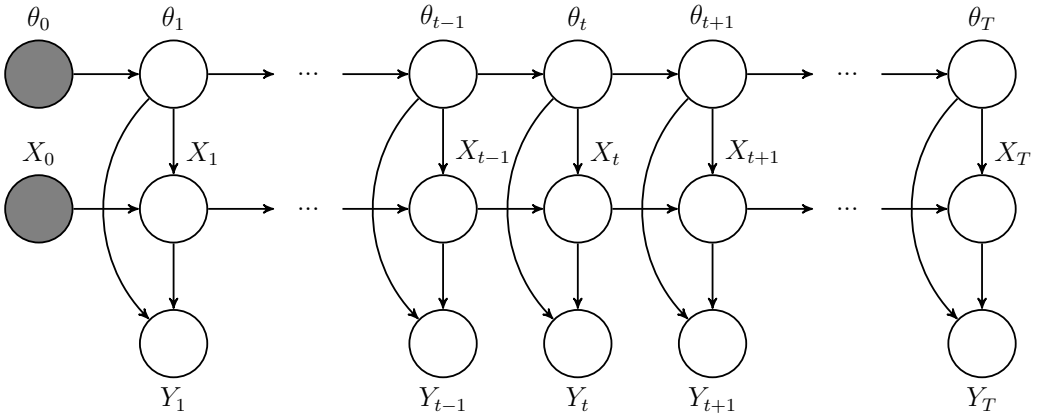
$$f(z_t \,|\, z_{t-1}) = p(x_t, \theta_t \,|\, x_{t-1}, \theta_{t-1}) = p(x_t \,|\, x_{t-1}, \theta_t, \theta_{t-1}) p(\theta_t \,|\, x_{t-1}, \theta_{t-1}).$$

By the Markov assumptions on both the state and parameter,

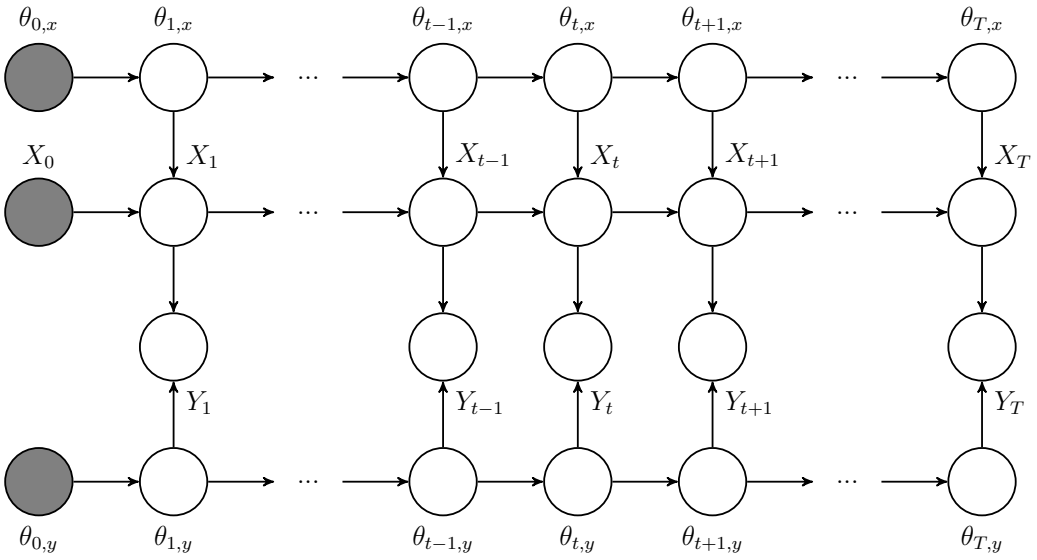
$$\begin{aligned} f(z_t \,|\, z_{t-1}) &= f(x_t \,|\, x_{t-1}, \theta_{t,x}) q(\theta_t \,|\, \theta_{t-1}) \\ &= \phi[x_t; m_t(x_{t-1}, \theta_{t,x}), U_t(\theta_{t,x})] q(\theta_t \,|\, \theta_{t-1}). \end{aligned}$$

The measurement density is

$$g(y_t \,|\, z_t) = g(y_t \,|\, x_t, \theta_{t,y}) = \phi[y_t; H_t(\theta_{t,y}) x_t, V_t(\theta_{t,y})] \tag{4.5}$$




(a) *Single node for the parameters.*




(b) *Two separate nodes for the state and observation parameters.*

Figure 4.1: *State-space model under artificial evolution of parameters: a graphical model*

Equipped with the state transition and measurement densities, the optimal filtering densities are easily derived by plugging them into the optimal filtering densities of Equation (2.1):

$$\begin{aligned} &p(z_t \mid y_{0:t-1}) \\ &\quad = \int\int \phi[x_t; m_t(x_{t-1}, \theta_{t,x}), U_t(\theta_{t,x})] q(\theta_t \mid \theta_{t-1}) p(x_{t-1}, \theta_{t-1} \mid y_{0:t-1}) dx_t d\theta_t, \end{aligned} \tag{4.6a}$$

$$\begin{aligned} &p(z_t \mid y_{0:t}) \\ &\quad = \frac{g(y_t \mid x_t, \theta_{t,y}) p(x_t, \theta_t \mid y_{0:t-1})}{p(y_t \mid y_{0:t-1})}, \\ &\quad = \frac{\phi[y_t; H_t(\theta_{t,y}) x_t, V_t(\theta_{t,y})] p(x_t, \theta_t \mid y_{0:t-1})}{p(y_t \mid y_{0:t-1})}, \end{aligned} \tag{4.6b}$$

$$\begin{aligned} &p(y_t \mid y_{0:t-1}) \\ &\quad = \int\int \phi[y_t; H_t(\theta_{t,y}) x_t, V_t(\theta_{t,y})] p(x_t, \theta_t \mid y_{0:t-1}) dx_t d\theta_t. \end{aligned} \tag{4.6c}$$

When convenient, we use a similar shorthand for the optimal filtering densities as in the previous part of the dissertation:

$$\begin{aligned} \pi_{s:t}(a) &\equiv p(a \mid y_{s:t}), \\ \pi_{s:t}(a|b) &\equiv p(a \mid b, y_{s:t}), \end{aligned}$$

where $a, b \in \{x_1, ..., x_T, y_0, ..., y_T\}$ and $s, t \in \{0, 1, .., T\}$. Therefore, the optimal filtering densities from Equation (4.6) are denoted as

$$\begin{aligned} \pi_{0:t-1}(z_t) &= p(z_t \mid y_{0:t-1}), \\ \pi_{0:t}(z_t) &= p(z_t \mid y_{0:t}), \\ \pi_{0:t-1}(y_t) &= p(y_t \mid y_{0:t-1}), \end{aligned}$$

respectively.

Not only are the optimal filtering densities easily derived for the augmented state-space model, parameters are estimated using almost any filtering algorithm that can filter the particular state-space model in Equation (4.4). Of the filtering algorithms presented in the dissertation, the PF is the most flexible filtering algorithm and can easily be applied to augmented state-space model. In fact, many parameter estimation algorithms take this approach with the PF (Ionides et al., 2006, 2011; Liu and West, 2001). At first glance, it seems that the EnKF is also applicable, but upon closer inspection, $\theta_t$ is not always updated by measurements for a few reasons described in the next section.

### 4.3.1 Parameter estimation with the ensemble Kalman filter: shortcomings

Before describing the shortcomings, we examine how the EnKF updates the parameter. Suppose we have samples of the augmented state:

$$z_{t-1}^{a(m)} \equiv (x_{t-1}^{a(m)}, \theta_{t-1}^{a(m)}) \dot{\sim} p(z_{t-1} \,|\, y_{0:t-1}).$$

At time $t$, generate augmented state samples $z_t^{f(m)} \equiv (x_t^{f(m)}, \theta_t^{f(m)})$ from the forecast distribution:

$$\theta_t^{f(m)} \sim q(\theta_t \,|\, \theta_{t-1}^{a(m)}),$$
$$x_t^{f(m)} \sim g(x_t \,|\, x_{t-1}^{a(m)}, \theta_t^{f(m)}).$$

Calculate the sample covariance with the forecast samples:

$$\hat{\Sigma}_t^f = \begin{bmatrix} \hat{\Sigma}_{t,xx}^f & \hat{\Sigma}_{t,x\theta}^f \\ \hat{\Sigma}_{t,\theta x}^f & \hat{\Sigma}_{t,\theta\theta}^f \end{bmatrix},$$

where $\hat{\Sigma}_{t,xx}^f$ and $\hat{\Sigma}_{t,\theta\theta}^f$ are the sample variances of the forecast state and parameter, respectively, and $\hat{\Sigma}_{t,x\theta}^f = (\hat{\Sigma}_{t,\theta x}^f)^T$ is the sample covariance between the forecast state and parameter.

After observing $y_t$, the analysis samples are calculated by applying Equation (2.14) to the state-space model in Equation (4.4). The particular choice of EnKF update (i.e., original EnKF, perturbed observations, or square-root filter) is not important to the discussion, thus for simplicity, we apply the update of the original EnKF (c.f., Equation (2.14)) to each Gaussian kernel with center $z_t^{f(m)}$ and bandwidth $\hat{\Sigma}_t^f$:

$$z_t^{a(m)} = z_t^{f(m)} + K_t[y_t - H_t(\theta_t^{f(m)}) z_t^{f(m)}], \tag{4.7}$$

where $K_t$ is the Kalman gain:

$$K_t \equiv \mathbb{C}\mathrm{ov}(Z_t, Y_t \,|\, y_{0:t-1}) \, \mathbb{V}\mathrm{ar}(Y_t \,|\, y_{0:t-1})^{-1}, \tag{4.8}$$

where the expression is taken from Equation (2.8).

Notice that there are three parts to the update: the measurement residual $y_t - H_t(\theta_t^{f(m)}) z_t^{f(m)}$; the conditional variance of the measurement, $\mathbb{V}\mathrm{ar}(Y_t \,|\, y_{0:t-1})$; and the conditional covariance between the measurement and the state, $\mathbb{C}\mathrm{ov}(Y_t, Z_t \,|\, y_{0:t-1})$. While the measurement residual provides important information about the magnitude that the forecast state sample differs from

the measurement, the last term $\mathbb{C}\text{ov}(Y_t, Z_t \,|\, y_{0:t-1})$ is the most important because it *controls the influence* of the measurement residual. For example, if $\mathbb{C}\text{ov}(Y_t, Z_t \,|\, y_{1:t-1})$ is zero, then the state does not get updated. Let's examine the term more closely:

$$\mathbb{C}\text{ov}(Y_t, Z_t \,|\, y_{1:t-1}) = \tilde{H}_t(\theta_t^{f(m)})\hat{\Sigma}_t^f = \begin{bmatrix} H_t(\theta_t^{f(m)})\hat{\Sigma}_{t,xx}^f \\ H_t(\theta_t^{f(m)})\hat{\Sigma}_{t,x\theta}^f \end{bmatrix}$$

The first and second rows of this conditional covariance—and thus the Kalman gain—updates the state $X_t$ and parameter $\theta_t$, respectively. This shows us that there are two important components to the EnKF **linear update** of the parameters: the covariance between the state and parameter, $\hat{\Sigma}_{t,\theta x}^f$, as mentioned by previous authors[3], *and* the observation mapping $H_t(\cdot)$. Therefore, the parameter gets updated *only if* there is correlation between the forecasted state and parameter ($\hat{\Sigma}_{t,\theta x}^f$) *and* the part of the state that is correlated with the parameter is observed ($H_t(\cdot)$). Otherwise, the parameter diverges randomly or stays the same as its proposed initial state when applying artificial evolution of parameters or state augmentation, respectively. In the next section, we demonstrate with the Lorenz 1963 system that the observation mapping $H_t(\cdot)$ is just as important as the covariance between the parameter and the state, $\hat{\Sigma}_{t,\theta x}^f$.

Besides the linear update issue, we remark on a few other issues not mentioned by previous authors:

- **Unbounded update**. Often, parameters, especially those governing deterministic atmospheric models, have a range of values in which the dynamics of the system are realistic and/or numerically stable. Notice that the measurement residual in the update equation in Equation (4.7) is unbounded, hence a user cannot restrict the parameter update to be within pre-specified bounds.

- **Different Kalman gain for each update**. When estimating the measurement parameter $\theta_y$, the update of the analysis distribution requires the calculation of a different Kalman gain for each kernel in the mixture of Gaussian densities—a computationally expensive procedure. Frei and Künsch (2012) faced this issue in their development of a combined EnKF and PF algorithm to estimate the parameter $\rho$ in the measurement error $V_t(\rho)$. They circumvented the problem by replacing the samples $V_t(\theta_t^{f(m)})$

[3]Stroud and Bengtsson (2007), Delsole and Yang (2010) and Frei and Künsch (2012) are the authors in reference.

in Equation (4.8) with a point estimator, e.g., $V_t(\bar{\theta}_t^{f(m)})$ or $\overline{V_t(\cdot)}$ (see Algorithm 2 in their paper), where $\bar{x}$ denotes the sample mean calculated with the samples $\{x_i\}_{i=1}^n$. While a promising approach, a bias is introduced when sampling with a point estimator, which they do not correct for in their algorithm. We use a similar approach in our algorithm, but we also add another step to correct for the bias from using a point estimator.

In Section 4.5, we use these principles to demonstrate that the EnKF can poorly estimate parameters of interest. Before doing that, we briefly divert our attention to parametrizing variance inflation, a technique applied with the EnKF to avoid filter divergence (see Section 2.3.2 for a review).

## 4.4 Variance inflation: parametrization of the inflation value

There are two ways to parametrize the inflation value that affects whether it is a state or measurement parameter. Since the forecast ensemble is inflated, it is odd to include the inflation parameter in the measurement model—the measurement model reflects how the measurements are collected and is *not* a reflection of the state. Rather, inflation should be considered as model error: the inflation value is increasing the variance of the state, hence it reflects an increase in our uncertainty about the state transition model. We briefly outline our preferred parametrization.

We reparametrize the state transition model from Equation (4.2a) to reflect an inflation value. Suppose the state transition model is similar:

$$\tilde{x}_t = m_t(x_{t-1}, \theta_x) + \eta_t,$$

where $\eta_t$ is distributed as before. Here, we have simply replaced the state $X_t$ in Equation (4.2a) with $\tilde{X}_t$. Then, a state transition model that reflects the inflation value $\lambda$ is as follows:

$$x_t = \mu_t + \lambda(\tilde{x}_t - \mu_t)$$

where $\mu_t$ is the expected value of the state $\tilde{X}_t$. In practice, $\mu_t$ is replaced with its sample estimate to avoid estimating more parameters. With this parametrization, $\lambda$ is part of the state parameter $\theta_x$.

## 4.5 Demonstration of shortcomings with the Lorenz 1963 system

Delsole and Yang (2010) claim that state augmentation is ineffective for "stochastic" parameters, e.g. a parameter of a Wiener process in the continuous state-transition model or a parameter in the measurement noise, like the problem Stroud and Bengtsson (2007) and Frei and Künsch (2012) set out to resolve. In this section, we show that the issue is a little more nuanced and demonstrate that artificial evolution of parameters is ineffective, even for non-"stochastic" parameters, like state parameters. In this section, we use the principles outlined in Section 4.3.1 to demonstrate the shortcomings of artificial evolution of parameters with the EnKF using the Lorenz 1963 system as the state transition model. To do this, we construct state-space models to estimate the state parameter $\sigma$ (see Section 1.2.1 for a review). To examine the range of estimated values, we apply the parameter estimation algorithm multiple times. All parameter estimation runs have the same underlying data generation process. Only samples from the initial condition and the initial parameter values to start the algorithm are different between runs. The details of the runs are broken into three sections: one section on the construction of the state-space model and two sections that describe the settings of the EnKF and parameter estimation algorithm. We examine the results at the end of the section.

### Construction of state-space model and data generation

In Section 4.3.1, we highlighted two important components of the EnKF parameter update: the conditional covariance between the state and parameter, $\hat{\Sigma}^f_{t,\theta x}$, and the observation mapping $H_t(\cdot)$. The covariance $\hat{\Sigma}^f_{t,\theta x}$ cannot be relied upon to construct a state-space model to demonstrate the shortcomings of parameter estimation with the EnKF. Unlike the observation parameter $\theta_y$, the state parameter $\theta_x$ is both marginally and conditionally dependent on the state and thus the correlation is zero only for special cases. We take an alternative route: find where the state and parameters are conditionally correlated and minimize that correlation with a particular measurement mapping $H_t(\cdot)$. However, the conditional correlation cannot be obtained analytically and must be estimated by applying a filtering algorithm to the state-space model with many different parameter values and forecast lead times, which is computationally expensive. We instead rationalize measurement models that minimize the conditional correlation between the state and state parameter $\sigma$ and explain our thought process in this section.

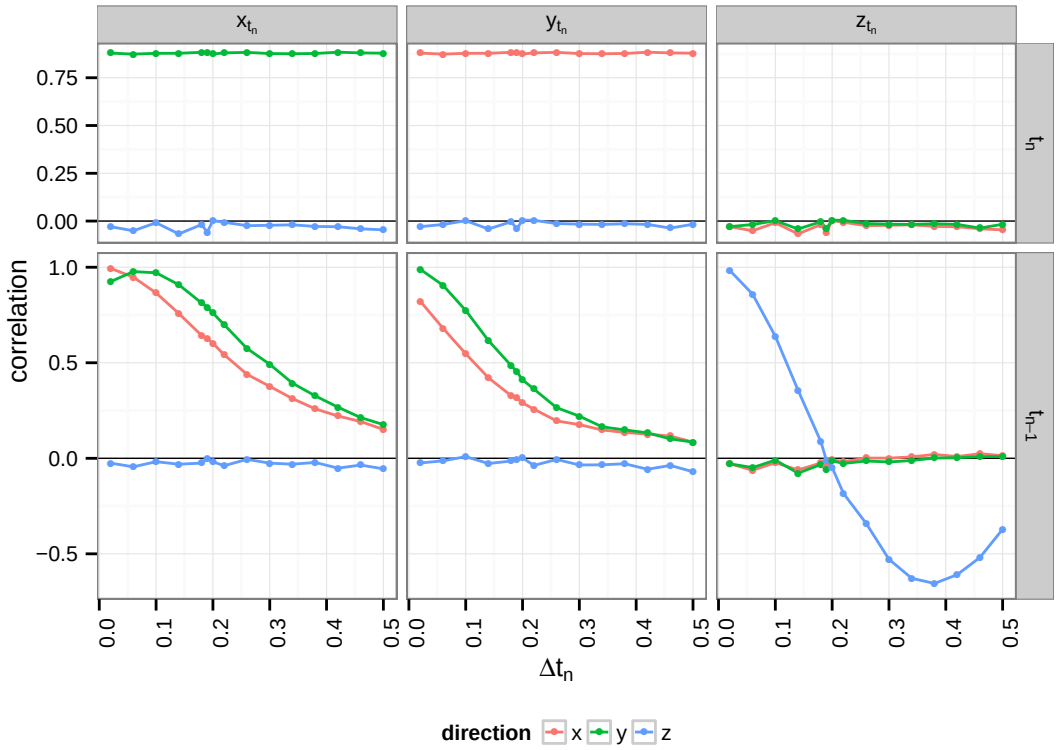



Figure 4.2: *Lorenz 1963: correlation between the integrated state and the previous state against integration time* $\Delta t_n$. The data of previous states, $(x_{t_{n-1}}, y_{t_{n-1}}, z_{t_{n-1}})$, are generated by integrating the initial condition $(x_{t_0}, y_{t_0}, z_{t_0}) = (0.01, 0.01, 0.01)$ for 30,000 timesteps of size 0.01 using an adaptive timestepping algorithm (LSODE). The first 1,000 integrated states are discarded to ensure the states are on the attractor. For each $\Delta t_n$, 5,000 initial states are randomly chosen from the set of previous states and then integrated forward with the chosen $\Delta t_n$ to obtain the integrated state $(x_{t_n}, y_{t_n}, z_{t_n})$ using second-order Runge-Kutta. The first row shows the correlations between pairs of the integrated state $(x_{t_n}, y_{t_n}, z_{t_n})$, omitting self-correlation. The second row shows the correlations of the integrated state $(x_{t_n}, y_{t_n}, z_{t_n})$ with its previous state $(x_{t_{n-1}}, y_{t_{n-1}}, z_{t_{n-1}})$.

Examining the set of equations governing the Lorenz 1963 system (cf. Equation (1.2)), the $x$-direction of the Lorenz 1963 system is dependent on the parameter $\sigma$, which seems to indicate that the $x$-direction would be highly correlated with $\sigma$. Therefore, parameter estimation with the EnKF applied to a state-space model with little to no information about the $x$-direction and any correlated directions would not estimate $\sigma$ well. To learn more about the correlation between the states, we integrated the state $(x_{t_{n-1}}, y_{t_{n-1}}, z_{t_{n-1}})$ forward to its integrated state $(x_{t_n}, y_{t_n}, z_{t_n})$ for different integration times $\Delta t_n$; Figure 4.2 graphs the correlations. We glean a few key pieces of information from this figure:

- The correlations between pairs of $x_{t_n}$-, $y_{t_n}$-, and $z_{t_n}$-directions (first row of

Figure 4.2) indicates that the $x$-direction is strongly correlated with the $y$-direction while the $z$-direction is uncorrelated with both the $x$- and $y$-directions.

- Recall from the discussion in Section 1.2.2 that chaotic models are approximately linear for small forecast lead times $\Delta t_n$ and the nonlinearities increase as $\Delta t_n$ increases. This is quantified by the second row of Figure 4.2, which show correlations of the integrated state with its previous state. The plots indicate that the state transition model is generally more nonlinear as $\Delta t_n$ increases. At $\Delta t_n \approx 0.19$, the state transition model in the $z$-direction is distinctly nonlinear. Beyond that, the nonlinearities decrease in the $x$- and $y$-directions, but becomes increasingly linear in the the $z$-direction until $\Delta t_n \approx 0.38$, at which the model becomes decreasingly nonlinear.

Since the $y$-direction is so highly correlated with the $x$-direction, the $y$-direction may also be correlated with $\sigma$ and thus should not be measured in addition to the $x$-direction. The $z$-direction can however be measured since it is uncorrelated with both the $x$- and $y$-directions. For example, we suspect an observation mapping of the following form, for small $0 \leq \gamma \leq \frac{1}{2}$:

$$H_{t_n} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \text{ or } H_{t_n} = \begin{bmatrix} \gamma & \gamma & 1-2\gamma \\ 0 & 0 & 1 \end{bmatrix}$$

would minimize the conditional correlation between the state of the system and the parameter $\sigma$. A forecast lead time of $\Delta t_n = 0.19$ would additionally eliminate any linearities of the state transition model.

Equipped with this information, we construct a state-space model for the Lorenz 1963 system in which we suspect EnKF to not estimate $\sigma$ well. When not estimating an inflation value, the state transition model is

$$\begin{bmatrix} x_{t_n} \\ y_{t_n} \\ z_{t_n} \end{bmatrix} = m_{t_n} \left( \begin{bmatrix} x_{t_{n-1}} \\ y_{t_{n-1}} \\ z_{t_{n-1}} \end{bmatrix} ; \sigma \right);$$

when estimating an inflation value, the state transition model is

$$\begin{bmatrix} x_{t_n} \\ y_{t_n} \\ z_{t_n} \end{bmatrix} = \mu_{t_n} + \lambda \left[ m_{t_n} \left( \begin{bmatrix} x_{t_{n-1}} \\ y_{t_{n-1}} \\ z_{t_{n-1}} \end{bmatrix} ; \sigma \right) - \mu_{t_n} \right].$$

The discrete state transition model $m_{t_n}$ is defined to be

$$m_{t_n}(\cdot; \sigma, dt) = \mathcal{I}_W(\cdot; \sigma) + O(e(dt)),$$

where $W$ is the continuous state transition model of Equation (1.6) and $e(\cdot)$ and $dt$ are defined in Section 1.2.1. Depending on whether the inflation value is estimated or not, the state parameter is either $\theta_x = (\sigma, \lambda)$ or $\theta_x = \sigma$, respectively.

The measurement model is

$$w_{t_n} = H(\gamma_i) \begin{bmatrix} x_{t_n} \\ y_{t_n} \\ z_{t_n} \end{bmatrix} + \epsilon_{t_n}, \quad \epsilon_{t_n} \sim \mathcal{N}(0, U).$$

The measurement mapping $H(\gamma_i)$ depends on a measurement parameter $\gamma_i$ and takes the following form:

$$H(\gamma_i) = \begin{bmatrix} \gamma_{i1} & \gamma_{i2} & \gamma_{i3} \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.9}$$

This particular measurement mapping measures the $z$-direction (second row of matrix) and some linear combination of the state represented by the measurement parameter vector $\gamma_i$ (first row of matrix). We fix $\gamma_i$ to reflect varying levels of measuring the directions suspected to be correlated with the state parameter $\sigma$, i.e., the $x$- and $y$-directions. Figure 4.3 lists our choices and Figure 4.4 illustrates the measurement locations when the state is on a spherical latitude band. We expect to estimate $\sigma$ well for some $\gamma_i$, but we only need one

| | $\gamma_{i1}$ | $\gamma_{i2}$ | $\gamma_{i3}$ | direction(s) |
|---|---|---|---|---|
| | 0 | 0 | 1 | $z$ |
| $\gamma_1$ | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | $xz$ |
| $\gamma_2$ | 1 | 0 | 0 | $x$ |
| $\gamma_3$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | $yz$ |
| $\gamma_4$ | 0 | 1 | 0 | $y$ |
| $\gamma_5$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | $xy$ |

Figure 4.3: *Lorenz 1963: measurement locations and their corresponding parameter.* The measurement mapping $H(\gamma_i)$ takes the form listed in Equation (4.9). The last column represents the linear combination of the measured directions, as illustrated in Figure 4.4. For example, "$xz$" represents an average of the $x$- and $z$-directions.

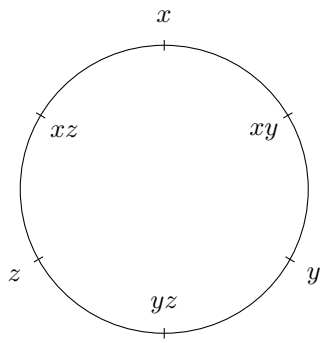



Figure 4.4: *Lorenz 1963: an illustration of measurement locations.* The tick marks of the outer labels illustrate the state locations on a spherical latitude band. The tick marks of the inner labels correspond to the measurement locations as described in Figure 4.3. The labels correspond to the last column of the aforementioned table.

case where $\sigma$ is *not* estimated well to demonstrate our point. Though an artificially constructed example, this measurement model demonstrates a realistic scenario where measurements are not collected near regions of the parameter of interest, e.g., the difficulty of measuring ocean temperature in the Arctic. The variance of the measurement noise is taken to be the identity, i.e., $U = I_2$.

The initial condition $(x_{t_0}, y_{t_0}, z_{t_0})$ is generated by taking a random state from a uniform distribution and integrating it forward with a large $\Delta t$, making sure that the initial condition is on the attractor. The states $(x_{t_n}, y_{t_n}, z_{t_n})$ are generated by taking the initial condition $(x_{t_0}, y_{t_0}, z_{t_0})$ and integrating it sequentially $N = 300$ times with a forecast lead time of $\Delta t_n = 0.19$ for all $n$. Samples from the initial condition are generated from a standard multivariate normal distribution with mean $(x_{t_0}, y_{t_0}, z_{t_0})$.

### EnKF settings

We apply the square-root filter with localization. The localization halfwidth is chosen such that the forecast covariance among all directions is zero. Ensemble sizes are set to be 6, 12, 24, 48, and 96. Parameter estimation is run both with and without inflation. When inflating, the forecast ensemble is inflated in two ways: with a fixed inflation of $\lambda = 1.5$ and with inflation values estimated by the parameter estimation procedure, as discussed further in the next section.

### Parameter estimation settings

Ionides et al. (2006, 2015) developed iterated filters to estimate MLEs using artificial evolution of parameters; the authors claim that the methods can be applied to estimate parameters with any filter, including the EnKF. However, since it relies on artificial evolution of parameters, this is not quite true, which we demonstrate by applying the iterated filter from Ionides et al. (2015) with the EnKF to estimate $\theta_x$. We first discuss the parameter estimation settings and then show the results.

Since both the parameter $\sigma$ and the inflation value $\lambda$ have positive support and we did not want unbounded proposals of the parameter, we choose the perturbation density $q(\theta_{t_n} \,|\, \theta_{t_{n-1}})$ to be a truncated normal distribution (or a product of truncated normal distributions, when estimating the inflation value) with mean $\theta_{t_{n-1}}$. The truncated normal distribution to propose new inflation values mimics time-varying adaptive inflation of Anderson (2007) without necessitating a Gaussian approximation since the constraints are enforced naturally by the truncated normal distribution. Therefore, we choose the standard deviation and constraints with guidance provided by the documentation

of Data Assimilation Research Testbed (DART). DART's documentation recommends an initial standard deviation of 0.6 and a maximum inflation value of 100, but there is a conflicting note that mentions "[e]xpected inflation values are generally in the 1 to 10 range; if values grow much larger than this it usually indicates a problem with the assimilation,"[4] so we choose a maximum value of 20 instead. For $\sigma$, we simply take 0.25 to be the initial standard deviation and 0 and 50 for the minimum and maximum values, respectively. Table 4.1 summarizes the settings.

| parameter | truth | initial sd | minimum | maximum |
|---|---|---|---|---|
| $\sigma$ | 10 | 0.25 | 0 | 50 |
| $\lambda$ | 1 | 0.6 | 1 | 20 |

Table 4.1: *Lorenz 1963 parameter estimation settings.* The perturbation density $q(\theta_{t_n} \,|\, \theta_{t_{n-1}})$ is (a product of) truncated normal distribution(s) with the above parameters.

Each run of the iterated filter is initialized with the true value of $\sigma$ (10) perturbed by the initial standard deviation multiplied by two and an inflation value of one. The iterated filter is run with 50 iterations and the MLE is reported to be the median of the final ensemble.

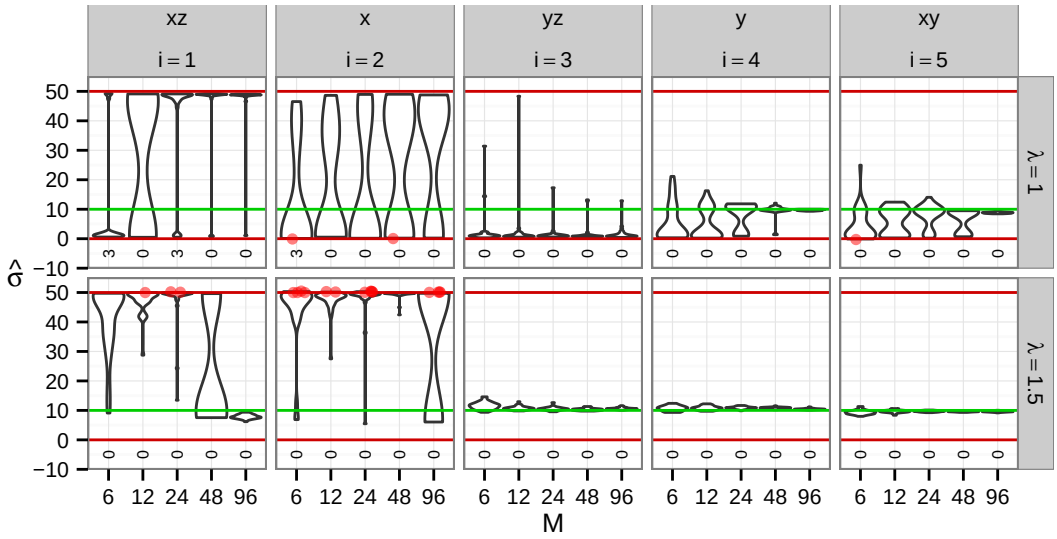
### Results

Figure 4.5 illustrates the distribution of MLEs of $\sigma$, i.e., $\hat{\sigma} = \text{argmax}_\sigma \, p_\sigma(y_{1:t_N})$. When no inflation is applied (top row of Figure 4.5a), the parameter estimation algorithm is generally ineffective at capturing the true value for all observation models, except when sample sizes are large enough for certain observation models. Since the EnKF is more successful with inflation, we henceforth discuss the results with inflation. In the construction of the state-space model, we expected the $x$-direction to be the important direction to measure when estimating $\sigma$, but the results do not corroborate our theory. With fixed inflation (bottom row of Figure 4.5a), the distribution of estimated values shows how easy it is to capture the true parameter value when the measurement includes some aspect of the $y$-direction ($i = 3, 4, 5$, where $i$ corresponds to the measurement parameter $\gamma_i$ as defined in Figure 4.3): as the ensemble size increases, the estimated values are more tightly centered around the true value. This characterizes what we expect: that the true value is better captured with larger

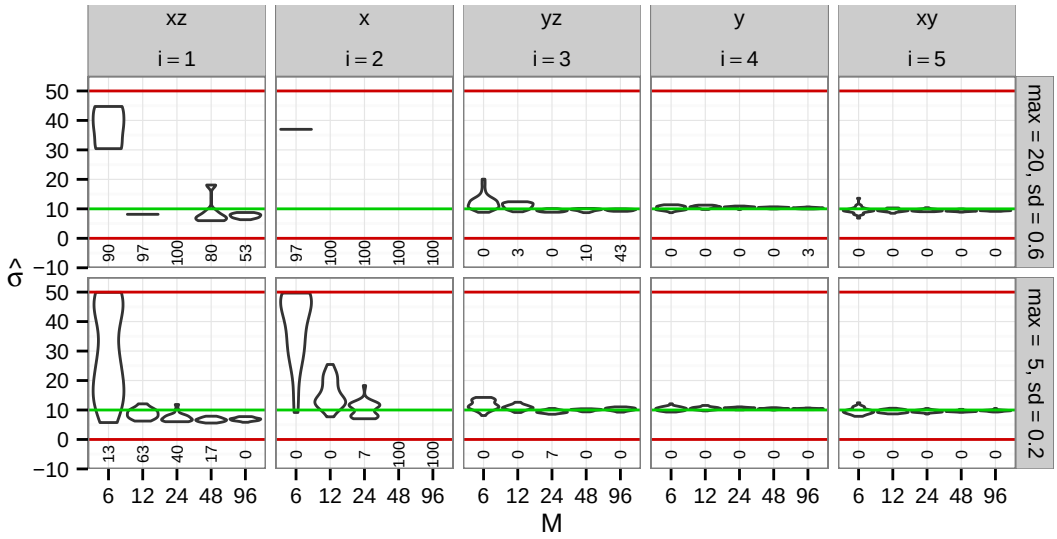
[4] Quote taken from "filter.html" of the Kodiak version of DART.

ensemble sizes. However, when the $y$-direction is not measured ($i = 1, 2$), the algorithm has a much harder time capturing the true value, even with large ensemble sizes. In fact, a handful of runs estimated values outside of the parameter constraints specified, illustrating our point about the danger of having an unbounded update with the EnKF.

The results are no better when estimating the inflation value (top row of Figure 4.5b). Again, the results show that it is easy to capture the true value of $\sigma$ when some aspect of the $y$-direction is measured and is otherwise impossible. However, instead of estimating values outside of the pre-specified constraints, many runs fail to complete the estimation procedure. The failures predictably occur for the "harder" measurement models (i.e., $i = 1, 2$). We oddly observe failures for the "easier" measurement models too; in fact, there are cases when the number of failures *increase* with larger ensemble sizes (e.g., $i = 3$). We suspected the failures were due to the settings of the perturbation density for the inflation value, so we tested our hypothesis by running the parameter estimation procedure with different settings than the default. The bottom row of Figure 4.5b shows the results from changing the default settings to a maximum value of 5 and initial standard deviation of 0.2. While the results tend to look better than with the default settings, the overall conclusions do not change: there are state-space models where the EnKF is ineffective at estimating parameters. Furthermore, the runs that completely failed demonstrate another issue with using EnKF for parameter estimation: the EnKF has no mechanism to check parameter proposals (and hence states) that lead to numerical instabilities and and hence lead to failure in the estimation procedure—an issue we further discuss when comparing these results with the ones from our proposed algorithm.




(a) *Fixed inflation.* Inflation values ($\lambda$) are fixed at the values specified by the labels to the right of each row of plots. A value of 1 indicates no inflation.




(b) *Estimated inflation.* Inflation values ($\lambda$) are estimated with maximum values and initial standard deviations as specified by the labels to the right of each row of plots. The minimum is set at 1.

Figure 4.5: *Lorenz 1963: distribution of $\hat{\sigma}$ as estimated by EnKF.* Each violin plot illustrates the distribution of $\hat{\sigma}$ resulting from 30 runs of the iterated filter varied by ensemble size, observation model, and inflation setting. Ensemble size is varied across the $x$-axis of each plot. The measurement model and inflation settings are varied across columns and rows of plots, respectively. The number below each violin plot is the percentage of runs (out of 30) that failed to estimate an MLE. The green horizontal line indicates the true parameter value: $\sigma = 10$. The red lines indicate the constraints of the perturbation density for $\sigma$ as specified in Table 4.1. A red dot is a jittered value of $\hat{\sigma}$ that is estimated to be outside of the constraint specified.

# 5 EnKF-APF: an ensemble Kalman filter to update states and a particle filter to update parameters

In the last chapter, we demonstrated on a low-dimensional model that artificial evolution of parameters with the EnKF algorithm is not generally applicable, except when the practitioner knows that the parameter being estimated is conditionally correlated with the state. Outside of fully observed systems, these situations are not straightforward to analyze a priori, especially with chaotic systems like the Lorenz 1963 system. For example, when examining the set of equations governing the Lorenz 1963 system, we thought that the $x$-direction is the most important direction to measure when estimating $\sigma$, but the results did not corroborate our theory—it turns out the $y$-direction is the most important. Practitioners could alternatively use a PF approach to estimate the parameters, but the PF has been proven to be impractical for high-dimensional state-spaces (Snyder et al., 2008).

In this chapter, we propose an algorithm that combines the best qualities of both the EnKF and PF algorithms: a hybrid approach that uses the EnKF to filter the high-dimensional states and the PF to filter low-dimensional parameters. There are many ways to combine the two filters, but we choose to combine the two filters in such a way that allows for plug-and-play into existing systems such as DART. We fulfill this goal by constructing an updating algorithm that doesn't affect the implementation of the EnKF to update the state. As the reader will learn, our particular approach is reminiscent of the widely-used PF algorithm: the auxiliary particle filter (APF) proposed by Pitt and Shephard (1999). For this reason, we name our algorithm EnKF-APF to remind the reader that the EnKF is applied to the states and a PF approach similar to the APF is applied to the parameters. We further note that our

algorithm is specifically developed to be used with artificial evolution of parameters. There is no need to use our algorithm with parameter estimation algorithms such as particle MCMC (Andrieu et al., 2010).

Similar to the derivations of the sampling algorithms in Chapter 2, we begin with an estimator of the analysis distribution at time $t-1$ and derive estimators to the filtering densities at time $t$, which are divided into sections in this chapter. Suppose we have simple random samples from the approximate analysis distribution at time $t-1$: $\{(x_{t-1}^{a(m)}, \theta_{t-1}^{a(m)})\}_{m=1}^{M} \dot{\sim} p(x_{t-1}, \theta_{t-1} \,|\, y_{0:t-1})$. An estimator of the joint analysis distribution at time $t-1$, i.e., $\pi_{0:t-1}(x_{t-1}, \theta_{t-1})$, is

$$\pi_{0:t-1}^{M}(x_{t-1}, \theta_{t-1}) \equiv \frac{1}{M} \sum_{m=1}^{M} \delta_{(x_{t-1}^{a(m)}, \theta_{t-1}^{a(m)})}(x_{t-1}, \theta_{t-1}).$$

We now derive estimators of the filtering densities and thus construct the EnKF-APF algorithm.

## 5.1 Forecast step

By conditional probability, the joint forecast distribution is factorized as

$$p(x_t, \theta_t \,|\, y_{0:t-1}) = p(x_t \,|\, \theta_t, y_{0:t-1}) p(\theta_t \,|\, y_{0:t-1}), \tag{5.1}$$

Notice that the state is conditionally dependent on the parameter; this suggests a sequential sampling scheme to first sample from the forecast distribution of the parameter and then use the newly sampled parameter to then sample from the state distribution. Algorithm 5.1 outlines the forecasting algorithm. Not only does the algorithm provide samples from the joint forecast distribution $p(x_t, \theta_t \,|\, y_{0:t-1})$, it also provides samples from the conditional distributions $p(x_t \,|\, \theta_t, y_{0:t-1})$ and $p(\theta_t \,|\, y_{0:t-1})$. Furthermore, this forecasting algorithm is the same whether the EnKF or PF algorithm is applied; nothing new is proposed here. The new contribution is the estimators used to approximate the joint forecast distribution, which consequently affects the construction of the updating algorithms.

Recall that the key defining difference between the EnKF and the PF is the estimator used to approximate the forecast distribution: the PF uses the empirical density and the EnKF uses a KDE with Gaussian kernels centered at each sample with a bandwidth equal to the sample covariance (see discussion in Section 3.1). We use similar estimators for the conditional forecast

**Input**: $\{(x_{t-1}^{a(m)}, \theta_{t-1}^{a(m)})\}_{m=1}^{M} \dot{\sim} p(x_{t-1}, \theta_{t-1} \,|\, y_{0:t-1})$
**Output**:
- $\{\theta_t^{f(m)}\}_{m=1}^{M} \dot{\sim} p(\theta_t \,|\, y_{0:t-1})$
- $\{x_t^{f(m)}\}_{m=1}^{M} \dot{\sim} p(x_t \,|\, \theta_t, y_{0:t-1})$
- $\{(x_t^{f(m)}, \theta_t^{f(m)})\}_{m=1}^{M} \dot{\sim} p(x_t, \theta_t \,|\, y_{0:t-1})$

For each $m = 1, ..., M$,

1. Sample $\theta_t^{f(m)} \sim q(\theta_t \,|\, \theta_{t-1}^{a(m)})$.
2. Sample $x_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)}, \theta_t^{f(m)})$.

Algorithm 5.1: *EnKF-APF: Forecast step*

distributions:

$$\pi_{0:t}^{M}(\theta_t) \equiv \frac{1}{M} \sum_{m=1}^{M} \delta_{\theta_t^{f(m)}}(\theta_t), \tag{5.2a}$$

$$\pi_{0:t}^{M}(x_t \,|\, \theta_t) \equiv \frac{1}{M} \sum_{m=1}^{M} \phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f), \tag{5.2b}$$

where $\hat{\Sigma}_t^f$ is the sample covariance calculated from the state samples $\{x_t^{f(m)}\}_{m=1}^{M}$ and the superscript $M$ is to remind the reader that $\pi_{0:t}^{M}$ is an estimator of $\pi_{0:t}$ that depends on the sample size $M$. These estimators will play a role in the derivation of the estimator to the analysis distribution in the next section, but will not be as fundamental as the estimator of the joint forecast distribution.

At first glance, Equation (5.1) seems to indicate that an estimator of the joint forecast distribution is simply the product of the conditional forecast distributions of Equation (5.2). However, multiplying the conditional forecast distributions in that way is fundamentally incorrect: this results in an estimator that implies that the forecast state samples are independent of the forecast parameter samples and consequently artificially increases the sample size. Let's see how that happens. Multiplying the estimators of the conditional forecast distributions from Equation (5.2) together, we have

$$\frac{1}{M^2} \sum_{n=1}^{M} \sum_{m=1}^{M} \phi(x_t; x_t^{f(n)}, \hat{\Sigma}_t^f) \delta_{\theta_t^{f(m)}}(\theta_t),$$

implying that $\{x_t^{f(n)}\}_{n=1}^{M}$ is a simple random sample from $\pi_{0:t-1}(x_t \,|\, \theta_t^{f(m)})$ for each $m$ or, in other words, $X_t^f$ is independent of $\Theta_t^f$. In actuality, the state

sample $x_t^{f(n)}$ is generated by $\theta_t^{f(m)}$ for $m = n$ and *together* they are a sample from the joint distribution. With this in mind, we propose a hybrid estimator of the joint forecast distribution: a mixture distribution where each component is a product of a Gaussian kernel centered at a state sample and a point mass at a parameter sample:

$$\pi_{0:t}^M(x_t, \theta_t) \equiv \frac{1}{M} \sum_{m=1}^{M} \phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f) \delta_{\theta_t^{f(m)}}(\theta_t). \tag{5.3}$$

Equipped with estimators of the conditional and joint forecast distributions, we now derive an estimator for the analysis distribution.

## 5.2 Update step: preliminaries

There are a couple ways to derive an estimator for the joint analysis distribution. The final estimator we derive in this section yields an algorithm that is amenable to plug-and-play for two reasons: the updating mechanism uses the same Kalman gain for all state samples and the state samples never have to be resampled in such a way that causes particle collapse. Recall that the EnKF's update step solely relies on the Kalman gain. In this chapter, we denote the Kalman gain by

$$K_t(\theta, \Sigma) \equiv \Sigma H_t^T(\theta)[V_t(\theta) + H_t(\theta)\Sigma H_t^T(\theta)]^{-1}. \tag{5.4}$$

Because each state sample $x_t^{f(m)}$ has a different parameter sample $\theta_t^{f(m)}$ associated with it, each state sample $x_t^{f(m)}$ is therefore updated with a different Kalman gain $K_t(\theta_{t,y}^{f(m)}, \cdot)$. Not only is the calculation of the Kalman gain computationally expensive, existing implementations of the EnKF, such as DART, update each state sample with the *same* Kalman gain (Step 3 of Algorithm 2.1). Regardless of the EnKF update algorithm used (original, perturbed observations, or deterministic square-root filters; see Section 2.3.2 for a discussion), we want to use an estimator reminiscent of Equation (3.8) as an estimator[1] of the conditional state analysis distribution for one simple reason: the state particles are equally weighted and need not be resampled, thus avoiding particle collapse.

With these two goals in mind, we derive estimators of the analysis distribution that yield algorithms ideal for plug-and-play. Before discussing the

[1]We want to use the estimator from Equation (3.8) as opposed to the intermediary estimator $\tilde{\psi}_{1:t}^M(x_t)$ of Equation (3.7). See Section 3.1 for a discussion.

updating algorithms, we end this section with estimators for the predictive likelihood, which are important in the development of the algorithms in the remaining sections. In Section 5.3, we consider the case when $\Theta_{t,y}$ is fixed: the measurement density no longer depends on the parameter and therefore each sample is updated with the same Kalman gain. We then derive two estimators based on different factorizations of the joint analysis distribution of Equation (4.6b). We begin with a direct application of Equation (4.6b), which suggests a joint update of the state and parameter, and discuss why this particular approach is not ideal for plug-and-play. We then derive another estimator based on a different factorization of the joint analysis distribution that suggests a sequential update of the parameter and then the state. Most of the chapter is dedicated to explaining the sequential updating algorithm in elaborate detail because the algorithm to estimate the observation parameter $\Theta_{t,y}$ (Section 5.4) is only a slight modification of the sequential updating algorithm when $\Theta_{t,y}$ is fixed.

### 5.2.1 Predictive and conditional predictive likelihoods

Before deriving the estimators of the analysis distribution, we derive estimators of two different forms of the predictive likelihood: one that has been introduced before—the predictive likelihood $p(y_t \mid y_{0:t-1})$ from Equation (4.6c)—and one that has not been seen before—the conditional predictive likelihood $p(y_t \mid \theta_t, y_{0:t-1})$. These two forms of the predictive likelihood are important in the construction of the updating algorithms.

The predictive likelihood from Equation (4.6c) is repeated here for convenience:

$$\pi_{0:t-1}(y_t) \equiv p(y_t \mid y_{0:t-1}) = \int\int \phi[y_t; H_t(\theta_{t,y})x_t, V_t(\theta_{t,y})]p(x_t, \theta_t \mid y_{0:t-1})dx_t d\theta_t$$

An estimator of the predictive likelihood is derived by plugging the estimator of the joint forecast distribution (Equation (5.3)) into the above equation:

$$\begin{aligned}\pi^M_{0:t-1}(y_t) &\equiv \int\int \phi[y_t; H_t(\theta_{t,y})x_t, V_t(\theta_{t,y})]\pi^M_{0:t-1}(x_t, \theta_t)dx_t d\theta_t \\ &= \frac{1}{M}\sum_{m=1}^{M}\int \phi[y_t; H_t(\theta^{f(m)}_{t,y})x_t, V_t(\theta^{f(m)}_{t,y})]\phi(x_t; x^{f(m)}_t, \hat{\Sigma}^f_t)dx_t\end{aligned}$$

The summand of the predictive likelihood is important in the derivation of the estimator to the analysis distribution, so we denote the summand by the

function

$$l_t(x_t^{f(m)}, \theta_t^{f(m)}, \Sigma) \equiv \int \phi[y_t; H_t(\theta_t^{f(m)})x_t^{f(m)}, V_t(\theta_t^{f(m)})]\phi(x_t; x_t^{f(m)}, \Sigma)dx_t$$
$$= \phi[y_t; H_t(\theta_t^{f(m)})x_t^{f(m)}, V_t(\theta_t^{f(m)}) + H_t(\theta_t^{f(m)})\Sigma H_t^T(\theta_t^{f(m)})], \tag{5.5}$$

where $(x_t^{f(m)}, \theta_t^{f(m)}) \sim \pi_{0:t-1}^M(x_t, \theta_t)$. Therefore, the estimator of the predictive likelihood is

$$\pi_{0:t-1}^M(y_t) = \frac{1}{M}\sum_{m=1}^{M} l_t(x_t^{f(m)}, \theta_{t,y}^{f(m)}, \hat{\Sigma}_t^f).$$

We introduce the conditional predictive likelihood $p(y_t \,|\, \theta_t, y_{0:t-1})$ that will be important in the construction of the sequential updating algorithm. By the axiom of conditional probability,

$$\pi_{0:t-1}(y_t \,|\, \theta_t) \equiv p(y_t \,|\, \theta_t, y_{0:t}) = \int p(y_t \,|\, x_t, \theta_t, y_{0:t-1})p(x_t \,|\, \theta_t, y_{0:t-1})dx_t$$
$$= \int \phi[y_t; H_t(\theta_{t,y})x_t, V_t(\theta_{t,y})]p(x_t \,|\, \theta_t, y_{0:t-1})dx_t,$$

where the last equality comes from the measurement independence assumption:

$$p(y_t \,|\, x_t, \theta_t, y_{0:t-1}) = g(y_t \,|\, x_t, \theta_{t,y}) = \phi[y_t; H_t(\theta_{t,y})x_t, V_t(\theta_{t,y})]. \tag{5.6}$$

An estimator of the conditional predictive likelihood is constructed by plugging in the estimator of the conditional forecast distribution of the state (Equation (5.2b)):

$$\pi_{0:t-1}^M(y_t \,|\, \theta_t) \equiv \int \phi[y_t; H_t(\theta_{t,y})x_t, V_t(\theta_{t,y})]\pi_{0:t-1}^M(x_t \,|\, \theta_t)dx_t$$
$$= \frac{1}{M}\sum_{m=1}^{M}\left[\int \phi[y_t; H_t(\theta_{t,y})x_t, V_t(\theta_{t,y})]\phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f)dx_t\right]$$
$$= \frac{1}{M}\sum_{m=1}^{M} l_t(x_t^{f(m)}, \theta_{t,y}, \hat{\Sigma}_t^f).$$

## 5.3 Update step: when the measurement parameter is fixed

Suppose the measurement parameter $\Theta_{t,y}$ is fixed at some vector $\bar{\theta}_{t,y}$ for each $t$. Then, the parameter is $\Theta_t = (\Theta_{t,x}, \bar{\theta}_{t,y})$ and consequently samples from its forecast and analysis distributions are $\theta_t^{f(m)} = (\theta_{t,x}^{f(m)}, \bar{\theta}_{t,y})$ and $\theta_t^{a(m)} = (\theta_{t,x}^{a(m)}, \bar{\theta}_{t,y})$, respectively. Since the difference between the predictive and conditional predictive likelihoods comes from the observation parameter $\theta_{t,y}$, they are equivalent when $\Theta_{t,y}$ is fixed at $\bar{\theta}_{t,y}$ and thus

$$\pi_{1:t-1}^M(y_t) = \pi_{1:t-1}^M(y_t \,|\, \theta_t) = \frac{1}{M}\sum_{m=1}^{M} l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f).$$

Because the measurement parameter $\Theta_{t,y}$ is fixed, the algorithm in this section only updates the state parameter $\Theta_{t,x}$.

### 5.3.1 Joint update of the state and parameter

The fixed value of the measurement parameter $\bar{\theta}_{t,y}$ is plugged into Equation (4.6b) to derive the joint analysis distribution as follows:

$$\pi_{1:t}(x_t, \theta_t) \equiv p(x_t, \theta_t \,|\, y_{1:t}) = \frac{\phi[y_t; H_t(\bar{\theta}_{t,y})x_t, V_t(\bar{\theta}_{t,y})]p(x_t, \theta_t \,|\, y_{1:t-1})}{p(y_t \,|\, y_{1:t-1})} \tag{5.7}$$

A direct application of the above factorization suggests a joint update of the state and parameter by weighting each sample proportionally to the observation density $\phi[y_t; H_t(\bar{\theta}_{t,y})x_t, V_t(\bar{\theta}_{t,y})]$. We show why this particular approach is not ideal for plug-and-play.

Plugging in the estimator of the joint forecast distribution from Equation (5.3), we derive an estimator of the joint analysis distribution:

$$\begin{aligned}
\pi_{1:t}^M(x_t, \theta_t) &\equiv \frac{\phi[y_t; H_t(\bar{\theta}_{t,y})x_t, V_t(\bar{\theta}_{t,y})]\pi_{1:t-1}^M(x_t, \theta_t)}{\pi_{1:t-1}^M(y_t)} \\
&= \frac{\frac{1}{M}\sum_{m=1}^{M} \phi[y_t; H_t(\bar{\theta}_{t,y})x_t, V_t(\bar{\theta}_{t,y})]\phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f)\delta_{\theta_t^{f(m)}}(\theta_t)}{\frac{1}{M}\sum_{n=1}^{M} l_t(x_t^{f(n)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)}
\end{aligned}$$

Multiplying the summands of the numerator by the identity, $l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)/l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)$, we have

$$\begin{aligned} & l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f) \frac{\phi[y_t; H_t(\bar{\theta}_{t,y})x_t, V_t(\bar{\theta}_{t,y})]\phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f)}{l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)} \delta_{\theta_t^{f(m)}}(\theta_t) \\ & = l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f) \underbrace{\frac{\phi[y_t; H_t(\bar{\theta}_{t,y})x_t, V_t(\bar{\theta}_{t,y})]\phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f)}{\phi[y_t; H_t(\bar{\theta}_{t,y})x_t, V_t(\bar{\theta}_{t,y}) + H_t(\bar{\theta}_{t,y})\hat{\Sigma}_t^f H_t^T(\bar{\theta}_{t,y})]}}_{} \delta_{\theta_t^{f(m)}}(\theta_t). \end{aligned}$$

As in the re-interpretation of the EnKF in Section 3.1, the term above the underbrace should remind the reader of the analysis distribution of the Kalman filter (cf. Equation (2.6)) and is thus equal to

$$\phi(x_t; x_t^{a(m)}, \hat{\Sigma}_t^a)$$

with

$$\begin{aligned} x_t^{a(m)} &= x_t^{f(m)} + K_t(\bar{\theta}_{t,y}, \hat{\Sigma}_t^f)[y_t - H_t(\bar{\theta}_{t,y})x_t^{f(m)}], \\ \hat{\Sigma}_t^a &= \hat{\Sigma}_t^f - K_t(\bar{\theta}_{t,y}, \hat{\Sigma}_t^f)\hat{\Sigma}_t^f H_t^T(\bar{\theta}_{t,y}). \end{aligned}$$

Therefore, the estimator is

$$\pi_{1:t}^M(x_t, \theta_t) = \sum_{m=1}^M w_t^{(m)} \phi(x_t; x_t^{a(m)}, \hat{\Sigma}_t^a) \delta_{\theta_t^{f(m)}}(\theta_t),$$

with importance weights

$$w_t^{(m)} = \frac{l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)}{\sum_{n=1}^M l_t(x_t^{f(n)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)}.$$

The samples $\{(x_t^{f(m)}, \theta_t^{f(m)})\}_{m=1}^M$ do not generally have uniform weights, i.e., $w_t^{(m)} \neq 1/M$, and, since our goal is to develop an algorithm that more closely resembles the EnKF, uniform weights are needed. To achieve uniform weights, there are two ways forward at this point:

1. Resample the samples as pairs: $(x_t^{a(m)}, \theta_t^{a(m)}) \sim \pi_{1:t}^M(x_t, \theta_t)$. This particular approach is akin to updating with a PF and does not fully take advantage of EnKF's ability of not requiring resampling, thus particle collapse is inevitable.

2. Approximate the weights $w_t^{(m)}$ to be uniform as done with the EnKF (see Equation (3.6)). Recall from the discussion of the EnKF in Section 3.1 that this approximation consequently does not require resampling and thus avoids particle collapse. However, since the weights are a crucial component of the PF's updating algorithm, this approach will not update the parameter.

In fact, these approaches are what is performed when directly artificial evolution of parameters with the PF and EnKF, respectively. These approaches are unsatisfactory for the reasons discussed in Chapter 4. Since the state and parameter particles are tightly coupled in this particular estimator, we cannot separately use PF to update the parameters and EnKF to update the states. Another approach is required.

### 5.3.2 Sequential update of the state and parameter

The approach in this section circumvents the tightly coupled update of the state and parameters by first updating the parameter and then jointly updating the state and parameter. This approach is based on another factorization of the joint analysis distribution, derived by the axiom of conditional probability:

$$\pi_{0:t}(x_t, \theta_t) \equiv p(x_t, \theta_t \,|\, y_{0:t}) = p(x_t \,|\, \theta_t, y_{0:t}) p(\theta_t \,|\, y_{0:t}). \tag{5.8}$$

Similar to the factorization of the joint forecast distribution in Section 5.1, the conditional dependence of the state on the parameter in the above factorization suggests a two-stage sequential updating algorithm. In this section, we first summarize the algorithm with a single equation and then elaborate on each stage of the algorithm further by deriving estimators relevant to that particular stage.

The single equation that summarizes the algorithm is

$$\pi_{0:t}(x_t, \theta_t) = \frac{\phi[y_t; H_t(\bar{\theta}_{t,y}) x_t, V_t(\bar{\theta}_{t,y})]}{p(y_t \,|\, \theta_t, y_{0:t-1})} \times \frac{p(y_t \,|\, \theta_t, y_{0:t-1})}{p(y_t \,|\, y_{0:t-1})} p(x_t, \theta_t \,|\, y_{0:t}), \tag{5.9}$$

which is derived by applying Bayes theorem to the conditional distributions on the right hand side of Equation (5.8) or, even more simply, multiplying the joint analysis distribution from Equation (5.7) by the identity constructed by dividing the conditional predictive likelihood $p(y_t \,|\, \theta_t, y_{0:t-1})$ by itself. For clarity, we have strategically separated parts of the above equation with the product symbol ("$\times$") to symbolize the separation of the two stages of the algorithm:

1. The first-stage of the algorithm updates the parameter via a PF update with weights proportional to the conditional predictive likelihood $p(y_t \mid \theta_t, y_{0:t-1})$, which is effectively equivalent to deriving a biased estimator of the joint forecast distribution, i.e., the term to the right of the "$\times$" symbol.

2. The second-stage corrects for the bias introduced in the first-stage via an EnKF update with weights proportional to the term to the left of the "$\times$" symbol. Not only does this stage correct for the bias, it also jointly updates the state and parameter.

As mentioned earlier, our two-stage algorithm is reminiscent of the APF (Pitt and Shephard, 1999), hence we name our algorithm the EnKF-APF algorithm. With the APF, the updating and resampling steps of the BF are interchanged: the first-stage (the resampling step) uses the measurement to pre-sample particles that are more likely to survive and the second-stage (the updating step) updates the state samples in a manner that corrects for the bias introduced by the pre-sampling step in the first-stage. Like the first-stage of the APF, the first-stage of our algorithm pre-samples pairs of state and parameter samples with the conditional predictive likelihood, effectively updating *only* the parameter. The new samples provide a biased estimator of the joint forecast distribution. Though biased, it is crucial to avoid resampling when applying the EnKF update to the pre-sampled particles in the second-stage of the algorithm. The second-stage then performs an EnKF update, which serves two purposes: (1) it updates the state particles with existing implementations of the EnKF and (2) it corrects for the bias introduced in the first-stage. Consequently, the algorithm provides simple random samples from an estimator of the joint analysis distribution $p(x_t, \theta_t \mid y_{0:t})$. The algorithm, however, requires two evaluations of the state transition per sample, increasing the computational time of the algorithm, but it is a necessary expense to avoid the issues faced by applying artificial evolution of parameters with the EnKF. We further elaborate on each step by deriving the estimators necessary for each stage of the algorithm.

Let's begin the development of the first-stage of the algorithm by deriving an estimator for the conditional distribution of the parameter. Applying Bayes

theorem, the conditional distribution of the parameter is

$$\begin{aligned}\pi_{0:t}(\theta_t) &\equiv p(\theta_t \,|\, y_{0:t}) \\ &= \frac{p(y_t \,|\, \theta_t, y_{0:t-1}) p(\theta_t \,|\, y_{0:t-1})}{p(y_t \,|\, y_{0:t-1})} \\ &= \frac{\int \phi[y_t; H_t(\theta_{t,y}) x_t, V_t(\theta_{t,y})] p(x_t, \theta_t \,|\, y_{0:t-1}) dx_t}{p(y_t \,|\, y_{0:t-1})}.\end{aligned} \tag{5.10}$$

The conditional distribution suggests updating the parameter with weights proportional to the conditional predictive likelihood $p(y_t \,|\, \theta_t, y_{0:t-1})$. Plugging in our estimators, we derive an estimator of the conditional parameter distribution:

$$\begin{aligned}\tilde{\pi}^M_{0:t}(\theta_t) &\equiv \frac{\int \phi[y_t; H_t(\bar{\theta}_{t,y}) x_t, V_t(\bar{\theta}_{t,y})] \pi^M_{0:t-1}(x_t, \theta_t) dx_t}{\pi^M_{0:t-1}(y_t)} \\ &= \frac{\frac{1}{M} \sum_{m=1}^M \left[ \int \phi[y_t; H_t(\bar{\theta}_{t,y}) x_t, V_t(\bar{\theta}_{t,y})] \phi(x_t; x_t^{f(m)}, \hat{\Sigma}_t^f) dx_t \right] \delta_{\theta_t^{f(m)}}(\theta_t)}{\frac{1}{M} \sum_{n=1}^M l_t(x_t^{f(n)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)} \\ &= \frac{\sum_{m=1}^M l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f) \delta_{\theta_t^{f(m)}}(\theta_t)}{\sum_{n=1}^M l_t(x_t^{f(n)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)}\end{aligned}$$

Let $w_t^{(m)} = l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f) / \sum_{n=1}^M l_t(x_t^{f(n)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)$. Then,

$$\tilde{\pi}^M_{0:t}(\theta_t) = \sum_{m=1}^M w_t^{(m)} \delta_{\theta_t^{f(m)}}(\theta_t)$$

Each parameter sample has a nonuniform weight—a weight that is undesirable to carry over to the state update in the second-stage of the algorithm. Therefore, we resample the parameter samples *before* updating the state: $\theta_t^{a(m)} \sim \tilde{\pi}^M_{0:t}(\theta_t)$ for each $m = 1, ..., M$. These samples provide a new estimator of the conditional parameter distribution:

$$\pi^M_{0:t}(\theta_t) = \frac{1}{M} \sum_{m=1}^M \delta_{\theta_t^{a(m)}}(\theta_t).$$

Though resampling parameter samples avoids resampling the state samples in the second-stage of the algorithm, it introduces a new problem: as with forecasting, the state and parameter samples should be considered as a sample from the joint forecast distribution *as pairs*. However, since only the parameter

samples are resampled, the resampling disentangled the pairings between the parameter and the state samples. The problem is remedied by sampling from the conditional forecast distribution of the state again—this time with the resampled parameter samples:

$$\tilde{x}_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)}, \theta_t^{a(m)})$$

for each $m = 1, ..., M$. This step is equivalent to sampling from an estimator of the following distribution:

$$\frac{p(y_t \,|\, \theta_t, y_{0:t-1})}{p(y_t \,|\, y_{0:t-1})} p(x_t, \theta_t \,|\, y_{0:t-1}),$$

i.e., the term to the right of the "$\times$" symbol in the single equation that describes the algorithm (Equation (5.9)). Effectively, the samples are biased samples from the forecast distribution $p(x_t, \theta_t \,|\, y_{0:t-1})$, biased by the weight $p(y_t \,|\, \theta_t, y_{0:t-1})$ used to update the parameter. As in Section 5.1, we construct a hybrid estimator for the joint forecast distribution with these new samples:

$$\tilde{\pi}_{0:t-1}^M(x_t, \theta_t) = \frac{1}{M} \sum_{m=1}^{M} \phi(x_t; \tilde{x}_t^{f(m)}, \hat{\tilde{\Sigma}}_t^f) \delta_{\theta_t^{a(m)}}(\theta_t), \tag{5.11}$$

where $\hat{\tilde{\Sigma}}_t^f$ is the sample covariance calculated from the state particles $\{\tilde{x}_t^{f(m)}\}_{m=1}^M$. Similarly, a new estimator of the conditional predictive distribution is derived:

$$\tilde{\pi}_{0:t-1}^M(y_t \,|\, \theta_t) \equiv \frac{1}{M} \sum_{m=1}^{M} l_t(\tilde{x}_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\tilde{\Sigma}}_t^f).$$

In the second-stage of the algorithm, the pairs of state and parameter samples are jointly updated in a fashion similar to the joint update discussed in Section 5.3.1. The joint update also corrects for the bias introduced by the first-stage of the algorithm. Applying Bayes' theorem to the conditional state distribution, we have

$$\begin{aligned} \pi_{0:t}(x_t \,|\, \theta_t) \equiv p(x_t \,|\, \theta_t, y_{0:t}) &= \frac{p(y_t \,|\, x_t, \theta_t, y_{0:t-1}) p(x_t \,|\, \theta_t, y_{0:t-1})}{p(y_t \,|\, \theta_t, y_{0:t-1})} \\ &= \frac{\phi[y_t; H_t(\theta_{t,y}) x_t, V_t(\theta_{t,y})] p(x_t \,|\, \theta_t, y_{0:t-1})}{p(y_t \,|\, \theta_t, y_{0:t-1})}, \end{aligned} \tag{5.12}$$

where the last equality comes from the measurement independence assumption from Equation (5.6). Plugging in estimators of the conditional parameter and

state distributions from Equations (5.10) and (5.12), respectively, into the joint analysis distribution of Equation (5.8), Equation (5.9) that summarized our algorithm is derived. We repeat it here for convenience:

$$\pi_{0:t}(x_t, \theta_t) = \frac{\phi[y_t; H_t(\bar{\theta}_{t,y})x_t, V_t(\bar{\theta}_{t,y})]}{p(y_t \,|\, \theta_t, y_{0:t-1})} \times \frac{p(y_t \,|\, \theta_t, y_{0:t-1})}{p(y_t \,|\, y_{0:t-1})} p(x_t, \theta_t \,|\, y_{0:t}).$$

The estimator in Equation (5.11) is an estimator of the term to the right of the "$\times$" symbol, thus an estimator for the joint analysis distribution is derived in a similar manner as in Section 5.3.1:

$$\begin{aligned}\tilde{\pi}^M_{0:t}(x_t, \theta_t) &\equiv \frac{\phi[y_t; H_t(\bar{\theta}_{t,y})x_t, V_t(\bar{\theta}_{t,y})]\tilde{\pi}^M_{0:t-1}(x_t, \theta_t)}{\tilde{\pi}^M_{0:t-1}(y_t \,|\, \theta)} \\ &= \sum_{m=1}^{M} w_t^{(m)} \phi(x_t; x_t^{a(m)}, \hat{\Sigma}^a_t) \delta_{\theta_t^{a(m)}}(\theta_t),\end{aligned}$$

where

$$\begin{aligned} x_t^{a(m)} &= \tilde{x}_t^{f(m)} + K_t(\bar{\theta}_{t,y}, \hat{\tilde{\Sigma}}^f_t)[y_t - H_t(\bar{\theta}_{t,y})\tilde{x}_t^{f(m)}] \\ \hat{\Sigma}^a_t &= \hat{\tilde{\Sigma}}^f_t - K_t(\bar{\theta}_{t,y}, \hat{\tilde{\Sigma}}^f_t)\hat{\tilde{\Sigma}}^f_t H_t^T(\bar{\theta}_{t,y}), \\ w_t^{(m)} &= \frac{l_t(\tilde{x}_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\tilde{\Sigma}}^f_t)}{\sum_{n=1}^M l_t(\tilde{x}_t^{f(n)}, \bar{\theta}_{t,y}, \hat{\tilde{\Sigma}}^f_t)}. \end{aligned} \tag{5.13}$$

Like in Section 5.3.1, the pairs of state and parameter samples have unequal weights and we suggested two ways forward: either jointly update the pairs of state and parameter samples via a PF or EnKF update. We hesitated to move forward with either approach because the PF update would lead to particle collapse and the EnKF update requires approximations that were not ideal for updating the parameter samples. However, at this stage of the sequential updating algorithm, we do not share the same hesitation about the EnKF update as before: the parameter has already been updated in the construction of the new estimator of the forecast distribution, $\tilde{\pi}^M_{0:t-1}(x_t, \theta_t)$. Therefore, introducing the weight approximation from the EnKF at this stage updates the state in a manner faithful to an EnKF update. Therefore, pairs of state and parameter samples are jointly updated by directly applying the EnKF update (Step 3 of Algorithm 2.1). Algorithm 5.2 summarizes the sequential updating algorithm.

**Input**:
- $\{x_{t-1}^{a(m)}\}_{m=1}^{M} \dot{\sim} p(x_{t-1} \mid y_{0:t-1})$
- $\{(x_t^{f(m)}, \theta_t^{f(m)})\}_{m=1}^{M} \dot{\sim} p(x_t, \theta_t \mid y_{0:t-1})$ from Algorithm 5.1

**Output**: $\{(x_t^{a(m)}, \theta_t^{a(m)})\}_{m=1}^{M} \dot{\sim} p(x_t, \theta_t \mid y_{0:t})$

1. Update $\Theta_t$:

   a) Calculate unnormalized weights[a]:

   $$\tilde{w}_t^{(m)} = l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)$$

   for each $m = 1, ..., M$, where $\hat{\Sigma}_t^f$ is the (tapered) sample covariance calculated from $\{x_t^{f(m)}\}_{m=1}^{M}$. Normalize weights: $w_t^{(m)} = \tilde{w}_t^{(m)} / \sum_{n=1}^{M} \tilde{w}_t^{(n)}$ for each $m = 1, ..., M$.

   b) Sample

   $$\theta_t^{a(m)} \sim \sum_{m=1}^{M} w_t^{(m)} \delta_{\theta_t^{f(m)}}(\theta_t)$$

   for each $m = 1, ..., M$.

2. Update $(X_t, \Theta_t)$:

   a) Sample $\tilde{x}_t^{f(m)} \sim f(x_t \mid x_{t-1}^{a(m)}, \theta_t^{a(m)})$.

   b) Sample $\{x_t^{a(m)}\}_{m=1}^{M}$ using Step 3 of Algorithm 2.1 with forecast state samples $\{\tilde{x}_t^{f(m)}\}_{m=1}^{M}$, measurement mapping $H_t = H_t(\bar{\theta}_{t,y})$, and measurement variance $V_t = V_t(\bar{\theta}_{t,y})$.

[a] The $l_t(\cdot)$ term is defined in Equation (5.5).

Algorithm 5.2: *EnKF-APF: Update step when the measurement parameter is fixed*

**Another connection: Rao-Blackwellised particle filtering**

Up to this point, we only focused on the relationship of the EnKF-APF algorithm to the APF. We make a connection to another filtering algorithm: Rao-Blackwellised particle filtering. Suppose that the hidden state $X_t$ is divided into two groups $X_t^{(1)}$ and $X_t^{(2)}$ with $p(x_t \mid x_{t-1}) = p(x_t^{(1)} \mid x_{t-1}^{(1)}, x_{t-1:t}^{(2)})p(x_t^{(2)} \mid x_{t-1}^{(2)})$. Further suppose that the posterior distribution $p(x_{0:t}^{(1)} \mid x_{0:t}^{(2)}, y_{1:t})$ is analytically tractable for collected measurements $y_{1:t}$. Rao-Blackwellised particle filtering is an algorithm to marginalize out the part of the state that is more analytically tractable, $X_t^{(1)}$, and then apply a particle filtering algorithm to sample and then calculate a lower variance estimator of the state with reduced dimensions, $X_t^{(2)}$ (Chen, 2003; Doucet et al., 2000). The algorithm is based on the Rao-Blackwell theorem, which states that the expected value of an unbiased estimator conditioned on its sufficient statistic has a lower variance than the original estimator (see Casella and Berger (2002, Theorem 7.3.17) for a formal statement of the theorem). With the EnKF-APF, there is a natural division of the hidden variables: the parameter $\Theta_t$ and the state $X_t$. The first-stage of the algorithm is Rao-Blackwellised particle filtering of the parameter $\Theta_t$. However, unlike Rao-Blackwellised particle filtering, the posterior distribution of the marginalized hidden variable, the often high-dimensional state $X_t$, is not analytically tractable and thus the more practically effective EnKF is used to sample $X_t$ in the second-stage of the algorithm.

## 5.4 Update step: when estimating the measurement parameter

Now that we have an algorithm to update the state parameter $\Theta_{t,x}$, it is easy to construct an algorithm to update both the state and measurement parameter. The algorithm constructed in this section is a slight modification of Algorithm 5.2, which updates $\Theta_{t,x}$ when $\Theta_{t,y}$ is fixed. The estimation of the measurement parameter introduces computational expenses that are undesirable, e.g., the calculation of different Kalman gains. We briefly discuss this situation before constructing an algorithm that reduces that computational expense.

### 5.4.1 With different Kalman gains

Using a similar factorization of the joint analysis distribution as in Equation (5.9), we have

$$\pi_{0:t}(x_t, \theta_t) = \frac{\phi[y_t; H_t(\theta_{t,y})x_t, V_t(\theta_{t,y})]}{p(y_t \mid \theta_t, y_{0:t-1})} \times \frac{p(y_t \mid \theta_t, y_{0:t-1})}{p(y_t \mid y_{0:t-1})} p(x_t, \theta_t \mid y_{0:t}). \quad (5.14)$$

This factorization is quite similar to Equation (5.9) and suggests that Algorithm 5.2 is similarly applicable. However, since the measurement parameter is being sampled at every timestep, each state sample $x_t^{f(m)}$ is updated with a different parameter sample $\theta_{t,y}^{(m)}$ as opposed to the same parameter $\bar{\theta}_{t,y}$ as in Section 5.3.2. This consequently results in $M$ evaluations of $H_t(\theta_{t,y}^{(m)})$ and $V_t(\theta_{t,y}^{(m)})$, leading to other computationally expensive modifications to Algorithm 5.2. After deriving estimators in a similar manner as Section 5.3.2, there are two important changes to the estimators that come with undesirable modifications to the algorithm:

1. The parameters are resampled with importance weights proportional to $l_t(x_t^{f(m)}, \theta_{t,y}^{f(m)}, \hat{\Sigma}_t^f)$ as opposed to $l_t(x_t^{f(m)}, \bar{\theta}_{t,y}, \hat{\Sigma}_t^f)$ (see Step 1a of Algorithm 5.2). Examining the equation of $l_t(\cdot)$ (Equation (5.5)), the evaluation of the weight for each parameter sample requires $M$ calculations of the variance $V_t(\theta_{t,y}^{f(m)}) + H_t(\theta_{t,y}^{f(m)})\hat{\Sigma}_t^f H_t^T(\theta_{t,y}^{f(m)})$ for each $m = 1, .., M$ as opposed to the calculation of one calculation of the variance $V_t(\bar{\theta}_{t,y}) + H_t(\bar{\theta}_{t,y})\hat{\Sigma}_t^f H_t^T(\bar{\theta}_{t,y})$ for all samples.

2. After resampling the forecast parameter particles to generate analysis parameter particles $\{\theta_t^{a(m)}\}_{m=1}^M$, each state sample is updated as

$$x_t^{a(m)} = \tilde{x}_t^{f(m)} + K_t(\theta_{t,y}^{a(m)}, \hat{\tilde{\Sigma}}_t^f)[y_t - H_t(\theta_{t,y}^{a(m)})\tilde{x}_t^{f(m)}].$$

   Compare this equation with Equation (5.13) from Section 5.3.2: instead of updating each state sample with the same Kalman gain $K_t(\bar{\theta}_{t,y}, \hat{\tilde{\Sigma}}_t^f)$, each sample is updated with a different Kalman gain $K_t(\theta_{t,y}^{a(m)}, \hat{\tilde{\Sigma}}_t^f)$, requiring $M$ calculations of the Kalman gain. Not only is the calculation of the Kalman gain computationally expensive, it is not ideal for plug-and-play, since existing implementations of the EnKF use the same Kalman gain for each state sample.

Frei and Künsch (2012) faced a similar issue in their development of a hybrid filter that uses the EnKF to filter the states and the PF to estimate a

measurement noise parameter. They circumvented the problem by replacing the relevant parts of the Kalman gain with a point estimator. With parameter samples $\{(\theta_t^{(m)}, w_t^{(m)})\}_{m=1}^M$ from $p(\theta_t \mid \cdot)$, they suggested replacing $V_t(\theta_t^{(m)})$ of the Kalman gain $K_t(\theta_t^{(m)}, \cdot)$ (see Equation (5.4)) with a point estimator[2], i.e.,

$$V_t(\bar{\theta}_t) \equiv V_t\left(\sum_{m=1}^M w_t^{(m)}\theta_t^{(m)}\right) \quad \text{or} \quad \overline{V_t(\cdot)} \equiv \sum_{m=1}^M w_t^{(m)} V_t(\theta_t^{(m)}).$$

While a practical solution to the problem, a bias is introduced when updating with only a point estimator, which they do not correct for in their algorithm. We do think their approach is promising and develop a similar algorithm that uses a point estimator, which allows for the usage of the same Kalman gain, *and* corrects for the bias in the next section.

### 5.4.2 With the same Kalman gain

The computational expense is reduced by replacing posterior samples of $\Theta_{t,y}$ with a point estimator $\bar{\theta}_{t,y}^f$. Consequently, the updated samples are biased towards the point estimator, so the bias is corrected similar to the sequential updating algorithm of Section 5.3.2. Let $\bar{\theta}_{t,y}^f$ denote a point estimator of $p(\theta_{t,y} \mid y_{0:t-1})$. Before delving into the details, we introduce notation to distinguish between when the point estimator is used and when it is not. Denote the parameter with the point estimator as $\tilde{\Theta}_t = (\Theta_{t,x}, \bar{\theta}_{t,y}^f)$. Consequently, $\tilde{\theta}_t \equiv (\theta_{t,x}, \bar{\theta}_{t,y}^f)$ and the forecast samples are $\tilde{\theta}_t^{f(m)} \equiv (\theta_{t,x}^{f(m)}, \bar{\theta}_{t,y}^f)$.

As mentioned in Section 5.3.2, one way to view the equation that summarizes the sequential update algorithm is the joint analysis distribution multiplied by an special identity. Instead of using the identity based on the conditional predictive likelihood $p(y_t \mid \theta_t, y_{0:t-1})$ as in Section 5.3.2, the identity is based on the predictive likelihood conditioned on the point estimator of the measurement parameter, i.e., $p(y_t \mid \tilde{\theta}_t, y_{0:t-1})$. Thus, the joint analysis distribution is

$$\pi_{0:t}(x_t, \theta_t) = \frac{\phi[y_t; H_t(\theta_{t,y})x_t, V_t(\theta_{t,y})]}{p(y_t \mid \tilde{\theta}_t, y_{0:t-1})} \times \frac{p(y_t \mid \tilde{\theta}_t, y_{0:t-1})}{p(y_t \mid y_{0:t-1})} p(x_t, \theta_t \mid y_{0:t}).$$

However, like the algorithm in Section 5.3.2, updating with an estimator of this equation will only update the state parameter and not update the measurement parameter since the predictive likelihood is conditioned on the point estimator $\bar{\theta}_{t,y}^f$. To update the measurement parameter, the equation is multiplied by

[2]See discussion at the end of Section 4b in their paper.

another identity: an identity constructed with the measurement density conditioned on the point estimator $\bar{\theta}^f_{t,y}$, i.e., $g(y_t \,|\, x_t, \bar{\theta}_{t,y}) = \phi[y_t; H_t(\bar{\theta}^f_{t,y})x_t, V_t(\bar{\theta}^f_{t,y})]$. There are a couple of ways to reshuffle the identity in a way that is illustrative to constructing the algorithm.

One way to update the measurement parameter is via this particular factorization of the joint analysis distribution:

$$\pi_{0:t}(x_t, \theta_t) = \underbrace{\frac{\phi[y_t; H_t(\theta_{t,y})x_t, V_t(\theta_{t,y})]}{\phi[y_t; H_t(\bar{\theta}^f_{t,y})x_t, V_t(\bar{\theta}^f_{t,y})]}} \times \frac{\phi[y_t; H_t(\bar{\theta}^f_{t,y})x_t, V_t(\bar{\theta}^f_{t,y})]}{p(y_t \,|\, \tilde{\theta}_t, y_{0:t-1})} \times \frac{p(y_t \,|\, \tilde{\theta}_t, y_{0:t-1})}{p(y_t \,|\, y_{0:t-1})} p(x_t, \theta_t \,|\, y_{0:t}). \tag{5.15}$$

Ignoring the term above the underbrace, notice that the factorization is the same as Equation (5.9) from the sequential update of the state and parameters when $\Theta_{t,y}$ is fixed. For this reason, this particular factorization suggests a *three*-stage algorithm: the two-stage algorithm of Algorithm 5.2 with an additional third stage. Specifically, sample pairs of state and parameter samples via Algorithm 5.2, thus updating only the state and the state parameter, and then the additional third stage resamples those samples with importance weights proportional to the term above the underbrace to update the measurement parameter. Although resampling is a necessary step in PF algorithms, it is an undesirable step that can lead to particle collapse, so adding *another* resampling step to the algorithm will only exacerbate the problem. Furthermore, our goal is to *avoid* resampling the state samples, which this particular modification of the algorithm fails to achieve.

Let's examine another factorization of the joint analysis distribution:

$$\pi_{0:t}(x_t, \theta_t) = \frac{\phi[y_t; H_t(\bar{\theta}^f_{t,y})x_t, V_t(\bar{\theta}^f_{t,y})]}{p(y_t \,|\, \tilde{\theta}_t, y_{0:t-1})} \times \underbrace{\frac{\phi[y_t; H_t(\theta_{t,y})x_t, V_t(\theta_{t,y})]}{\phi[y_t; H_t(\bar{\theta}^f_{t,y})x_t, V_t(\bar{\theta}^f_{t,y})]}} \frac{p(y_t \,|\, \tilde{\theta}_t, y_{0:t-1})}{p(y_t \,|\, y_{0:t-1})} p(x_t, \theta_t \,|\, y_{0:t}). \tag{5.16}$$

Again, ignoring the term above the underbrace, the factorization is the same as Equation (5.9) when $\Theta_{t,y}$ is fixed. This particular factorization suggests resampling the parameter particles with weights proportional to the conditional predictive likelihood $p(y_t \,|\, \tilde{\theta}_t, y_{0:t-1})$—as in the first-stage of Algorithm 5.2—*in addition to* the term above the underbrace. Specifically, each parameter

**Input**:
- $\{x_{t-1}^{a(m)}\}_{m=1}^{M} \dot{\sim} p(x_{t-1} \,|\, y_{0:t-1})$
- $\{(x_t^{f(m)}, \theta_t^{f(m)})\}_{m=1}^{M} \dot{\sim} p(x_t, \theta_t \,|\, y_{0:t-1})$ from Algorithm 5.1

**Output**: $\{(x_t^{a(m)}, \theta_t^{a(m)})\}_{m=1}^{M} \dot{\sim} p(x_t, \theta_t \,|\, y_{0:t})$

1. Update $\Theta_t$:

   a) Calculate the point estimator $\bar{\theta}_{t,y}^{f}$ from samples $\{\theta_{t,y}^{f(m)}\}_{m=1}^{M}$.

   b) Calculate unnormalized weights[a]:

$$\tilde{w}_t^{(m)} = \frac{\phi[y_t; H_t(\theta_{t,y}^{f(m)}) x_t^{f(m)}, V_t(\theta_{t,y}^{f(m)})]}{\phi[y_t; H_t(\bar{\theta}_{t,y}^{f}) x_t^{f(m)}, V_t(\bar{\theta}_{t,y}^{f})]} l_t(x_t^{f(m)}, \bar{\theta}_{t,y}^{f}, \hat{\Sigma}_t^{f}),$$

   for each $m = 1, ..., M$, where $\hat{\Sigma}_t^{f}$ is the (tapered) sample covariance calculated from $\{x_t^{f(m)}\}_{m=1}^{M}$. Normalize weights: $w_t^{(m)} = \tilde{w}_t^{(m)} / \sum_{n=1}^{M} \tilde{w}_t^{(n)}$ for each $m = 1, ..., M$.

   c) Sample

$$\theta_t^{a(m)} \sim \sum_{m=1}^{M} w_t^{(m)} \delta_{\theta_t^{f(m)}}(\theta_t)$$

   for each $m = 1, ..., M$.

2. Update $(X_t, \Theta_t)$:

   a) Sample $\tilde{x}_t^{f(m)} \sim f(x_t \,|\, x_{t-1}^{a(m)}, \theta_t^{a(m)})$.

   b) Sample $\{x_t^{a(m)}\}_{m=1}^{M}$ using Step 3 of Algorithm 2.1 with forecast state samples $\{\tilde{x}_t^{f(m)}\}_{m=1}^{M}$, measurement mapping $H_t = H_t(\bar{\theta}_{t,y}^{f})$, and measurement variance $V_t = V_t(\bar{\theta}_{t,y}^{f})$.

[a] The $l_t(\cdot)$ term is defined in Equation (5.5).

Algorithm 5.3: *EnKF-APF: Update step when estimating the measurement parameter*

sample is updated with importance weights

$$w_t^{(m)} \propto \frac{\phi[y_t; H_t(\theta_{t,y}^{f(m)})x_t^{f(m)}, V_t(\theta_{t,y}^{f(m)})]}{\phi[y_t; H_t(\bar{\theta}_{t,y}^{f})x_t^{f(m)}, V_t(\bar{\theta}_{t,y}^{f})]} l_t(x_t^{f(m)}, \bar{\theta}_{t,y}^{f}, \hat{\Sigma}_t^{f}).$$

Not only does the term above the underbrace update the measurement parameter, it allows for the construction of an algorithm that uses the same Kalman gain $K_t(\bar{\theta}_{t,y}^{f}, \cdot)$ in the EnKF update of the state samples, i.e.,

$$x_t^{a(m)} = \tilde{x}_t^{f(m)} + K_t(\bar{\theta}_{t,y}^{f}, \hat{\tilde{\Sigma}}_t^{f})[y_t - H_t(\bar{\theta}_{t,y}^{f})\tilde{x}_t^{f(m)}].$$

We do not derive the estimators of the filtering densities since they are similarly derived as in Section 5.3.2. Algorithm 5.3 summarizes the algorithm. Notice that the algorithm is exactly the same as Algorithm 5.2, except that the parameters are resampled with different weights (Step 1b).

## 5.5 Demonstrations

In this section, we apply the EnKF-APF algorithm to estimate static parameters. First, we complete the demonstration from Section 4.5 by applying the EnKF-APF with the same settings outlined in that section. We compare the results and demonstrate that the EnKF-APF is able to better estimate parameters than the EnKF. To demonstrate that EnKF-APF does not suffer from the dimensionality problem faced by many PFs, we apply the EnKF-APF algorithm to a high-dimensional state-space model—one with the Lorenz 2005-II as a state transition model.

### 5.5.1 Lorenz 1963

The EnKF-APF is applied with the same settings as in Section 4.5 to estimate $\sigma$; Figure 5.1 shows the results. In every panel, including without inflation, we see what we expect: the true value is better captured with larger ensemble sizes. Like the EnKF, EnKF-APF has a harder time capturing the true value when some aspect of the $y$-direction is not measured (i.e., $i = 3, 4, 5$). However, unlike the EnKF, any difficulty is remedied with larger ensemble sizes. While some estimated values are near the bounds of the constraint range, none fall outside the pre-specified constraints. There are a few runs that failed to complete the parameter estimation procedure, but they only occur for the "harder" measurement models (i.e., $i = 1, 2$) with small ensemble sizes and the default settings for the inflation values. These are the same situations in

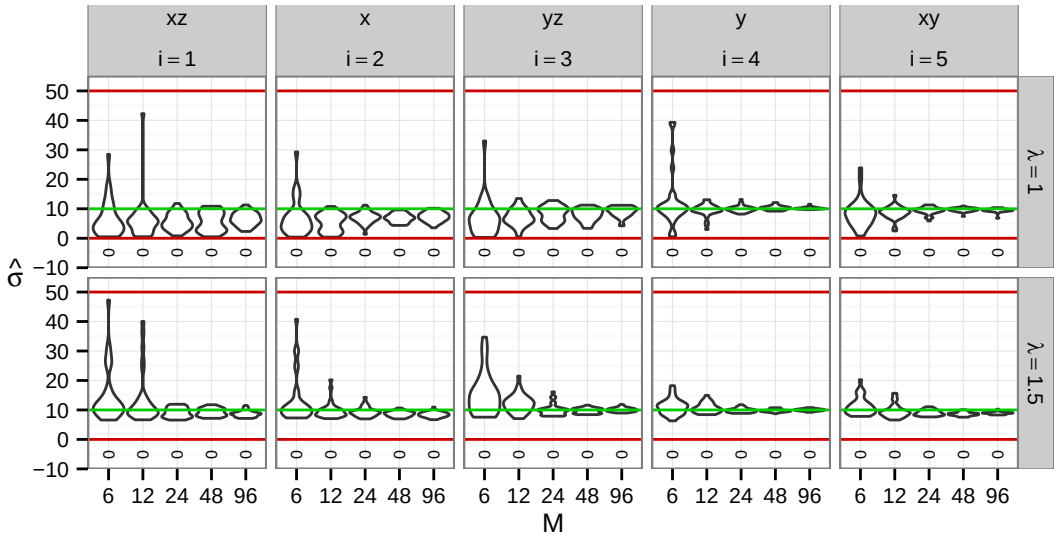
which the EnKF failed to complete the parameter estimation procedure, even with larger ensemble sizes! However, the failures are not a problem for the EnKF-APF with larger ensemble sizes. The resampling step in the first-stage of the EnKF-APF algorithm is the main reason behind the reduction in failures between the EnKF-APF and the EnKF. With the EnKF-APF, the proposed parameters may lead to numerical instabilities but the resampling step from the first stage of the algorithm removes those parameters and hence reduces the number of failures in the estimation procedure. Therefore, forecasting the state twice is worth the extra computational expense.

Figure 5.2 shows selected comparisons between the EnKF and EnKF-APF. With every inflation setting, including no inflation, the EnKF-APF is able to capture the true value more effectively than the EnKF for all measurement models and almost all ensemble sizes. The places where the EnKF-APF is not as effective as EnKF occur for the "easier" measurement models with small ensemble sizes, e.g., $i = 3, 4, 5$ with $M = 6, 12$, too small for any PF to be effective.

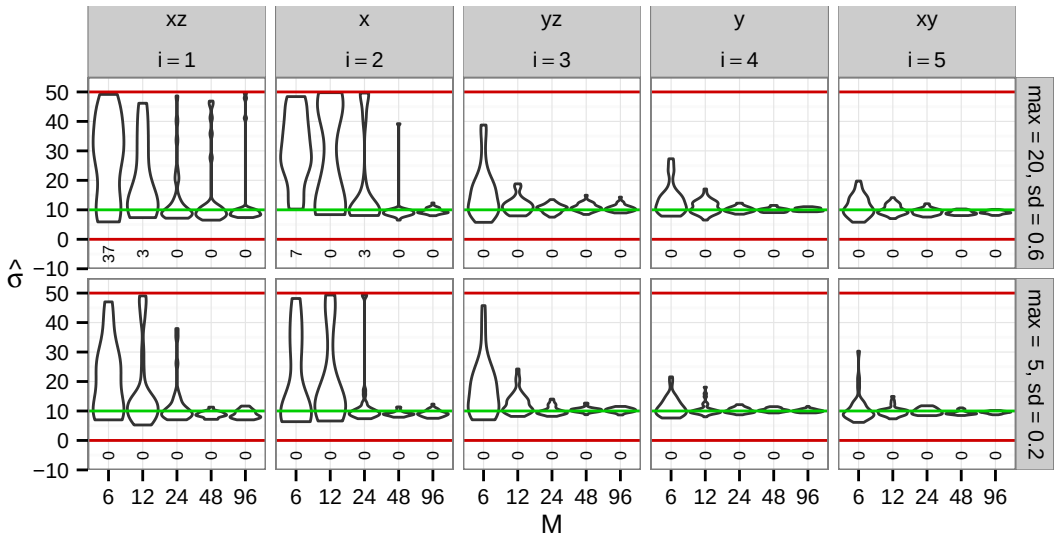
### 5.5.2 Lorenz 2005

While the EnKF-APF has performed well for the Lorenz 1963 system, good results are not observed until an ensemble size of 48 is used for many measurement models—an ensemble size 16 times larger than the state dimensions! If we extrapolated this to high-dimensional systems, the ensemble size needed for good results quickly becomes computationally infeasible, thus one may conclude that the EnKF-APF is not a practical method for parameter estimation. In this section, we demonstrate that the extrapolation is unfounded and show that the EnKF-APF is perfectly capable of capturing true parameter values of the Lorenz 2005-II system for small ensemble sizes, oftentimes better than the EnKF. Furthermore, the previous demonstration with the Lorenz 1963 system only illustrates the ability of EnKF-APF to estimate state parameters, i.e., Algorithm 5.2. We additionally estimate a measurement parameter with the Lorenz 2005 system, thus demonstrating the full capabilities of Algorithm 5.3.

Before discussing the results, we detail the settings used in the parameter estimation algorithm. Like the Lorenz 1963 demonstration, all parameter estimation runs have the same underlying data generation process; only samples from the initial condition and the initial parameter values to start the algorithm are different between runs.

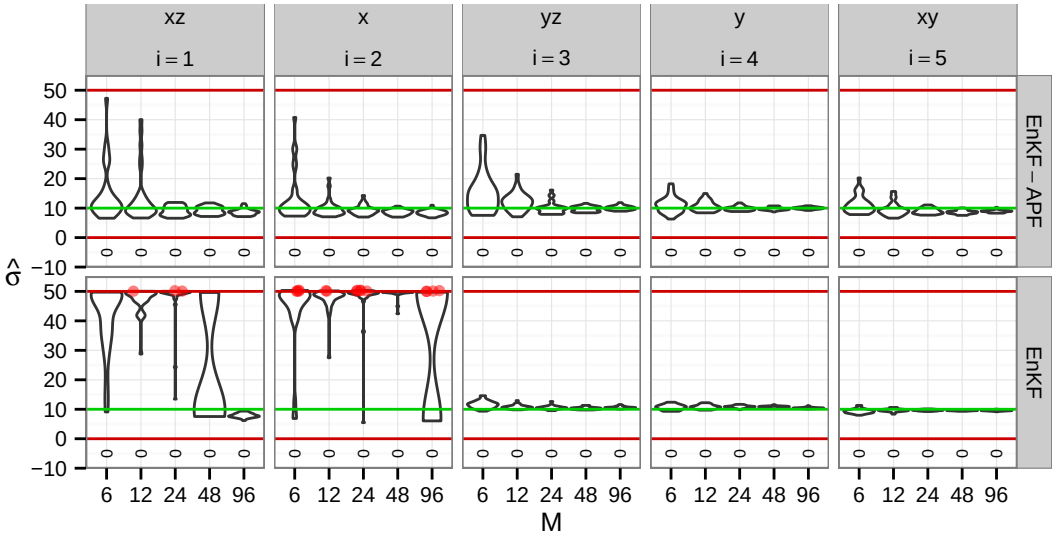



(a) *Fixed inflation.*

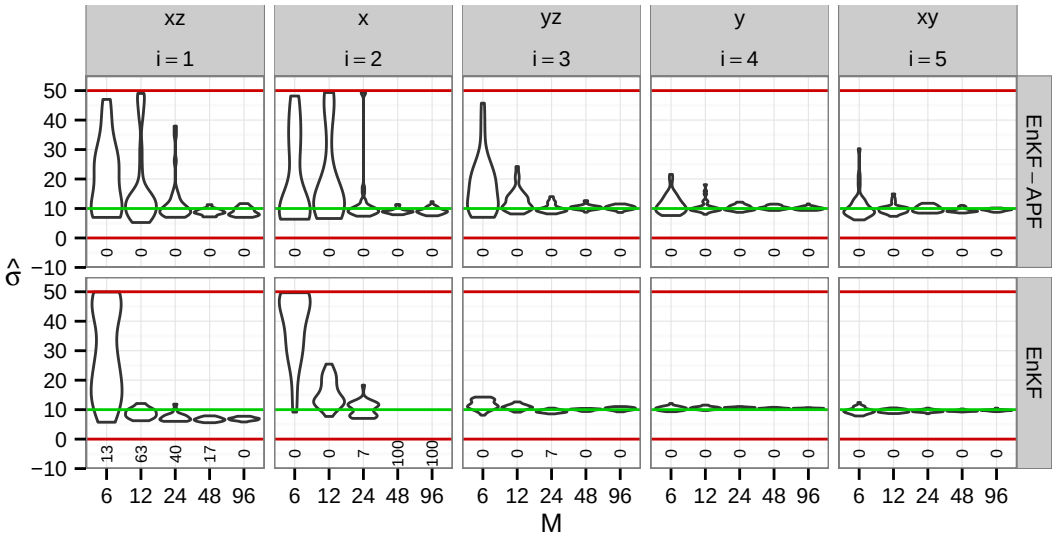



(b) *Estimated inflation.*

Figure 5.1: *Lorenz 1963: distribution of $\hat{\sigma}$ as estimated by EnKF-APF.* The settings here are the same as in Figure 4.5 except parameter estimation is performed with EnKF-APF.




(a) *Fixed inflation,* $\lambda = 1.5$.




(b) *Estimated inflation with a maximum value of 5 and an initial standard deviation of 0.2.*

Figure 5.2: *Lorenz 1963: comparison of the distributions of* $\hat{\sigma}$ *as estimated by EnKF-APF and EnKF.* The settings here are the same as in Figure 4.5.

### State-space model and data generation

Examining the Lorenz 2005-II equations (Equation (1.4)), there are three state parameters, i.e., $N$, $K$, and $F$. The parameter $N$ controls resolution of the mathematical grid. In practice, the resolution is chosen based on computational resources. For this reason, we set $N$ to be the default value of 960 (and thus the state dimension is $d_x = 960$) and aim to estimate the parameters $K$ and $F$. Let $R = (r_1, ..., r_{d_x})$ be the vector of the state locations equispaced on a spherical latitude band as illustrated in Figure 1.6, where $r_i$ is the location of the $i$th state located at an angle of $\frac{(i-1)}{d_x} \times 2\pi$ for all $i = 1, ..., d_x$.

When not estimating an inflation value, the state transition model is

$$x_{t_n} = m_{t_n}\left(x_{t_{n-1}}; K, F\right);$$

and when estimating an inflation value, the state transition model is

$$x_{t_n} = \mu_{t_n} + \lambda\left[m_{t_n}\left(x_{t_{n-1}}; K, F\right) - \mu_{t_n}\right].$$

The discrete state transition model $m_{t_n}$ is defined to be

$$m_{t_n}(\cdot; K, F, dt) = \mathcal{I}_W(\cdot; K, F) + O(e(dt))$$

where $W$ is the continuous state transition model for the Lorenz 2005-II system, i.e., Equation (1.7), and $e(\cdot)$ and $dt$ are defined in Section 1.2.1. Depending on whether the inflation value is estimated or not, the state parameter being estimated is either $\theta_x = (K, F, \lambda)$ or $\theta_x = (K, F)$, respectively.

The measurement model is

$$w_{t_n} = H(S_i, \gamma_i)x_{t_n} + \epsilon_{t_n}, \quad \epsilon_{t_n} \sim \mathcal{N}(0, U_i),$$

where the measurement mapping $H(s_i, \gamma_i)$ depends on the locations $S_i$ where the measurements are collected and the halfwidth parameter $\gamma_i$. Recall that the linear measurement mapping is a matrix of weights that interpolates the state values to the measurement values: state locations that are closer to the measurement location are assigned higher weights than those further away. The halfwidth parameter $\gamma_i$ controls the distance at which the weights are zero: specifically, the weights are zero for distances beyond $2\gamma_i$. The weights of the measurement mappings are constructed with the Gaspari-Cohn function (Gaspari and Cohn, 1999) with five interpolating points.

We choose three measurement mappings. The first measurement mapping is the fully observed system, where the measurement mapping is simply the identity matrix $I_{d_x}$. In this case, the measurement locations exactly match the

state locations, i.e., $S_1 = R$, and there is no interpolation between the state and measurement locations, i.e., $\gamma_1 = 0$. The next two measurement mappings are partially observed systems. The first partially observed system measures every other state location, i.e., $S_2 = (r_1, r_3, r_5, ..., r_{d_x-1})$, without interpolation, i.e., $\gamma_2 = 0$. Specifically, its measurement mapping is a $(d_x/2) \times d_x$ matrix that takes the following form:

$$H(s_2, \gamma_2) \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

The second partially observed model has $d_x/2$ measurement locations randomly chosen on the latitude band via a uniform distribution, i.e., the $i$th element of $S_3$ is located at an angle of $U_i \times 2\pi$ where $U_i \sim \mathrm{U}(0, 1)$ for all $i = 1, ..., d_x/2$, and the halfwidth is chosen to be $\gamma_3 = \frac{5}{960} \times 2\pi$. The variance of the measurement noise is taken to be the identity, i.e., $U_1 = I_{d_x}$ and $U_2 = U_3 = I_{d_x/2}$.

The initial condition $x_{t_0}$ is generated by taking a random state from a uniform distribution and integrating it forward with a large $\Delta t$. The state $x_{t_n}$ is generated by taking the initial condition $x_{t_0}$ and integrating it sequentially to collect the equivalent of one year's worth of measurements with a measurement collected once per day, i.e., $N = 365$ times with a forecast lead time of $\Delta t_n = 0.24$ for all $n$. Samples from the initial condition are generated from a standard multivariate normal distribution with mean $x_{t_0}$.

### EnKF settings

We apply the square-root filter with localization. The localization halfwidth is $\frac{5}{960} \times 2\pi$ using the Gaspari-Cohn function to construct the tapering matrix. Ensemble sizes are set to be 51, 101, and 201. Parameter estimation is run both with and without inflation. When inflating, the forecast ensemble is inflated in two ways: with a fixed inflation of $\lambda = 1.5$ and with inflation values estimated with the default DART settings as described in Section 4.5.

### Parameter estimation settings

We apply the iterated filter from Ionides et al. (2015) with both the EnKF-APF and EnKF to estimate parameters. We are interested in estimating the state parameters $K$ and $F$ and the measurement parameter $\gamma_3$. However, the

parameter $\gamma_3$ is only estimated with the EnKF-APF and is fixed with the EnKF. We choose the perturbation density $q(\theta_{t_n} \mid \theta_{t_{n-1}})$ to be a multivariate truncated normal distribution with mean $\theta_{t_{n-1}}$. Table 5.1 lists the initial standard deviations and the constraints. The correlation between the parameters is set to be zero, thus the multivariate truncated normal distribution is simply a product of univariate truncated normal distributions. To respect the integer-valued nature of $K$, proposals are rounded to the nearest integer.

| parameter | truth | initial sd | minimum | maximum |
|---|---|---|---|---|
| $K$ | 32 | 0.5 | 1 | 100 |
| $F$ | 15 | 0.5 | -50 | 50 |
| $\lambda$ | 1 | 0.6 | 1 | 20 |
| $\gamma_3$ | $\frac{5}{d_x}\pi$ | $\frac{1}{d_x}\pi$ | 0 | $\pi$ |

Table 5.1: *Lorenz 2005-II parameter estimation settings.* The perturbation density $q(\theta_{t_n} \mid \theta_{t_{n-1}})$ is a product of truncated normal distributions with the above parameters.

Each run of the iterated filter is initialized with the true value perturbed by the initial standard deviation multiplied by two and an inflation value of one. The iterated filter is run with 25 iterations and the MLE is reported to be the median of the final ensemble.

### Results

The results without inflation are omitted because the parameter estimation procedure failed to capture the true values with both filters. This is as we suspected, even before running the parameter estimation algorithm, since the EnKF requires inflation for good performance when filtering high-dimensional state-spaces. Figures 5.3 and 5.4 show the results with inflation from both the EnKF-APF and EnKF, respectively.

With the EnKF-APF, the results show that the EnKF-APF consistently captures the true parameter values well for all measurement models and ensemble sizes, including the small ensemble size of 51! The state parameter $K$ is captured almost perfectly (top rows of both Figure 5.3a and Figure 5.3b): most MLEs are exactly the true value of 32 with a few MLEs with values of 31, 33, or 34—values very close to the true value. The state parameter $F$ is often captured well (middle rows of both Figure 5.3a and Figure 5.3b): the range

of MLEs cover the the true value. The range of estimated values of $F$ tend to be larger with fixed inflation than with estimated inflation. Similarly, the measurement parameter $\gamma_3$ is captured better with estimated inflation than with fixed inflation (bottom rows of both Figure 5.3a and Figure 5.3b). In fact, with estimated inflation, the distribution of the estimated values of $\gamma_3$ show the characteristic decrease in the range of estimated values with larger ensemble sizes, indicating that parameter estimation for the Lorenz 2005-II system is better performed when the inflation value is estimated. The ability of the EnKF-APF to estimate the halfwidth parameter $\gamma_3$ demonstrates its full capabilities: the algorithm is able to successfully estimate both state *and* measurement parameters.

On the other hand, the results show that the EnKF is generally able to capture the true parameter values but it does sometimes have trouble. Oddly, the EnKF is able to better capture the true parameter values in partially observed systems (measurement models $i = 2, 3$ in the latter two columns of Figure 5.4) than the fully observed system ($i = 1$, first column of Figure 5.4). We suspect this oddity is due to the sensitivity of the EnKF linear update: the parameter is updated by the conditional correlations between the state and the parameter that are perhaps unimportant. While there were no failures in the parameter estimation procedure, a few runs estimated values of $K$ to be outside of its pre-specified constraints.

Figure 5.5 compares the results from the EnKF-APF and EnKF. The plots in this figure are zoomed-in versions of the previously examined figures, i.e., Figures 5.3b and 5.4b, omitting the results for the measurement model $i = 3$. The last measurement model is omitted because the results are not comparable between the two filters: $\gamma_3$ is estimated with the EnKF-APF and is fixed with the EnKF. For the fully observed system ($i = 1$, first column of Figure 5.2), the EnKF-APF clearly captures the true parameter values of $K$ and $F$ better than the EnKF. For the partially observed system ($i = 2$, second column of Figure 5.2), the EnKF-APF better captures the true parameter value of $K$ and both filters capture $F$ similarly. Overall, the EnKF-APF is better at capturing the true parameter values than the EnKF.

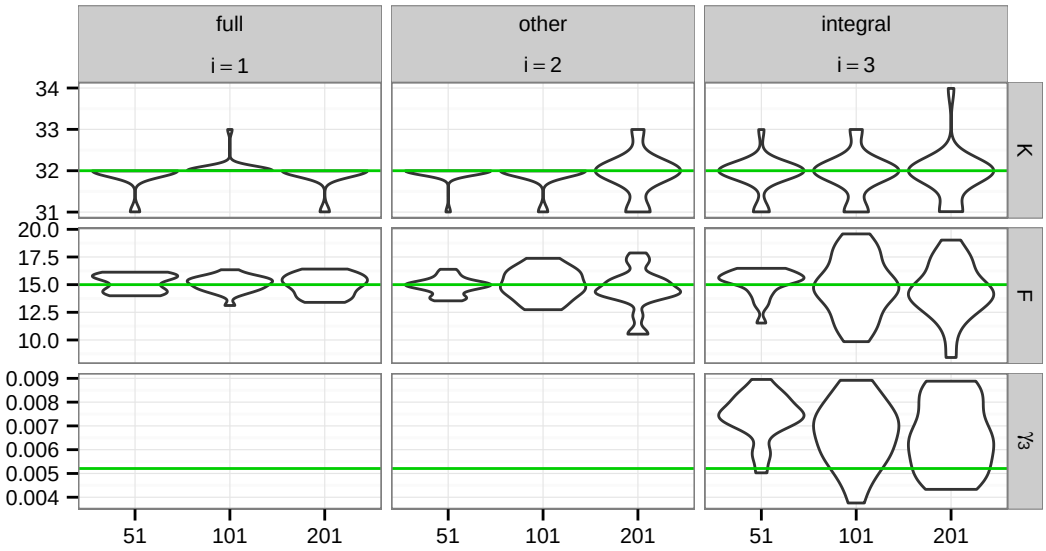
## 5.6 Limitations

In this chapter, we combined the best qualities of the EnKF and the PF to develop methodology for estimating parameters with the method of artificial evolution of parameters. We have demonstrated that the EnKF-APF better captures true parameter values than the EnKF in both low- and high-

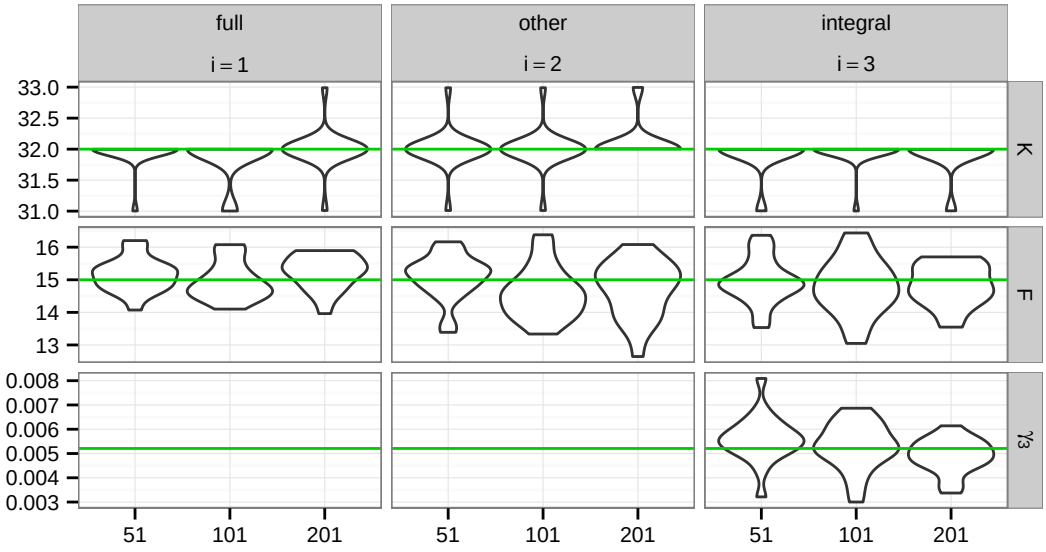
dimensional state-spaces. Since the EnKF-APF uses the EnKF algorithm to update the state, it does not suffer from particle collapse like the PF when applied to high-dimensional state-spaces. Its versatility and plug-and-play quality allows it to be easily added to existing implementations of EnKF, such as DART.

By its combined nature, the EnKF-APF algorithm not only shares the successes of the EnKF in filtering states and of the PF in filtering parameters, but it also shares the limitations of both filters. Like the PF, the EnKF-APF will require an impractically large ensemble size to estimate parameters when the parameter space is large. For practical reasons, the practitioner is forced to apply the EnKF to estimate parameters, which assumes that the parameters are conditionally correlated with the state. As seen with the Lorenz 1963 system (Figure 5.2, measurement models $i = 1, 2$), that assumption may not be true and thus the EnKF will have trouble estimating parameters. However, if the assumption is applicable (Figure 5.2, measurement models $i = 3, 4, 5$), the EnKF will be more efficient, in terms of ensemble size required, than the EnKF-APF, even if the parameter space is not high-dimensional. The ensemble size required to sufficiently estimate the likelihood and thus the parameters is answered in the PF literature (Doucet et al., 2014).

Furthermore, like the EnKF, the EnKF-APF will not estimate the state well when the state transitions are "highly" nonlinear and consequently will not estimate parameters well. In this case, the PF will perform better than the EnKF-APF, but there is currently not a good answer to the degree of nonlinearity when one performs better than the other. The EnKF must be studied more rigorously to answer such a question and we hope the connection in Chapter 3 will be helpful in answering those types of questions. With these caveats in mind, we recommend applying the EnKF-APF when the parameter space is low-dimensional and when the EnKF has been previously demonstrated to be effective for the state-space model being studied.

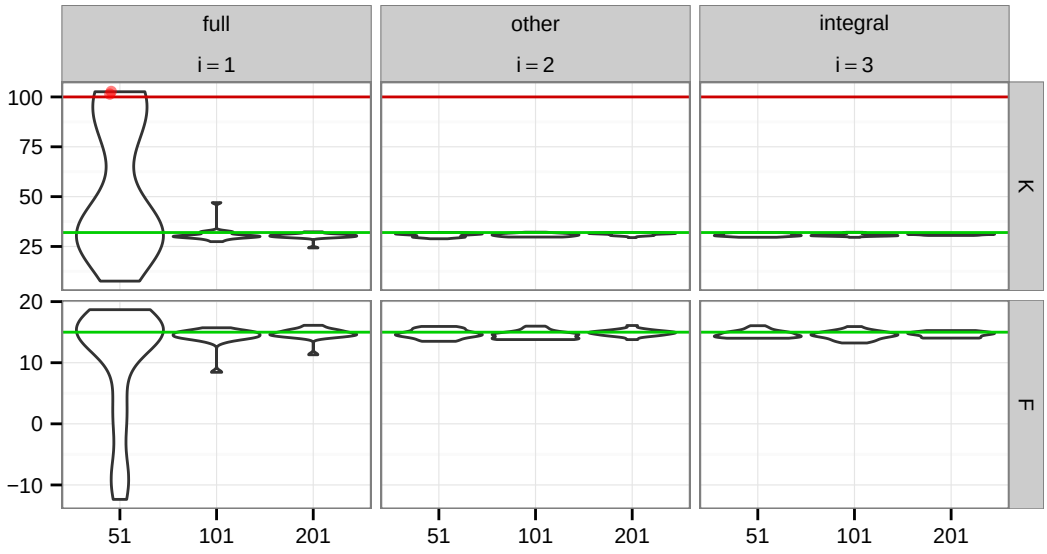



(a) *Fixed inflation,* $\lambda = 1.5$

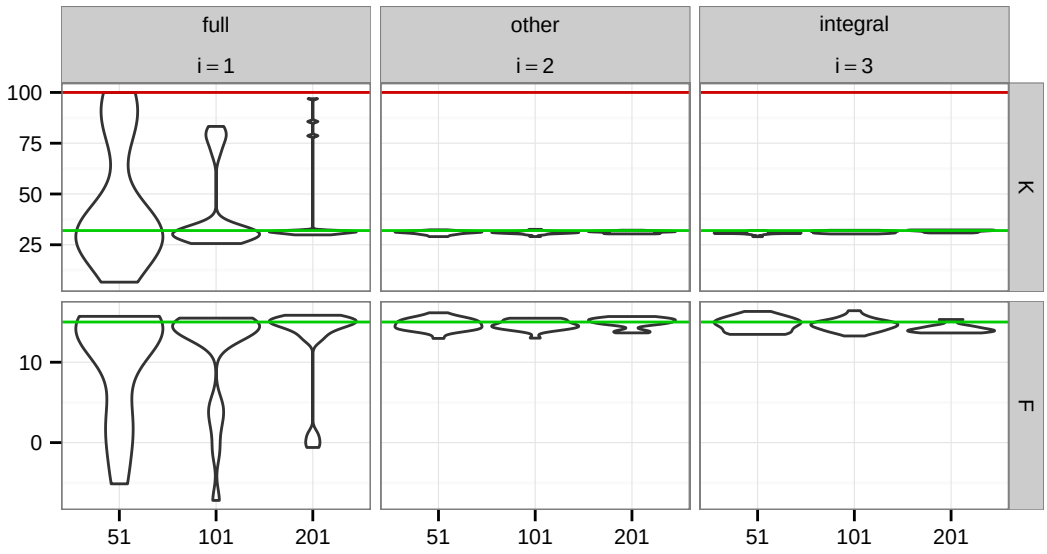



(b) *Estimated inflation*

Figure 5.3: *Lorenz 2005-II: distribution of MLEs as estimated by EnKF-APF.* Each violin plot illustrates the distribution of the MLEs of the parameters labelled to the right of each row. The estimated values come from 20 runs of the iterated filter varied by ensemble size and measurement model. Ensemble size is varied across the $x$-axis of each plot and the measurement model are varied across columns of the plots. Each figure has a different inflation setting as indicated by the caption. The green horizontal line indicates the true parameter value and the red lines, if shown, indicate the constraints of the perturbation density as specified in Table 5.1. A red dot, if shown, is a jittered value of an MLE that is estimated to be outside of the constraints specified.

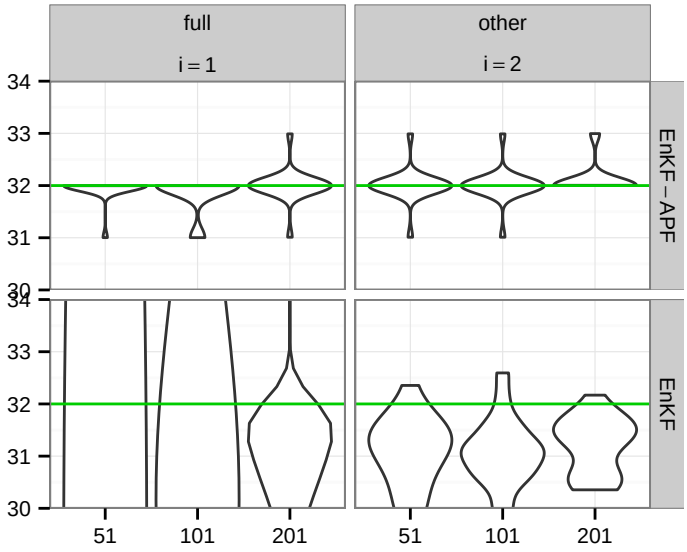



(a) *Fixed inflation,* $\lambda = 1.5$

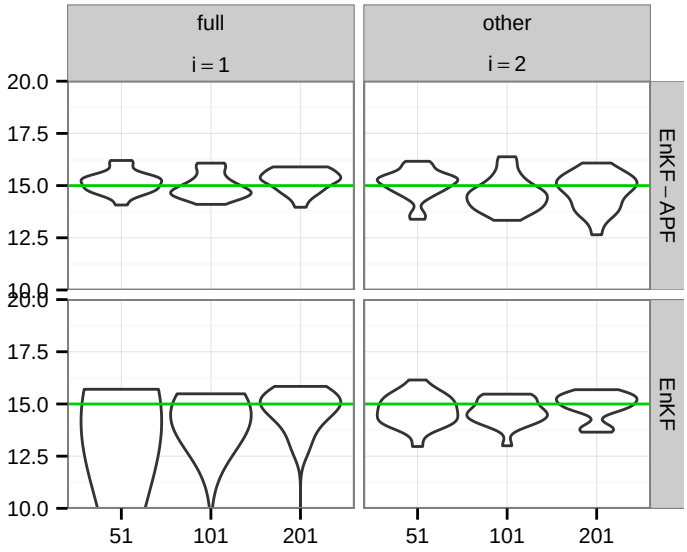



(b) *Estimated inflation*

Figure 5.4: *Lorenz 2005-II: distribution of MLEs as estimated by EnKF.* The settings here are the same as in Figure 5.3.




(a) $K$




(b) $F$

Figure 5.5: *Lorenz 2005-II: comparison of MLEs as estimated by EnKF-APF and EnKF.* The plots are zoomed-in versions of Figures 5.3b and 5.4b (with estimated inflation). Measurement model $i = 3$ is omitted because the results are not comparable between filters: $\gamma_3$ is estimated with EnKF-APF and is fixed with EnKF.

# Part III

# Model error

This page
intentionally
left blank

# 6 Model error: an introduction

With the tool developed in the last part of the dissertation, we are finally able to tackle the original problem we are interested in: model error. In reality, investigators seldom have the true model that governs the systems that they are studying; they can only design models based on their educated, but imperfect, knowledge of the system. Even then, investigators may be unable to implement the model they wish to design for practical reasons, e.g., computational reasons. In this part of this dissertation, we examine the feasibility of correcting model error with a linear correction in the continuous state-transition model. While not a new idea, our contribution is two-fold. First, we demonstrate that a linear correction improves forecasts even when the model error is not linear in the Lorenz 1963 and 2005 systems, both of which have characteristics of atmospheric models. Secondly, we introduce a low-rank approximation to the linear correction that respects the spatial structure prevalent in atmospheric models. This particular parametrization is easily estimated with the parameter estimation algorithm developed in Chapter 5, even when the state is high-dimensional.

## 6.1 Model error: notation and examples

Suppose we are interested in studying the state $X$. The true model of the state's rate of change at time $t$ is:

$$\dot{x}(t) = R[x(t); \theta_R], \tag{6.1a}$$

where $\theta_R$ is the parameter of the system and the dot above $x$ represents the first time-derivative, i.e., $\dot{x}(t) = \frac{dx(t)}{dt}$. The model is named $R$ to remind the reader that it is the "right" model. Measurements of the state $y_{t_n}$ are collected at discrete timepoints $t_n$. Let $\Delta t_n$ be the forecast lead time: the time between

the $(n-1)$-th and $n$th measurements, i.e., $\Delta t_n = t_n - t_{n-1}$. To evaluate the state at the same temporal resolution as the measurements, the model $R$ is integrated as follows:

$$x_{t_n} = m_{t_n}(x_{t_{n-1}}; \theta_R, dt) \tag{6.2}$$

with

$$m_{t_n}(x_{t_{n-1}}; \theta) = x_{t_{n-1}} + \int_{t_{n-1}}^{t_n} \dot{x}(\tau) d\tau.$$

The true model is assumed to be deterministic and thus the state disturbance $\eta_{t_n} = 0$ does not appear in Equation (6.2). The state $x_{t_n}$ is interpolated to the spatial resolution of the measurements as

$$y_{t_n} = H_{t_n} x_{t_n} + \epsilon_{t_n}, \quad \epsilon_{t_n} \sim \mathcal{N}(0, V_{t_n}), \tag{6.3}$$

with a linear mapping $H_{t_n}$ of size $d_y \times d_x$ and measurement noise $V_{t_n}$, both assumed to be known. Recall that Equation (6.3) is called the *measurement model*.

In reality, the true model $R$ is not known, but there is expert knowledge of the evolution of the system. Specifically, the following is a model of the evolution of the approximate state $\hat{X}$:

$$\dot{\hat{x}}(t) = W[x(t); \theta_W] \tag{6.4}$$

where $\theta_W$ is the parameter of the continuous state transition model $W$. This model is denoted $W$ to remind the reader that it is the "wrong" model. The approximate state $\hat{X}$ is integrated to the same temporal resolution as the measurements as follows:

$$\hat{x}_{t_n} = \widehat{m}_{t_n}(x_{t_{n-1}}; \theta_W) \tag{6.5}$$

with

$$\widehat{m}_{t_n}(x_{t_{n-1}}; \theta) = x_{t_{n-1}} + \int_{t_{n-1}}^{t_n} \dot{\hat{x}}(\tau) d\tau.$$

Oftentimes, the integral in the above expression does not have an analytic form and is instead numerically integrated with an appropriate numerical integrator $\mathcal{I}$:

$$\widehat{m}_{t_n}(x_{t_{n-1}}; \theta_W, dt) = \mathcal{I}_W(x_{t_{n-1}}) + O(e(dt)), \tag{6.6}$$

where $dt$ denotes the timestep used by the numerical integrator and $e$ is a function mapping the order of the numerical error, e.g., $e(x) = x$ for forward Euler and $e(x) = x^4$ for fourth-order Runge Kutta. The numerical integrator

$\mathcal{I}$ has a subscript $W$ to remind the reader that it depends on the continuous state transition model $W$. Recall that Equations (6.4) and (6.6) are called the *continuous* and *discrete state transition models*, respectively, and they both make up the *state transition model* of Equation (6.5); see Section 1.2.2 for a detailed discussion.

Since the true model $R$ is not known, the true state cannot be evaluated via Equation (6.2). Instead, the true state $X$ is approximated with the approximate state $\hat{X}$: $x_{t_n} \approx \hat{x}_{t_n}$. This may or may not be a good approximation. We examine the situation when it is not and aim to improve the model $W$ with an improved model $\widetilde{W}$. Assume the error between the true model and the imperfect model is additive:

$$R[x(t); \theta_R] = W[x(t); \theta_W] + D[x(t); \theta_D],$$

where the function $D$ represents the unknown dynamics of the true system not captured by the model $W$, i.e., the *model error*, and $\theta_D$ is the parameter of the continuous state transition model $D$. It is possible to integrate just the continuous state transition model $D$, i.e.,

$$d_{t_n}(x_{t_{n-1}}; \theta_D, dt) = \mathcal{I}_D(x_{t_{n-1}}) + O(e(dt)),$$

but the additivity in the continuous state transition model does not necessarily translate to additivity in the discrete state transition model, i.e.,

$$m_{t_n}(x_{t_{n-1}}; \theta_R, dt) \neq \widehat{m}_{t_n}(x_{t_{n-1}}; \theta_W, dt) + d_{t_n}(x_{t_{n-1}}; \theta_D, dt),$$

particularly for nonlinear $R$ and $W$. The discretization error of the integrator $O(e(dt))$ is assumed to be negligible with respect to the model error and thus we henceforth omit the input $dt$ from the state transition models. To fix ideas, we examine a few examples of model error.

**Example 6.1.1** (Lorenz 2005: model error)**.** Recall that Lorenz specifically constructed the models in his 2005 paper for atmospheric scientists to test their ideas on model error, thus providing an explicit example of model error. Let $z(t)$ be the state of the system with a resolution of $d_z$, i.e., $z(t) \equiv (z_1(t), ..., z_{d_z}(t))$. The true continuous state transition model $R$ is represented by Model III (Equation (1.5)), i.e.,

$$\begin{aligned}\dot{z}(t) &= R[z(t); \theta_R] \\ &= \begin{bmatrix} [x(t), x(t)]_{K_R,1} + b^2[y(t), y(t)]_{1,1} + c[y(t), x(t)]_{1,1} \\ \vdots \\ [x(t), x(t)]_{K_R,d_z} + b^2[y(t), y(t)]_{1,n} + c[y(t), x(t)]_{1,d_z} \end{bmatrix} - x(t) - by(t) + F_R,\end{aligned}$$

where $x(t)$ and $y(t)$ are functions of $z(t)$; the parameter $I$ is defined by Equation (1.5b); and the parameter $\theta_R$ is $(K_R, F_R, I, b, c)$. Model II (Equation (1.4)) captures the large-scale dynamics of Model III, but misses the small-scale dynamics of the system and thus serves as an imperfect, but good, model $W$:

$$W(z(t); \theta_W) = \begin{bmatrix} [z(t), z(t)]_{K_W,1} \\ \vdots \\ [z(t), z(t)]_{K_W,d_z} \end{bmatrix} - z(t) + F_W$$

with $\theta_W = (K_W, F_W)$. The imperfect model $W$ is equivalent to $R$ only when $I = 1$ and the parameters $K_W = K_R$ and $F_W = F_R$. The model error $D$ is quadratic in the state. While there is an expression for $D$, the expression is not simple and does not provide elucidating information beyond its quadratic nature, so we do not show it. ◁

While the Lorenz 2005 system provides an explicit example of model error, model error could alternatively arise from fixing a parameter to an incorrect value. We show an example with the Lorenz 1963 system.

**Example 6.1.2** (Lorenz 1963: model error)**.** Let $(x(t), y(t), z(t))$ be the state of the system. The true continuous state transition model $R$ is represented by Equation (1.2), i.e.,

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = R\left( \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}; \theta_R \right) = \begin{bmatrix} \sigma_R[y(t) - x(t)] \\ x(t)[\rho_R - z(t)] - y(t) \\ x(t)y(t) - \beta_R z(t) \end{bmatrix}.$$

with parameter $\theta_R = (\sigma_R, \rho_R, \beta_R)$. An imperfect model can simply come from setting a parameter to an incorrect value or getting one of the terms wrong in the above equation, e.g., accidentally multiplying the interactive term $x(t)y(t)$ by a value of $\alpha$:

$$W\left( \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}; \theta_W \right) = \begin{bmatrix} \sigma_W[y(t) - x(t)] \\ x(t)[\rho_W - z(t)] - y(t) \\ \alpha x(t)y(t) - \beta_W z(t) \end{bmatrix}.$$

with $\theta_W = (\sigma_W, \rho_W, \beta_W, \alpha)$. In contrast to the previous example, this model has a simple expression for the model error:

$$D\left( \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}; \theta_D \right) = \begin{bmatrix} \sigma_D[y(t) - x(t)] \\ \rho_D x(t) \\ (1 - \alpha)x(t)y(t) - \beta_D z(t) \end{bmatrix}$$

with parameter $\theta_D = (\sigma_D, \rho_D, \beta_D, \alpha)$, where $\sigma_D = \sigma_R - \sigma_W$, $\rho_D = \rho_R - \rho_W$, and $\beta_D = \beta_R - \beta_W$. There is no model error when the parameters $\sigma_D, \rho_D$ and $\beta_D$ are zero and $\alpha = 1$. Otherwise, the model error is linear in the state when any of the parameters $\sigma_D, \rho_D$, or $\beta_D$ is nonzero and quadratic when $\alpha \neq 1$. $\lhd$

In this particular example, model error is simply corrected by better estimating the parameters using an algorithm such as the one developed in Chapter 5. However, it is not always known a priori when the parameters are fixed to incorrect values. Large and complex codes often accompany state transition models of high-dimensional state-space models, especially those studied in the atmospheric sciences. A seemingly innocent error, such as incorrectly modifying a term, could easily be missed, leading to an imperfect model that produce seemingly realistic results but affect the precision and accuracy of forecasts due to the chaotic nature of the model.

Without knowing the true model $R$, there is often no way to discern whether model error arises from missing the model dynamics, such as Example 6.1.1, or from setting incorrect parameter values, such as Example 6.1.2. Therefore, an investigator may desire to parametrize $D$ in such a way that reflects her hypothesis of the form of the model error. We continue a line of work that examines one of the simplest parametrizations of model error: linear in the state. While simple, the linear correction comes with a complication. The high-dimensional nature of many atmospheric models mean that the parameters of the linear correction is also high-dimensional. For this reason, we formulate a low-rank linear correction by leveraging the spatial correlation often found in atmospheric models. Furthermore, we emphasize that the main goal is to improve forecasts and not to learn the form of the model error. While the linear correction could potentially help locate and diagnose parts of the model that are wrong, it is only an auxiliary benefit to our main goal.

Before reviewing the linear correction and its low-rank approximation, we briefly review even simpler ways to correct model error. In the following section, the true state $X$ is approximated with an approximate state $\tilde{X}$ with state transition model

$$\tilde{x}_{t_n} = \widetilde{m}_{t_n}(x_{t_{n-1}}; \theta_{\widetilde{W}})$$

and parameter $\theta_{\widetilde{W}}$, where $\widetilde{m}_{t_n}$ is some function of the imperfect model $\widehat{m}_{t_n}$. All the improved models $\widetilde{m}_{t_n}$, except the model with the linear correction, modify the discrete state transition model and thus an improved continuous state transition model $\widetilde{W}$ is not proposed; only the model with the linear correction has an improved continuous state transition model $\widetilde{W}$.

## 6.2 Correcting model error: a short review

### 6.2.1 Increasing the variance

If $W$ is believed to be imperfect, more faith should be put into the measurements than the forecasts. Statistically speaking, this is equivalent to increasing the variance of the state transition model to reflect our uncertainty about the model. In the EnKF literature, there are two main ways to increase the uncertainty: random and deterministic variance inflation. The latter is previously discussed in Section 2.3.2. Both inflate the forecasts and are included as options in DART. Though these methods modify the forecasts, the discrete state transition model can be re-parametrized to reflect the nature of these modifications as done with deterministic variance inflation (Section 4.4). We briefly review both.

Random variance inflation is the method of adding mean-zero perturbations $\eta_{t_n}$ to the discrete state transition model:

$$\widetilde{m}_{t_n}(x_{t_{n-1}};\theta_W,U_{t_n}) = \widehat{m}_{t_n}(x_{t_{n-1}};\theta_W) + \eta_{t_n}, \quad \eta_{t_n} \sim \mathcal{N}(0,U_{t_n}). \tag{6.7}$$

with variance $U_{t_n}$. In this model, the approximate state $\tilde{X}_t$ has the same expectation as $\hat{X}_t$ but with a larger variance: $\mathbb{E}(\tilde{X}_{t_n}) = \mathbb{E}(\hat{X}_{t_n})$ and $\mathbb{V}\text{ar}(\tilde{X}_{t_n}) = \mathbb{V}\text{ar}(\hat{X}_{t_n}) + U_{t_n}$. Though it achieves the purpose of increasing the uncertainty in the approximate state $\hat{X}$, this method is not used in practice because it performs poorly with the EnKF. For this reason, deterministic variance inflation was introduced as an alternative method. In fact, even though DART includes random variance inflation as an option, its documentation recommends to "always" use deterministic variance inflation.

Deterministic variance inflation is the method of scaling the state in such a way that maintains its expectation but increases its variance:

$$\widetilde{m}_{t_n}(x_{t_{n-1}};\theta_W,\lambda) = \mu_{t_n} + \Lambda_{t_n}^{1/2}[\widehat{m}_{t_n}(x_{t_{n-1}};\theta_W) - \mu_{t_n}], \tag{6.8}$$

where $\mu_{t_n}$ is the expectation of $\widehat{m}_{t_n}(X_{t_{n-1}};\cdot)$, $\Lambda_{t_n}$ is a diagonal matrix with entries $(\lambda^2_{t_n,1},...,\lambda^2_{t_n,d_x})]$ and $\lambda_i \geq 1$ for all $i = 1,...,d_x$. If $\lambda_{t_n,i} = 1$ for all $i$, then no modification is made to the state transition model arising from the model $W$. Otherwise, the expectation stays the same and the variance is scaled by the factor $\Lambda_{t_n}$. Therefore, this particular method also fulfills the purpose of increasing the uncertainty of the state transition model. Anderson and Anderson (1999) and Hamill et al. (2001) both independently proposed fixed spatially-constant deterministic variance inflation, i.e., fixing $\lambda_{t_n,i} = \lambda_{t_n,j} = \lambda_{t_n}$ to some fixed value $\lambda_{t_n}$ for all $1 \leq i,j \leq d_x$. Adaptive versions, i.e.,

estimated inflation values, were proposed later by Anderson (2007) (spatially-constant) and Anderson (2009) (spatially-varying), along with others (Altaf et al., 2013; Miyoshi, 2011).

### 6.2.2 State-constant correction

While increasing the uncertainty does help improve forecasts, a potentially better solution is to estimate the structural form of the model error. The simplest structural form of model error is to add a state-constant correction to the model, either in the continuous or discrete state transition model. A state-constant correction in the continuous state transition model approximates $D(\cdot;\theta_D)$ with a function that depends on the parameter $\theta_D$ but not on the state $X$. This particular model has been previously implemented as a special case of correcting model error with a linear correction (Danforth and Kalnay, 2008). Alternatively, a state-constant correction is added to the discrete state transition model (Dee and da Silva, 1998; Friedland, 1969, 1978; Ignagni, 1981, 1990; Zupanski and Zupanski, 2006) and has the following form:

$$\widetilde{m}_{t_n}(x_{t_{n-1}};\theta_W,b_{t_n},U_{t_n}) = \widehat{m}_{t_n}(x_{t_{n-1}};\theta_W) + \eta_{t_n},$$

with $\eta_{t_n} \sim \mathcal{N}(b_{t_n},U_{t_n})$. Unlike the methods from the last section, these methods modify the expectation of the approximate state $\hat{X}$, sometimes in addition to increasing the variance, e.g., when $U_{t_n}$ is not the zero matrix. In the literature, these methods are called *model bias*. To distinguish them from statistical bias, we call them *state-constant corrections*, since the correction does not depend on the state $X$. Though reasonable, these methods do not address the very likely scenario that model error is a function of the state.

### 6.2.3 Linear correction: Leith's thought experiment

A linear model is the next simplest form for model error that additionally allows model error to vary with the state. Like the state-constant correction, the linear model is added either to the continuous or discrete state transition models. Although it is easier to estimate a linear model in the discrete state transition model, the linear correction is difficult to interpret. By the assumptions of model error laid out in Section 6.1, the continuous model $W$ is incorrect and should thus be fixed at the source of that error: in the continuous-time model. We believe this line of work is the most promising, so we begin by reviewing the original idea behind this work, first introduced by Leith (1978).

Leith considered approximating the model error $D$ with a linear term and derived the linear correction under a few assumptions, some of which are im-

practical. First, he assumed that the true continuous state transition model $R$ can be evaluated and the true state $X(t)$ can be obtained without measurement error. With these two assumptions, the model error $D$ is evaluated as follows:

$$D[x(t);\theta_D] = R[x(t);\theta_R] - W[x(t);\theta_W].$$

Then, he considered approximating the model error $D$ with a linear model:

$$D[x(t);\theta_D] \approx Lx(t) + b + \epsilon(t)$$

where the parameter $\theta_D$ contains the $d_x \times d_x$ matrix $L$ and $d_x$-vector $b$ and $\epsilon(t)$ is the model error that is not captured by the linear term and assumed to have mean zero. Then, the least-squares solution is:

$$\hat{L} = \mathbb{C}\text{ov}\{D[X(t);\cdot], X(t)\}\,\mathbb{V}\text{ar}[X(t)]^{-1}, \tag{6.9a}$$

$$\hat{b} = \mathbb{E}\{D[X(t);\cdot]\} - \hat{L}\,\mathbb{E}[X(t)]. \tag{6.9b}$$

We call this the *optimal linear correction*. Though not explicitly stated, he made one more assumption to derive the least-squares solution: the state and the model error are temporally independent. These temporal observations provide the replicates necessary to estimate the implied least-squares solution.

Unfortunately, the optimal linear correction is only a thought experiment: the least-squares solution cannot be evaluated because neither does one have the perfect model $R$ to evaluate $D$ nor can the true state $X$ be measured exactly. The first problem is intractable: the investigator does not have the true model $R$ and even if he did, he should use the perfect model $R$ as the state transition model. The second problem, however, is tractable: while the true state $X$ cannot be measured exactly, its moments can be approximated with filtering algorithms. Unfortunately, nonlinear filtering algorithms, such as the EnKF or PF, were not developed at the time. Once nonlinear filtering algorithms had been established, DelSole and Hou (1999) proposed several solutions to overcome the issues that prevented the evaluation of the optimal linear correction; we elaborate on their methods in the next section.

## 6.3 The optimal linear correction: past work and their limitations

Under the same assumptions laid out by Leith, DelSole and Hou (1999) proposed solutions to the problems that prevented the evaluation of the optimal

linear correction: (1) employ a forward Euler approximation to estimate the model error $D$ and (2) approximate the state-space model with the imperfect model $W$ and apply filtering algorithms to estimate the true state.

A forward Euler approximation enables the approximation of the true state $X$ and the approximate state $\hat{X}$ as:

$$x(t_n) = \frac{x_{t_{n+1}} - x_{t_n}}{\Delta t_{n+1}} + O(\Delta t_{n+1}),$$
$$\hat{x}(t_n) = \frac{\hat{x}_{t_{n+1}} - x_{t_n}}{\Delta t_{n+1}} + O(\Delta t_{n+1}),$$

respectively. Subtracting the two expressions provides an approximation of the model error at $t_n$:

$$D[x(t_n); \theta_D] = x(t_n) - \hat{x}(t_n) = \frac{x_{t_{n+1}} - \hat{x}_{t_{n+1}}}{\Delta t_{n+1}} + O(\Delta t_{n+1}). \tag{6.10}$$

While the forward Euler approximation provides a seemingly tractable way to evaluate the model error $D$ at time $t_n$, it is a function of the true state $X$ that cannot be measured. Not only is it a function of the true state $X_{t_{n+1}}$, it contains the approximate state $\hat{X}_{t_{n+1}}$ that is also a function of the true state $\hat{X}_{t_n}$ (see Equation (6.5)). One more approximation is needed to obtain the model error $D$, required in the evaluation of the optimal linear correction (see Equation (6.9)).

The authors additionally proposed approximating the true state $X$ with the best estimate of the imperfect state $\hat{X}$ from a filtering algorithm. Specifically, approximate $X_{t_{n+1}}$ with the forecasted state $\hat{X}^f_{t_{n+1}} = \widehat{m}_{t_{n+1}}(\hat{X}^a_{t_n}; \cdot)$ and $X_{t_{n+1}}$ with the updated state $\hat{X}^a_{t_{n+1}}$, where $\hat{X}^f_{t_{n+1}} \overset{d}{=} \hat{X}_{t_{n+1}} \,|\, y_{0:t_n}$, $\hat{X}^a_{t_n} \overset{d}{=} \hat{X}_{t_n} \,|\, y_{0:t_n}$, and $\hat{X}^a_{t_{n+1}} = \hat{X}_{t_{n+1}} \,|\, y_{0:t_{n+1}}$. Plugging these approximations into Equation (6.10) and assuming that the numerical error from the Euler approximation is negligible, the model error is approximated as

$$D[x(t_n); \theta_D] \approx \frac{\hat{x}^a_{t_{n+1}} - \hat{x}^f_{t_{n+1}}}{\Delta t_{n+1}}.$$

and the least-squares solutions in Equation (6.9) is approximated as

$$\hat{L} \approx \frac{1}{\Delta t_{n+1}} \mathbb{C}\mathrm{ov}(\hat{X}^a_{t_{n+1}} - \hat{X}^f_{t_{n+1}}, \hat{X}^a_{t_{n+1}}) \, \mathbb{V}\mathrm{ar}(\hat{X}^a_{t_{n+1}})^{-1}, \tag{6.11a}$$
$$\hat{b} \approx \frac{1}{\Delta t_{n+1}} \mathbb{E}(\hat{X}^a_{t_{n+1}} - \hat{X}^f_{t_{n+1}}) - \hat{L} \, \mathbb{E}(\hat{X}^a_{t_{n+1}}). \tag{6.11b}$$

Of course, the above moments cannot be evaluated analytically and are replaced by their sample estimates. The authors demonstrated that the linear correction improved forecasts for a relatively low-dimensional state-space ($d_x = 160$).

Though DelSole and Hou provided a tractable solution to approximate the optimal linear correction, there remains a few limitations to their approach. The dynamics that are missed by the imperfect model $W$ are often not linear with the state, as in both examples introduced in Section 6.1. Because the error is not linear, a linear correction does not truly correct the model error and could potentially worsen the model error and thus forecasts. For example, in Example 6.1.2, if $\alpha \neq 1$ and all other parameters, i.e., $\sigma_D, \rho_D$, and $\beta_D$, are zero, a linear correction does not truly correct the model error since the error is an interactive effect between two parts of the state. While there is no general answer, the Lorenz systems introduced in Section 6.1 provide unique settings to test how a linear correction can improve nonlinear model error; we do this in the next chapter.

Although DelSole and Hou demonstrated that a linear correction does indeed improve forecasts for their particular application, it is not clear that their solution improved model error or some other error. While it is common to use the forward Euler method for numerical integration, it is a first-order method with numerical error proportional to the timestep $dt$, i.e., $O(dt)$, and thus the timestep $dt$ is often taken to be small. We give a sense of how small with the Lorenz systems: when using a higher-order method than forward Euler[1], the timestep is taken to be 0.01 for the Lorenz 1963 system and 0.001 for the Lorenz 2005 system. Therefore, an even smaller timestep $dt$ should be used when employing forward Euler to achieve similar accuracy as the higher-order methods. Unfortunately, practitioners do not have this luxury with DelSole and Hou's method: the timestep $dt$ in the approximations cannot be chosen to be as small as necessary to minimize the numerical error because the timestep is dictated by how often measurements are collected, which is the forecast lead time $\Delta t_n$ ($\Delta t_n \gg dt$). Therefore, it is difficult to pinpoint what their linear correction is improving; the linear correction could be correcting the numerical error from the forward Euler approximation, the model error, or a mixture of both errors. Our method eliminates any ambiguities by not making any numerical approximations to estimate the model error.

[1]We use second-order and fourth-order Runge Kutta for the Lorenz 1963 and 2005 systems, respectively. As their names suggest, they are second- and fourth-order methods, i.e., $O(dt^2)$ and $O(dt^4)$, respectively, and are thus more accurate than forward Euler for the same timestep $dt$.

Even if these limitations are overcame, the linear correction is not easily estimated when the state is high-dimensional. Notice that the optimal linear correction in Equation (6.9) requires sample estimates of two matrices: the variance of the true state $\mathbb{V}\text{ar}[X(t)]$ and the covariance of the model error and the true state $\mathbb{C}\text{ov}\{D[X(t);\cdot], X(t)\}$. Furthermore, the evaluation of the linear correction requires the inverse of $\mathbb{V}\text{ar}[X(t)]$, which means its sample estimate must be full rank. Since this matrix has dimension $d_x \times d_x$, where $d_x$ is the dimension of the state, samples on the order of $O(d_x^2)$ are required for the sample estimate of $\mathbb{V}\text{ar}[X(t)]$ to be full-rank and thus invertible.

Danforth et al. (2007) addressed the invertibility of $\mathbb{V}\text{ar}[X(t)]$ in high-dimensional state-spaces. In addition to the approximations introduced by DelSole and Hou, they introduce two additional approximations to obtain full-rank estimates of the two matrices required to calculate the optimal linear correction: tapering and applying singular value decomposition (SVD) to obtain a pseudoinverse of $\mathbb{V}\text{ar}[X(t)]$. We do not present their idea exactly and instead present a cleaner and more interpretable way to effectively accomplish their idea. In particular, let $\hat{\Sigma}$ be an appropriate sample estimate of $\mathbb{V}\text{ar}[X(t)]$ and $UDV^T$ be its SVD, where $D$ is the diagonal matrix with singular values $\gamma_1 \geq ... \geq \gamma_{d_x} \geq 0$. For a chosen $p \in [0, 1]$, let $i_p$ be the index such that $\sum_{d=1}^{i_p} \lambda_d / \sum_{d=1}^{d_x} \lambda_d \geq p$ and $D_p^{-1}$ be the diagonal matrix with $1/\gamma_1, ..., 1/\gamma_{i_p}$ as the first $i_p$ elements and zeroes for the remaining $d_x - i_p$ elements. Then, the linear correction $\hat{L}$ is approximated with

$$\tilde{L} = \mathbb{C}\text{ov}\{D[X(t);\cdot], X(t)\} V D_{95}^{-1} U^T, \tag{6.12}$$

where $\mathbb{C}\text{ov}\{D[X(t);\cdot], X(t)\}$ is replaced with an appropriate sample estimate. This solution is similar to performing least-squares regression with ridge (L2) regularization[2] (Hastie et al., 2009, Section 3.4.1). Although a reasonable solution to the singular matrix problem, their linear correction face the same limitations as the method previously proposed by DelSole and Hou because it employs the same approximations.

We overcome the issues faced by both DelSole and Hou (1999) and Danforth et al. (2007) by parametrizing the linear correction with only a few parameters, far fewer than $O(d_x^2)$. This is possible because the state of the systems studied by atmospheric scientists are often spatially correlated. With few parameters required to be estimated, the algorithm developed in Chapter 5 is easily applied to estimate those parameters. Before introducing our methodology, we

[2]They are not the same because ridge regression takes the SVD of $X(t)$ instead of its variance, which affects the $\mathbb{C}\text{ov}\{D[X(t);\cdot], X(t)\}$ term as well.

introduce a metric often used in the atmospheric sciences to measure forecast skill, which we use to evaluate our improvements over the imperfect model.

## 6.4 Energy score: an evaluation metric for forecast skill

Since the EnKF provides samples from the forecast distribution, it is desirable to evaluate forecasts on both how close the forecast is to the observation (*accuracy*) and the uncertainty in the forecasts (*precision*). The continuous ranked probability score (CRPS), first introduced by Matheson and Winkler (1976)[3], is one such metric for scalar forecasts and is widely used in the atmospheric sciences. Though the forecasts in our demonstrations are multidimensional, we first introduce the CRPS and provide some intuition behind how the metric measures forecast skill before discussing its generalization to multidimensional forecasts.

Let $y$ be the observed value and $P$ be the predictive distribution. The CRPS is defined as

$$\text{CRPS}(P, y) = \int_{-\infty}^{\infty} [F(x) - \mathbb{I}(y \geq x)]^2 dx,$$

where $F$ is the cumulative distribution function (CDF) of the distribution $P$ and $\mathbb{I}$ is the indicator function. When the predictive distribution comes from iid samples $x_i \overset{iid}{\sim} P, i = 1, ..., n$, denoted as $P_{ens}$, the CDF $F$ is either replaced with the empirical CDF, $\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(x_i \leq x)$, or approximated with the CDF of a Gaussian with the first two moments given by the sample mean and variance. CRPS is a negatively oriented metric, meaning that a lower value indicates a better prediction.

Figure 6.1 provides some intuition for the CRPS metric: it illustrates three different predictions $X_i, i = 1, 2, 3$, for the observed value $y = 0$ and their corresponding CRPS statistic. The distribution of all three predictions are Gaussian with different means and variances. Let's examine the first prediction $X_1$ (first row of plots) with a mean that matches the observation exactly and a spread of one standard deviation around the observation, i.e., $X_1 \sim \mathcal{N}(0, 1^2)$. The prediction has a CRPS score is 0.234 and is represented by the shaded region in the top right figure. The second prediction (green in the second row of plots) represents a less accurate prediction with equal precision, i.e., the mean prediction is no longer zero but has the same variance: $X_2 \sim \mathcal{N}(1, 1^2)$.

[3]See Wilks (2011) for a textbook review.

As expected, the CRPS score increases to 0.602, penalizing the prediction for being less accurate. On the other hand, the third prediction (blue in the second row of plots) represents a prediction with the same accuracy as the first but with lower precision, i.e., the mean prediction is zero but has a larger variance of 2.5 times the spread of the original prediction: $X_3 \sim \mathcal{N}(0, 2.5^2)$. The CRPS score (0.584) also increases in this case, penalizing the prediction for being less precise.

Gneiting and Raftery (2007) showed that the CRPS is equivalently written as

$$\text{CRPS}(P, y) = \mathbb{E}_P\,|X - y| - \frac{1}{2}\,\mathbb{E}_P\,|X - X'|,$$

where $X'$ is an independent random variable with the same distribution as $X$. With this insight, Gneiting et al. (2008) proposed the energy score (ES) as a generalization of the CRPS for multidimensional vectors:

$$\text{ES}(P, y) = \mathbb{E}_P\|X - y\| - \frac{1}{2}\,\mathbb{E}_P\|X - X'\|,$$

where $\|\cdot\|$ is the Euclidean norm. When the predictive distribution comes from a sample of size $n$, i.e., $P_{ens}$, the ES reduces to

$$\text{ES}(P_{ens}, y) = \frac{1}{n}\sum_{j=1}^{n}\|x_j - x\| - \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|x_i - x_j\|.$$

We use this particular metric to evaluate the improvement in forecast skill of the improved model $\widetilde{W}$.

| i | $X_i$ | legend | CRPS$(F_{X_i}, y)$ | |
|---|---|---|---|---|
| 1 | $\mathcal{N}(0, 1^2)$ | | 0.234 | |
| 2 | $\mathcal{N}(1, 1^2)$ | | 0.602 | less accurate than $X_1$ |
| 3 | $\mathcal{N}(0, 2.5^2)$ | | 0.584 | less precise than $X_1$ |

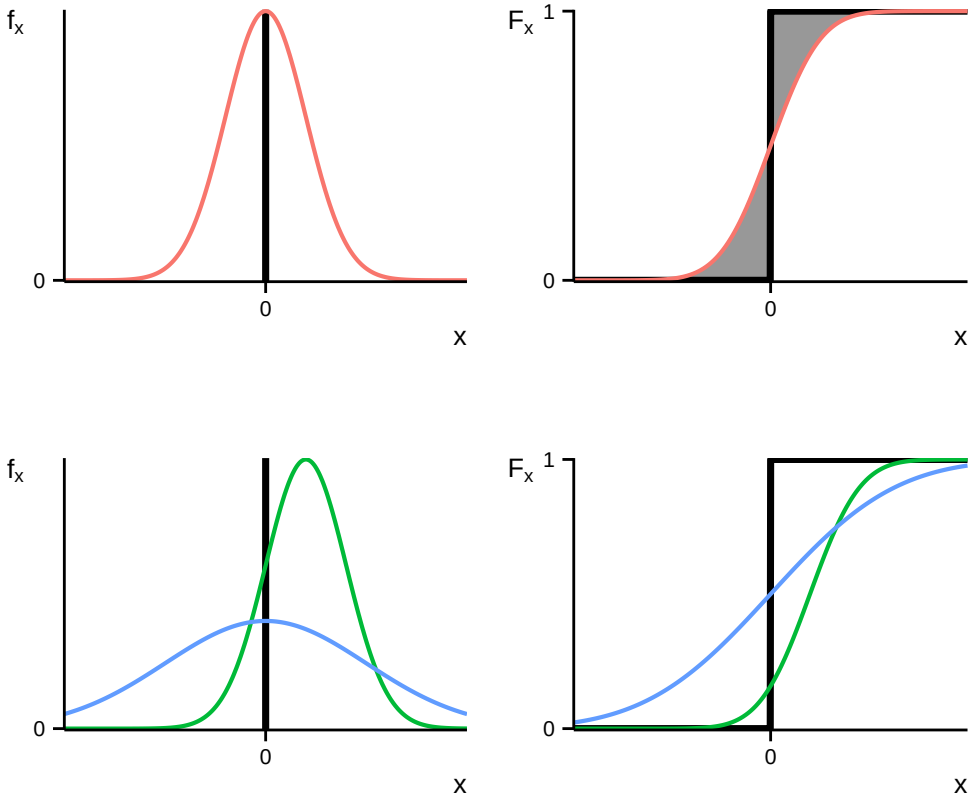



Figure 6.1: *Continuous ranked probability score (CRPS): an illustration.* Three different predictions $X_i, i = 1, 2, 3$, of the observed value $y = 0$ and their corresponding CRPS score. The first prediction $X_1$ has a mean that matches the observation exactly with a spread of one. The other two predictions perturbs the first prediction's accuracy ($X_2$) and precision ($X_3$)—both undesirable and is thus penalized with higher CRPS scores.

# 7 Low-rank linear correction

The goal of this section to demonstrate that a linear approach to correcting model error can improve forecasts even when the error is nonlinear. For real atmospheric systems, it is usually impossible to obtain realizations of the true state $X$ and its model error $D(X)$. Fortunately, both the Lorenz 1963 and 2005 systems have characteristics of real atmospheric systems and thus provide unique settings to obtain realizations of the true state $X$ and its model error $D(X)$—elements that are important to calculate the optimal linear correction (see Equation (6.9)).

In Section 7.1, we examine the situations when the optimal linear correction improves the imperfect model $W$ of the Lorenz 1963 system. We use this demonstration and the realizations of the Lorenz 2005 system to motivate the use of spatial covariance functions to construct low-rank linear corrections. We provide relevant background on spatial covariance functions and demonstrate its value in Section 7.2. In Section 7.3, we construct a low-rank linear correction under the unrealistic assumptions laid out in Leith (1978). This construction, however, provides a parametrization that is easily carried over to state-space models and requires fewer assumptions; we elaborate upon this parametrization in Section 7.4. We demonstrate both low-rank linear corrections on the Lorenz 2005 system. We end the chapter with a discussion of further explorations of the low-rank linear correction on real atmospheric systems.

## 7.1 The optimal linear correction: does it improve nonlinear model error? Demonstrating Leith's method on Lorenz 1963

For the Lorenz 1963 system, the true model $R$ has the default parameter value $\theta_R = (\sigma_R, \rho_R, \beta_R) = (10, 28, 8/3)$. The imperfect model $W$ has parameter $\theta_W = (\sigma_W, \rho_W, \beta_W, \alpha)$ with $\sigma_W = \sigma_R$, $\rho_W = \rho_W$, and $\beta_W = 2$ instead of the default value of $8/3$. When $\alpha = 1$, the model error is linear in the $z$-direction, meaning that the optimal linear correction perfectly captures the model error. Since our goal is to evaluate the viability of a linear correction for nonlinear model error, the model error is contaminated with a nonlinear perturbation by changing the parameter $\alpha$, which controls the interactive effect of the $x$- and $y$-directions. The model error therefore takes the form

$$D\left(\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}; \theta_D\right) = \begin{bmatrix} 0 \\ 0 \\ (1-\alpha)x(t)y(t) - \frac{2}{3}z(t) \end{bmatrix},$$

with $\theta_D = (0, 0, -\frac{2}{3}, \alpha)$. If $\alpha$ is one, the model error in the $z$-direction is exactly $-\frac{2}{3}z(t)$, which is easily captured by the optimal linear correction. When $\alpha$ is not one, the linear part of the model error is contaminated by the nonlinear term $(1-\alpha)x(t)y(t)$. The nonlinearity in the model error is increased with larger values of $\alpha$, overwhelming the linear part of the model error. We experiment with $\alpha$ between 1.1 and 9 to evaluate the performance of optimal linear correction under varying levels of nonlinearity. We only report the results for $\alpha$ of values of 1.1, 1.5, 2, and 3; the unreported values have similar trends as the reported values.

This section begins with a discussion on the generation of the true state $X$ and the model error $D(X)$ for the Lorenz 1963 system that are required in the calculation of the optimal linear correction. Then, we discuss the optimal linear correction used to construct the improved model $\widetilde{W}$ under increasingly nonlinear model error. Finally, we evaluate the forecast skill of the improved model $\widetilde{W}$.

### 7.1.1 Generating realizations of the true state and model error

Before we evaluate the optimal linear correction, we must first obtain the different pieces required to calculate the optimal linear correction (see Equation (6.9)). These are:

- realizations of the true state $X$ and
- the model error $D(X)$ at the realization $X$.

We first describe the generation of realizations of the true state $X$. A randomly chosen initial condition $(x(t_0), y(t_0), z(t_0))$ is numerically integrated with the true model $R$ with $\Delta t_1 = 0.01 \times 10,000$[†], resulting in a state that is on the attractor of the system, $(x(t_1), y(t_1), z(t_1))$. Using this state as the new initial condition, the true states $(x(t_n), y(t_n), z(t_n)), n = \{1, ..., 302400\}$, are sequentially generated by numerically integrating the true model $R$ again with $\Delta t_n = 0.01$. This process provides 302,400 realizations of the true state $X$.

Realizations of the model error $D(X)$ are easily obtained by evaluating the time-derivative functions $R$ and $W$ at each realization of the true state $X$ and taking their difference. For the Lorenz 1963 system, the model error is simply evaluated at the generated true states $(x(t_n), y(t_n), z(t_n))$:

$$\begin{aligned} D\left(\begin{bmatrix} x(t_n) \\ y(t_n) \\ z(t_n) \end{bmatrix}; \theta_D\right) &= R\left(\begin{bmatrix} x(t_n) \\ y(t_n) \\ z(t_n) \end{bmatrix}; \theta_R\right) - W\left(\begin{bmatrix} x(t_n) \\ y(t_n) \\ z(t_n) \end{bmatrix}; \theta_W\right) \\ &= \begin{bmatrix} 0 \\ 0 \\ (1-\alpha)x(t_n)y(t_n) - \frac{2}{3}z(t_n) \end{bmatrix}, \end{aligned}$$

for $n = \{1, ..., 302400\}$. This process provides 302,400 pairs of the true state $(x(t_n), y(t_n), z(t_n))$ and its corresponding model error $D[\cdot; \theta_D]$ that are important in calculating the optimal linear correction.

### 7.1.2 Forecast skill of the optimal linear correction

The optimal linear correction is calculated using the pairs of the true state $X$ and its corresponding model error $D[\cdot; \theta_D]$ using Equation (6.9). We use the

[†]The notation $\Delta t_1 = 0.01 \times 10,000$ indicates that numerical integration is performed with a timestep of 0.01 for 10,000 timesteps.

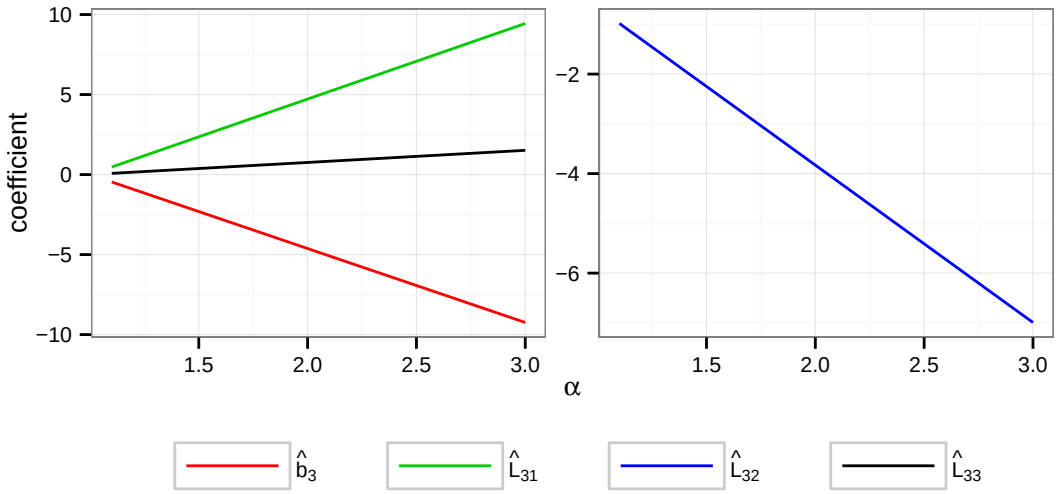



Figure 7.1: *Lorenz 1963: coefficients of the linear correction*

optimal linear correction to construct the improved model $\widetilde{W}$:

$$\widetilde{W}\left(\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}; \theta_{\widetilde{W}}\right) = W\left(\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}; \theta_W\right) + \hat{L}\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} + \hat{b},$$

where $\theta_{\widetilde{W}} = (\theta_W, \hat{L}, \hat{b})$. The parameter $\hat{L}$ is a $3 \times 3$ matrix and $\hat{b}$ is a 3-dimensional vector. Because the model error is only in the $z$-direction, the first two rows of $\hat{L}$ and the first two values in $\hat{b}$ have entries of zero. For varying levels of $\alpha$, Figure 7.1 plots the nonzero values of the optimal linear correction, i.e., the entries in the third row of $\hat{L}$ and the third element in $\hat{b}$. Notice that $\hat{L}_{33}$, the coefficient on the linear part of the model error, is close to the true value $-2/3$ when the model error is close to linear (small values of $\alpha$) and decreases as the model error is increasingly nonlinear (as $\alpha$ increases). Furthermore, the other components, $\hat{L}_{31}$, $\hat{L}_{32}$, and $\hat{b}_3$, are smaller than $\hat{L}_{33}$. For larger $\alpha$, $\hat{L}_{32}$ increases and $\hat{b}_3$ and $\hat{L}_{31}$ decreases. This figure suggests that a linear correction is perhaps unable to properly capture the model error with increasing nonlinearities of model error (higher $\alpha$).

Before evaluating the improved model $\widetilde{W}$, we must generate measurements of the true state $X$. The state-space model is assumed to be a fully observed system (all directions of the state are observed) with unit variances for the measurement noise. Measurements are generated for different forecast lead times $\Delta t_n$ to evaluate the optimal linear correction under increasing levels of nonlinearity. Note that this forecast lead time $\Delta t_n$ is different from the $\Delta t_n$

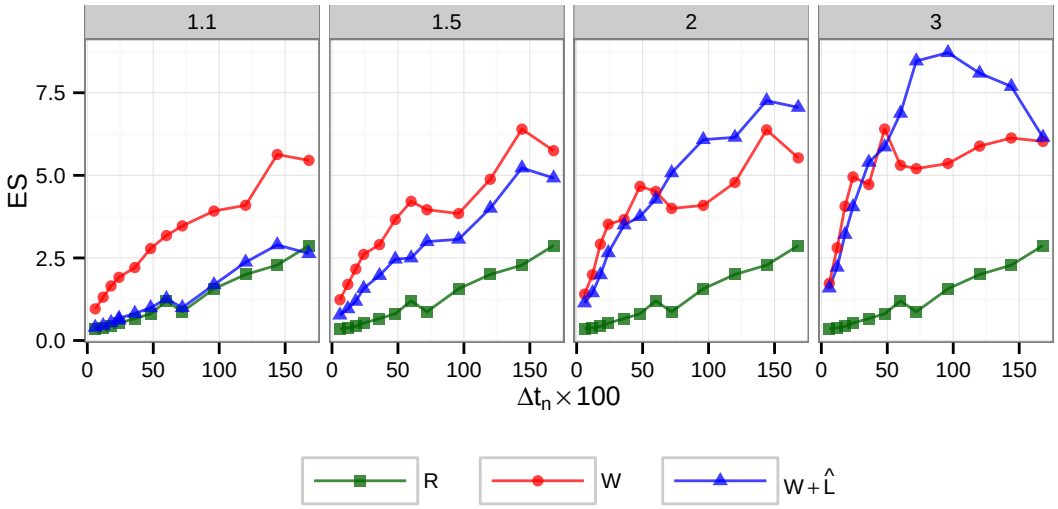



Figure 7.2: *Lorenz 1963: forecast skill of the true model R, the imperfect model W, and the improved model* $\widetilde{W}$. The median ES score of forecasts in the time period where $\sum_n \Delta t_n$ is between $17,520 \times 0.01$ and $26,280 \times 0.01$. Each panel represents a different value of $\alpha$, the parameter that controls the nonlinearity of the model error: the leftmost panel represents model error that is close to linear and the rightmost panel has the most nonlinear model error. The $x$-axis represents the forecast lead times $\Delta t_n$: larger values represent increasing nonlinearities in the state-space model. The true model $R$ has the default parameter value $\theta_R = (\sigma_R, \rho_R, \beta_R) = (10, 28, 8/3)$ and the imperfect model $W$ has the default parameter value $\theta_W = (\sigma_W, \rho_W, \beta_W, \alpha) = (10, 28, 2, \alpha)$.

that is used to generate realizations of the true state $X$ and its model error $D$ in the last section.

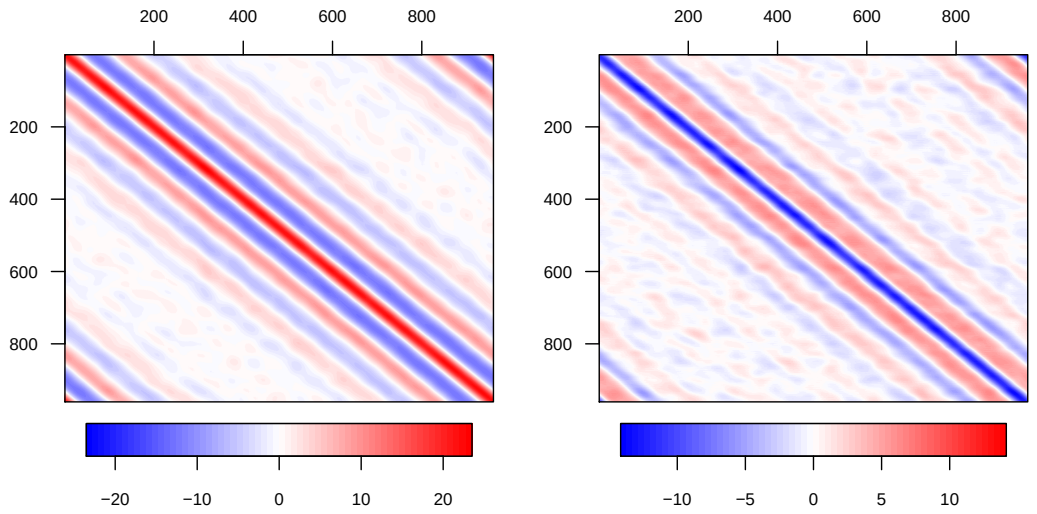
To evaluate the improved model $\widetilde{W}$, we generate forecasts by applying the square-root EnKF with localization on the state-space models with $R$, $W$, and $\widetilde{W}$ as the state transition models and the measurements from the true state $X$. We choose an ensemble size of 48. The localization halfwidth is chosen such that the forecast covariance between any pair of directions is zero. Adaptive inflation is also applied using the method of artificial evolution of parameters with the successful settings as discussed in Section 4.5: the perturbation density is a truncated normal distribution with a standard deviation of 0.6, minimum of 1, and maximum of 5.

We now evaluate the improved model $\widetilde{W}$ under increasing nonlinearities in the model error (larger values of $\alpha$) and state-space model (larger forecast lead times of $\Delta t_n$). Figure 7.2 plots the median ES score of the forecasts in the time period where $\sum_n \Delta t_n$ is between $17,520 \times 0.01$ and $26,280 \times 0.01$. Each panel represents a different value of $\alpha$, the parameter that controls the nonlinearity of the model error: the leftmost panel represents model error that is close to linear and the rightmost panel represents model error that is

significantly nonlinear. The $x$-axis represents the forecast lead time $\Delta t_n$: larger values represent increasing nonlinearities in the state-space model. For all three models, we see that forecast skill degrades under increasing nonlinearities of the state-space model: the ES score increases with increasing forecast lead times. As expected, the imperfect model $W$ never forecasts better than the true model $R$: the ES score is always higher for the imperfect model $W$ than the true model $R$ for all $\alpha$ and forecast lead times $\Delta t_n$.

For small values of $\alpha$ (1.1 and 1.5), the improved model $\widetilde{W}$ provides better forecasts than the imperfect model $W$. For large values of $\alpha$ (2 and 3), the improved model $\widetilde{W}$ generally does better or just as well as the imperfect model $W$ for small forecast lead times ($\Delta t_n < 75$). With $\alpha = 2$ and large forecast lead times $\Delta t_n$, the imperfect model $W$ forecasts better than the improved model $\widetilde{W}$. For $\alpha = 3$, a similar trend is seen but the gap between the two models increases with larger forecast lead times $\Delta t_n$ until $\Delta t_n = 96 \times 0.01$ at which the gap decreases. When the forecast lead time is $\Delta t_n = 168 \times 0.01$, both models perform similarly. In other words, regardless of the degree of nonlinearity, a linear model error does indeed improve nonlinear model error for the Lorenz 1963 system when the forecast lead time is short. With increasing nonlinearities of the state-space model (longer forecast lead times), the linear model error can potentially degrade forecasts.

Since the model error is quadratic, a better model for the model error is to project the state into the quadratic space with the basis functions $x^2(t)$, $y^2(t)$, $z^2(t)$, $x(t)y(t)$, and other interactions. Though these terms are nonlinear, a linear model can still be used to estimate the coefficients as long as these basis functions are chosen ahead of time. Therefore, even though the model error correction is described as "linear", it does not necessarily mean linear in the state because higher-order terms can be added to the correction. However, it is impossible to choose which basis functions to use without knowing the form of the model error a priori. Fortunately, when the true state $X$ is assumed to be a Gaussian process with a particular spatial covariance function, the choice of the basis functions is formalized without explicitly specifying them (Rasmussen and Williams, 2006, Chapter 2). In the next section, we motivate and introduce spatial covariance functions.



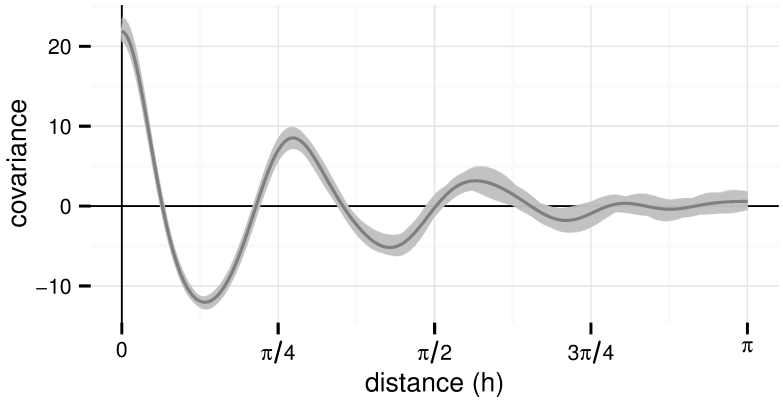

(a) $\mathbb{V}\text{ar}(Z(t))$ (b) $\mathbb{C}\text{ov}(D(Z(t)), Z(t))$

Figure 7.3: *Lorenz 2005: sample estimates of the components in* $\hat{L}$. The axes indicate the indices of the specified matrix with a total of 960 indices. The (co)variances are indicated by the colors.

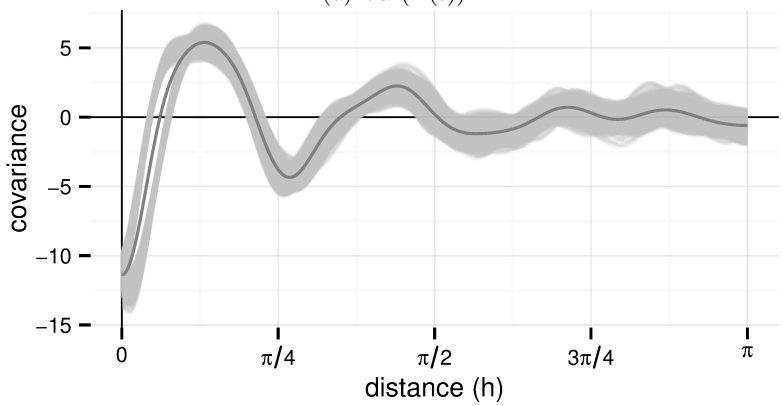
## 7.2 Covariance functions for spatial processes: motivation and background

We use the Lorenz 2005 system from Example 6.1.1 to motivate spatial covariance functions in this section. If the optimal linear correction $\hat{L}$ is applied to this example, the covariance and variance components in $\hat{L}$ both have a dimension of $960 \times 960$, indicating that samples on the order of $O(960^2)$ are needed to estimate these components well. In reality, that sample size is impossible to obtain. If the state is measured every 6 hours, then it would take on the order of hundreds of years to obtain a sufficient amount of samples! Fortunately, atmospheric models describe phenomena over geographic regions and thus the states are spatially correlated. This correlation in space is advantageous: correlation can be made to be a function of distance and thus be described by a few parameters, which is demonstrated at the end of Section 7.2.2.

In a similar manner to the Lorenz 1963 system in the previous chapter, the true states $z(t_n)$ are generated for $n = 1, ..., 302400$ by integrating the true model $R$ with a forecast lead time of $\Delta t_n = 0.001$. The model error $D$ is evaluated at the states $z(t_n)$: $D[z(t_n); \cdot] = R[z(t_n); \cdot] - W[z(t_n); \cdot]$. Assuming temporal independence, the realizations are used to calculate sample estimates of the components of the linear correction $\hat{L}$ from Equation (6.9a): $\mathbb{V}\text{ar}[Z(t_n)]$ and $\mathbb{C}\text{ov}\{D[Z(t_n); \cdot], Z(t_n)\}$; Figure 7.3 depicts the sample estimates. Notice




(a) $\mathbb{V}\text{ar}(Z(t))$




(b) $\mathbb{C}\text{ov}(D(Z(t)), Z(t))$

Figure 7.4: *Lorenz 2005: sample estimates of the components in $\hat{L}$ by distance.* The covariances from Figure 7.3 plotted by distance. There are a total of 1,920 light gray lines in each plot with the mean of those lines represented by the thick gray line. Because the locations of the elements of the state are equally-spaced on a circle, the distance are functions of the arc length of two nearby points $\Delta s$.

how both matrices are highly structured: the covariance is largest in magnitude for nearby states and decreases as the spatial distance increases. To illustrate that the covariance varies by distance, the elements of the matrices are plotted as a function of distance in Figure 7.4. The figure makes clear that the covariance is a smooth function of distance that can accurately be approximated with a few parameters—far less than the number required to properly estimate the matrices in $\hat{L}$. In fact, we show that the empirical mean of the covariance function (thick gray line in the figure) is well approximated by a function with only *four* parameters.

Of course, not all distance functions can be used as a covariance function: the function must be positive semidefinite, a requirement of covariance func-

tions. This is a well-studied problem in spatial statistics. We provide a short review that is relevant to our discussion. Our review is principally based on Gneiting and Guttorp (2010, Chapters 2 and 3) and Rasmussen and Williams (2006, Chapters 2 and 4); we refer the reader to the original sources for more comprehensive reviews.

Suppose there is a spatial stochastic process $Y$ that is a function of the continuous spatial locations $s$: $Y(\cdot) \equiv \{Y(s) : s \in \mathcal{D}\}$ for the spatial domain $\mathcal{D} \in \mathbb{R}^d$ with $d \geq 1$. A model for the spatial stochastic process is

$$Y(s) = \mu(s) + e(s),$$

where $\mu(s) = \mathbb{E}[Y(s)]$ is a deterministic mean function and $e(s)$ is the spatial variation not captured by the mean function. Two examples of the mean function are simply zero, i.e., $\mu(s) = 0$, and linear in a vector of covariates $X(s)$, i.e., $\mu(s) = \beta X(s)$, where $X(s)$ is a vector of covariates observed at $s$ and $\beta$ is the parameter. When there is measurement error from the data collection process, the spatial component $e(s)$ is often divided into two independent components:

$$e(s) = \eta(s) + \epsilon(s),$$

where $\eta(s)$ is the component describing the spatial variation and $\epsilon(s)$ is an additive measurement error with no spatial structure. The measurement error is often assumed to be iid with variance $\tau^2$. The remaining spatial process $\eta(s)$ is assumed to be stationary, which we further elaborate upon in the next section.

The spatial function $\eta$ is often assumed to be distributed as a Gaussian process: $\eta(\cdot) \sim \mathcal{GP}(0, C(\cdot, \cdot; \theta))$. Any finite subset of a Gaussian process is distributed as a multivariate normal distribution. In particular, let $S$ denote a finite subset in $\mathcal{D}$: $S = (s_1, ..., s_n)$ with $s_i \in \mathcal{D}$ for all $i = 1, ..., n$. Furthermore, let $\eta(S)$ be the column vector denoting the function $\eta$ evaluated at spatial locations $S$: $\eta(S) = (\eta(s_1), ..., \eta(s_n))^T$. Then, $\eta(S)$ has a multivariate normal distribution with mean zero and covariance $C(S, S; \theta)$, where $C(S, S; \theta)$ denotes the $n \times n$ covariance matrix evaluated at all pairs of $S$:

$$C(S, S; \theta) = \begin{bmatrix} C(s_1, s_1; \theta) & C(s_1, s_2; \theta) & \cdots & C(s_1, s_n; \theta) \\ C(s_2, s_1; \theta) & C(s_2, s_2; \theta) & \cdots & C(s_2, s_n; \theta) \\ \vdots & \vdots & \ddots & \vdots \\ C(s_n, s_1; \theta) & C(s_n, s_2; \theta) & \cdots & C(s_n, s_n; \theta) \end{bmatrix}.$$

Similarly, the measurement error $\epsilon(S)$ is often assumed to be distributed as a multivariate normal distribution. Since the convolution of two normally distributed random variables is also normally distributed, $Y(S)$ is also distributed

as a multivariate normal distribution. These assumptions are important in the construction of the low-rank linear correction that we introduce in Section 7.3. First, we provide relevant background on the stationarity assumption and covariance functions.

### 7.2.1 Compactly supported, isotropic covariance functions

One popular stationarity assumption is *second-order stationarity*: the covariance is a function of the relative locations $s - s'$:

$$\mathbb{C}\text{ov}[\eta(s), \eta(s')] = C(s - s'; \theta) \text{ for all } s, s' \in \mathcal{D}$$

for some covariance function $C$ with parameter $\theta$. For the covariance matrix of any finite subset of locations to be *symmetric*, the covariance function must satisfy an evenness property: $C(s - s'; \theta) = C(s' - s; \theta)$ for all $s, s' \in \mathcal{D}$. Furthermore, for the covariance matrix of any finite subset of locations to be a symmetric *positive semidefinite* matrix, the covariance function must be a positive semidefinite function:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j C(s_i, s_j) \geq 0 \tag{7.1}$$

for all finite sets $s_1, ..., s_n \in \mathcal{D}$ and constants $a_i, a_j \in \mathbb{R}$. The covariance function is *positive definite* when the greater than or equal sign "$\geq$" is replaced with the greater than "$>$" in the above expression. The function is called *valid* if it satisfies the condition in Equation (7.1). It is generally not easy to check the validity of a covariance function $C$. Since validity is preserved under sums, products, convex mixtures, and convolutions, new covariance functions are often constructed from existing covariance functions that have already been proven to be positive (semi)definite. In this section, we use this fact to construct a symmetric positive definite function to be used in the linear correction of the Lorenz 2005 system. For this reason, we restrict our attention to the two covariance functions used in the construction of this new covariance function.

If the covariance function $C$ is further assumed to be a function of distance between two spatial locations $h = \|s - s'\|$, the random process is assumed to be second-order *isotropic* stationary. These functions are reduced to be dependent only on the distance $h$: $C(s, s'; \theta) = C(h; \theta)$. There are a few features of isotropic covariance functions worth mentioning. The *sill* is defined

as $C(0;\theta) = \lim_{h\to\infty} C(h;\theta)$ and is denoted by $\nu$. When the sill $\nu$ is one, the function $C$ is called a *correlation function*. The *range* is the smallest $h$ where $C(h;\theta)$ equals the sill. Not all covariance functions have a range and thus a related notion called the *effective range* is defined as the the smallest $h$ for which $C(h;\theta)$ is equal to 95% of the sill, which is a function of the *range parameter*, denoted by $c$. For functions with a range, the range parameter is equal to the range. The two covariance functions used in the construction of the new covariance function are both isotropic covariance functions.

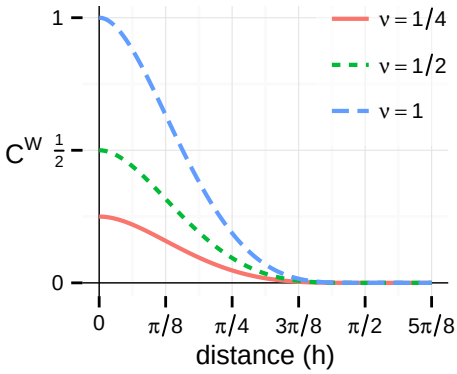
For computational reasons, it is often desirable to choose models with compactly supported covariance functions because these types of functions lead to sparse covariance matrices. Isotropic covariance functions that have a range are compactly supported and the Wendland function is an example of such a function for the domain $\mathcal{D} = \mathbb{R}^d$, $d \in \{1, 2, 3\}$. Let the function $a_+ \equiv \max(a, 0)$. The *Wendland function* $C_q^W$ is defined as:

$$C_q^W(h;\nu,c,\tau) = \begin{cases} \nu\left(1-\frac{h}{c}\right)_+^\tau & \text{if } q=0 \\ \nu\left(1-\frac{h}{c}\right)_+^\tau \left[1+\tau\frac{h}{c}\right] & \text{if } q=1 \\ \nu\left(1-\frac{h}{c}\right)_+^\tau \left[1+\tau\frac{h}{c}+\frac{\tau^2-1}{3}\left(\frac{h}{c}\right)^2\right] & \text{if } q=2 \end{cases} \tag{7.2}$$

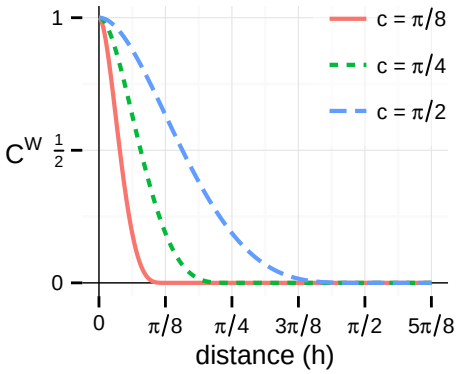
with sill parameter $\nu$, range parameter $c > 0$, differentiability parameter $q \in \{0, 1, 2\}$, and shape parameter $\tau \geq 2(q+1) \equiv \tau_0$. Figure 7.5 plots the Wendland function against distance $h$; each panel varies a parameter while keeping the others fixed. Though not obvious from the figure, the differentiability parameter $q$ controls the smoothness of the spatial process[1]. To illustrate the effect of the differentiability parameter, Figure 7.6 plots random functions sampled from a Gaussian process with the same parameter values used to plot Figure 7.5c. This figure also illustrates the ability of a Gaussian process to model complicated phenomena beyond linearity: an idea that we mentioned earlier.

The Wendland function doesn't quite capture the form of the covariance that we seek: since the Wendland function is positive for all distances, it cannot properly capture the negative covariances present in the Lorenz 2005 system as seen in Figure 7.4. The *hole effect* is the presence of fluctuations (or "wiggles") in the covariance function that results in negative covariances. Chiles and Delfiner (1999) advise against modeling hole effects unless there is a physical explanation. In their setting of modeling one spatial realization, their advice

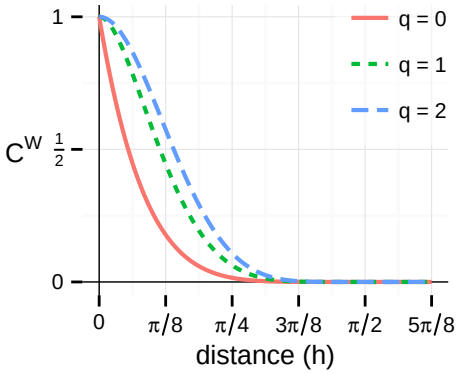
[1]In particular, the covariance function is $2q$-times continuously differentiable and thus the random process $Y(s)$ is $q$-times mean-square differentiable. See both recommended references for more information.



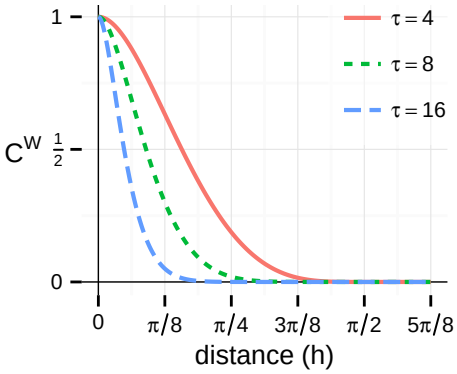

(a) *Varying sill parameter* $\nu$. The values of the parameters $c, q$, and $\tau$ are $\pi/2$, 1, and 4, respectively.




(b) *Varying range parameter* $c$. The values of the parameters $\nu, q$, and $\tau$ are 1, 1, and 4, respectively.




(c) *Varying differentiability parameter* $q$. The values of the parameters $\nu, c$, and $\tau$ are 1, $\pi/2$, and 6, respectively.




(d) *Varying shape parameter* $\tau$. The values of the parameters $\nu, c$, and $q$ are 1, $\pi/2$, and 1, respectively.

Figure 7.5: *Wendland function.* The Wendland function plotted against distance $h$. Each panel varies the value of a parameter while keeping the others fixed. The Wendland function is defined in Equation (7.2).

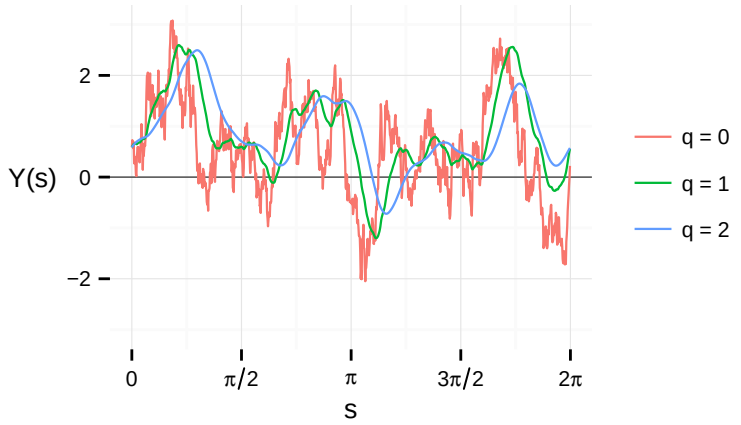



Figure 7.6: *Effect of the differentiability parameter.* Random functions drawn from a Gaussian process with mean zero and covariance given by the Wendland function (Equation (7.2)): $Y(\cdot) \sim \mathcal{GP}(0, C_q^W(\cdot, \cdot; 1, \pi/2, 6))$.

should be heeded because statistically insignificant fluctuations may appear during exploratory data analysis. However, there is good reason to believe that the states of the Lorenz 2005 system have a strong hole effect for two reasons: (1) there are prominent spatial waves in the state and (2) Figure 7.4 is constructed with many spatial realizations (302,400 of them).

The *exponentially damped cosine function* is an example of a valid positive definite function that allows for negative covariances. It has the domain $\mathcal{D} = \mathbb{R}^d$ with integer-values $d \geq 1$. The function is defined as:

$$\widetilde{C}^C(h; \nu, c, \tau) = \nu \exp\left(-\tau \frac{h}{c}\right) \cos\left(\frac{h}{c}\right)$$

with sill parameter $\nu$, range parameter $c > 0$, and shape parameter $\tau \geq 1/\tan(\frac{\pi}{2d}) \equiv \tau_1$. The cosine component is the part of the function that allows for negative covariances. The exponential component has the effect of "damping" the cosine component so that $\widetilde{C}^C(h; \cdot)$ has a limit as $h \to \infty$. The restriction on $\tau$ is important to maintain the positive definiteness of the function (Zastavnyi, 2000). This particular covariance function does not have a range and thus only has an effective range.

Not only are we interested in covariance functions that allow for negative covariances, we are interested in having these functions be compactly supported for computational reasons. One such covariance function is easily constructed by simply multiplying the exponentially damped cosine function by the compactly supported Wendland function. Doing this, however, results in an overparametrized covariance function: it has two sill parameters, two

range parameters, and two shape parameters. Since the Wendland function contains all features needed of a compactly supported covariance function except for a hole effect, we only desire the exponentially damped cosine function for its cosine component. Unfortunately, the cosine component in the current parametrization of the exponentially damped cosine function is not interpretable. We therefore reparametrize the function to have a parameter that controls the number of periods in the cosine function. This reparametrized function is more interpretable and used to construct our compactly supported covariance function with a hole effect.

Let $c = c'/(2\pi np)$, where $np$ controls the number of periods within the distance $c'$. Since $c$ is allowed to be any value strictly greater than zero, the ratio $c'/(np)$ must also be strictly greater than zero. The reason for choosing two parameters to represent the period parameter is explained after we reparametrize the exponentially damped cosine function. The reparametrized function is as follows:

$$C_n^C(h;\nu,c',\tau,p) = \nu \exp\left(-\tau \times 2\pi np \times \frac{h}{c'}\right) \cos\left(2\pi np \times \frac{h}{c'}\right)$$

with sill parameter $\nu$, range parameter $c' > 0$, shape parameter $\tau \geq 1/\tan(\frac{\pi}{2d}) \equiv \tau_1$, maximum period parameter $n$, and period fraction parameter $p$. Though $n$ and $p$ both take any positive, finite value, it is advantageous to restrict their ranges to avoid aliasing and thus identifiability issues when estimating these parameters. *Aliasing* occurs when two signals are indistinguishable from each other and is particularly an issue for sinusoidal functions, such as the cosine function. Specifically, input values that are separated by $2\pi$ are evaluated to the same value: for all $x \in \mathbb{R}$ and integer values $i$, $\cos(x) = \cos(x + 2\pi i)$ and, because cosine is an even function, $\cos(x) = (2\pi i - x)$. For the reparametrized exponentially damped cosine function, aliasing is avoided by choosing an integer $n \geq 0$ and restricting $p \in [0, 1]$ and thus $np$ is interpreted as the maximum number of periods within a distance of $c'$.

With this new reparametrization, we can now construct a compactly supported covariance function that allows for negative covariances by simply multiplying it by the Wendland function:

$$C_{q,n}^{WC}(h;\theta) = C_q^W(h;\nu,c,\tau)C_n^C(h;1,c,\tau_1,p). \tag{7.3}$$

To avoid overparametrization and identifiability issues, we fix a few parameters in the reparametrized exponentially damped cosine function portion: there is no need for two sill parameters, so the sill parameter $\nu$ is set to one; the shape parameter $\tau$ is set be the minimum value $\tau_1$ that makes it a valid positive

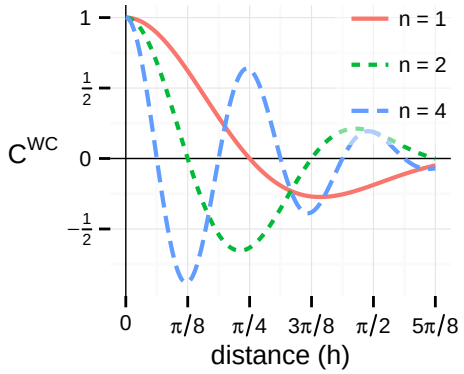
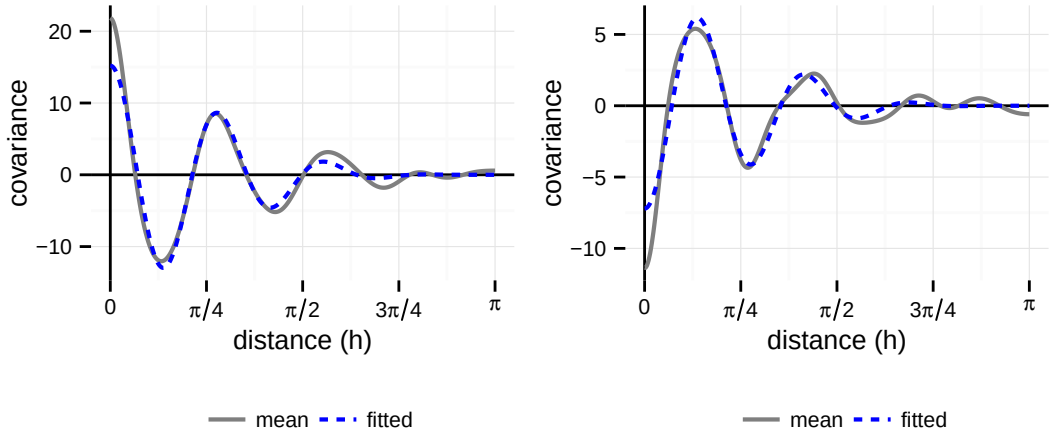



Figure 7.7: *Wendland exponentially damped cosine function.* Since Figure 7.5 already provides intuition for most of the function's parameters, this figure only depicts the variation in the maximum period parameter $n$. The values of the parameters $\nu, c, q, \tau$, $p$, and $d$ are 1, $\pi/2$, 1, 4, 1, and 1, respectively. The function is defined in Equation (7.3).

definite function; and since the Wendland function naturally has a maximum distance, i.e., the range, the range parameter $c'$ is set to equal the range parameter of the Wendland function. We call this function the *Wendland exponentially damped cosine function*. The function has a differentiability parameter $q \in \{0, 1, 2\}$, maximum period parameter $n \in \{1, 2, ...\}$, and parameter vector $\theta \equiv (\nu, c, \tau, p)$ containing the sill parameter $\nu$, range parameter $c > 0$, shape parameter $\tau \geq 2(q + 1)$, and period fraction parameter $p \in [0, 1]$. Because of the restriction imposed by the Wendland function, the function is valid for the domain $\mathcal{D} = \mathbb{R}^d$ with $d \in \{1, 2, 3\}$. When $p = 0$, the $C_n^C$ component is exactly one and thus the Wendland exponentially damped cosine function is equal to the Wendland function. Since Figure 7.5 already provides intuition for many of the parameters, Figure 7.7 shows how the function varies with different period parameter values.

### 7.2.2 Covariance functions on spherical domains: validity and computational gains in efficiency

Not all covariance functions are valid for spherical domains, like the domain of the Lorenz 2005 system. Gneiting (2013) reviews necessary and sufficient conditions for positive definite functions on spherical domains. Before discussing how to make our newly constructed covariance function valid for spherical domains, we describe notation necessary for spherical domains. Let $\mathbb{S}^d = \{s \in \mathbb{R}^{d+1} : \|s\| = 1\}$ denote the unit sphere in $\mathbb{R}^{d+1}$ for integer-valued $d \geq 1$. Define the distance function $\theta(s, s') = \arccos(\langle s, s' \rangle)$ to be the great circle, spherical, or geodesic distance on $\mathbb{S}^d$ where $\langle s, s' \rangle$ denotes the inner prod-




(a) $\mathbb{V}\text{ar}(Z(t))$. Fitted line has parameter value $(\nu, c, \tau, p) = (15.20, \pi, 4, 0.18)$.

(b) $\mathbb{C}\text{ov}(D(Z(t)), Z(t))$. Fitted line has parameter value $(\nu, c, \tau, p) = (-7.21, \pi, 4, 0.18)$.

Figure 7.8: *Lorenz 2005: fitting the Wendland exponentially damped cosine function to the components of* $\hat{L}$. The Wendland exponentially damped cosine function $C_{1,20}^{WC}$ is fit to the empirical mean covariance functions of Figure 7.4.

uct in $\mathbb{R}^{d+1}$. The covariance function now depends on $h = \theta(s, s') \in [0, \pi]$. For example, the domain of the Lorenz 2005 system is $\mathbb{S}^1$ and one distance metric is the shortest arc length between two spatial locations. To make the Wendland exponentially damped cosine function of Equation (7.3) valid for spherical domains, one can simply restrict the range parameter $c$ to be in $(0, \pi]$ by Theorem 3 in Gneiting (2013). Similar to how the Wendland exponentially damped cosine function on $\mathcal{D} \in \mathbb{R}^d$ is valid only for $d \in \{1, 2, 3\}$, the Wendland exponentially damped cosine function on $\mathcal{D} \in \mathbb{S}^d$ is also valid only for $d \in \{1, 2, 3\}$.

Now that the new constructed covariance function is guaranteed to be valid on spherical domains, we fit the Wendland exponentially damped cosine function to the empirical mean covariance functions represented by thick gray lines in Figure 7.4; Figure 7.8 plots the results. Notice how well the fitted functions match the empirical mean functions. Though the fits are not perfect, the fitted function has only four parameters, which is sufficiently estimated with far fewer samples than the number required to estimate the full variance or covariance matrix. We exploit this spatial structure in the construction of a low-rank linear correction in Section 7.3. Before we do that, we first introduce notation useful for the construction of the model and discuss computational gains that can be exploited for random processes on spherical domains.

With Gaussian processes, the largest computational expense is the inver-

sion of the covariance matrix at the spatial locations $S = (s_1, ..., s_n) \in \mathcal{D}$: this is an $O(n^3)$ operation. When a random process is assumed to be second-order isotropic stationary, the variance of the equispaced observations on a spherical domain is a *circulant matrix*. For these types of matrices, fast Fourier transforms (FFTs) can be used to quickly invert the matrix in $O(n \log n)$ time. Rue and Held (2005, Section 2.6.1) is a great resource on computational gains to be exploited with circulant matrices. To illustrate the circulant nature of the matrices, we use the Lorenz 2005 system as an example. We first generalize the notation used to describe model error, which is useful in the construction of a low-rank linear correction. In addition to indexing the state by time $t$ as previously done, i.e., $X(t)$, let the state also be indexed by space: $X(t, s)$ for a spatial location $s \in \mathbb{S}^d$. Furthermore, we similarly index the model error $D$ by space and further introduce a shorthand: $D(t, s) \equiv D[X(t, s); \cdot]$.

Recall that the state of the Lorenz 2005 system is denoted by $Z$ instead of by $X$. Let $d_z$ denote the resolution of the system, where $d_z = 960$ in our demonstrations. Suppose the states are observed at $s_i = \frac{i-1}{d_z} \times 2\pi$ for each $i \in \{1, ..., d_z\}$. Then, the distance between two nearby locations $s_i$ and $s_j$ for $|i - j| = 1$ is $\Delta s \equiv \frac{2\pi}{d_z}$. If the true state $Z(t, \cdot)$ for a particular time $t$ is assumed to be a second-order isotropic stationary random process with covariance function $C(\cdot; \theta_z)$, then the variance of $Z(t, S)$ is a $d_z \times d_z$ circulant matrix with the following form:

$$\mathbb{V}\text{ar}[Z(t, S)] = \begin{bmatrix} C(0; \theta_z) & C(\Delta s; \theta_z) & C(2\Delta s; \theta_z) & \cdots & C(2\Delta s; \theta_z) & C(\Delta s; \theta_z) \\ C(\Delta s; \theta_z) & C(0; \theta_z) & C(\Delta s; \theta_z) & \cdots & C(3\Delta s; \theta_z) & C(2\Delta s; \theta_z) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ C(\Delta s; \theta_z) & C(2\Delta s; \theta_z) & C(3\Delta s; \theta_z) & \cdots & C(\Delta s; \theta_z) & C(0; \theta_z) \end{bmatrix}.$$

Similarly, if the covariance between the model error $D(t, \cdot)$ and the state $Z(t, \cdot)$ for a particular time $t$ is assumed to be second-order isotropic with covariance function $C(\cdot; \theta_{dz})$, then the covariance is also a $d_z \times d_z$ circulant matrix with the following form:

$$\mathbb{C}\text{ov}[D(t, S), Z(t, S)] = \begin{bmatrix} C(0; \theta_{dz}) & C(\Delta s; \theta_{dz}) & C(2\Delta s; \theta_{dz}) & \cdots & C(2\Delta s; \theta_{dz}) & C(\Delta s; \theta_{dz}) \\ C(\Delta s; \theta_{dz}) & C(0; \theta_{dz}) & C(\Delta s; \theta_{dz}) & \cdots & C(3\Delta s; \theta_{dz}) & C(2\Delta s; \theta_{dz}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ C(\Delta s; \theta_{dz}) & C(2\Delta s; \theta_{dz}) & C(3\Delta s; \theta_{dz}) & \cdots & C(\Delta s; \theta_{dz}) & C(0; \theta_{dz}) \end{bmatrix}.$$

One more computationally advantageous feature is that a circulant matrix is uniquely defined by its first row (called the *base* of the matrix). Therefore,

only $d_z$-vectors are computed and stored instead of $d_z \times d_z$ matrices. If the random process is modeled with a compactly supported covariance function, the base vector is sparse, providing another computational gain.

## 7.3 Low-rank linear correction under unrealistic assumptions

Under the same assumptions made in Leith (1978), we construct a low-rank linear model for the model error $D$. Recall from our discussion in Section 6.3 that these assumptions are quite restrictive: they require the ability to evaluate the true state $X$ and model error $D(X)$. Therefore, the model that we construct in this section is impractical for state-space models, but it does provide some intuition for building a similar model that requires less assumptions than Leith (1978). In fact, the model easily carries over to state-space models with few modifications.

Suppose the model error $D$ at time $t$ is a random process that is a function of the continuous spatial locations $s \in \mathcal{D}$. Further suppose it is a linear function of another spatial random process $X$:

$$D(t, s) = L(s, \cdot)X(t, \cdot) + e(t, s), \tag{7.4}$$

where $L(s, \cdot)$ is a function that maps the state $X$ to the model error $D$ and $e(t, s)$ is the model error not captured by the mean model $L(s, \cdot)X(t, \cdot)$. If the joint distribution of the model error $D$ and state $X$ is assumed to be a Gaussian process, the linear model is easily derived as the conditional distribution of $D$ given $X$. We detail the assumptions that lead to this derivation.

Suppose that the model error $D$ and the true state $X$ are both second-order stationary with covariances

$$\begin{aligned}\mathbb{C}\text{ov}[D(t, s), D(t, s')] &= \sigma_d^2 C(s, s'; \theta_d),\\ \mathbb{C}\text{ov}[X(t, s), X(t, s')] &= \sigma_x^2 C(s, s'; \theta_x).\end{aligned}$$

The function $C$ is a correlation function[2] with parameters $\theta_d$ and $\theta_x$ for the model error and state processes, respectively. Similarly, the sill parameters are $\sigma_d^2$ and $\sigma_x^2$, respectively. There is no need to assume isotropy, so we do not make that assumption here. The mean of both processes is assumed to

[2]Recall from the previous section that correlation functions are covariance functions with a sill parameter of one.

be zero. Further assume that their relationship is also second-order stationary with covariance

$$\mathbb{C}\text{ov}[D(t,s), X(t,s')] = \rho\sigma_x\sigma_d C(s,s';\theta_{dx})$$

with correlation function $C$ with parameter $\theta_{dx}$. The sill parameter is a function of the square root of the sill parameters from both processes ($\sigma_x^2$ and $\sigma_d^2$) and a correlation parameter $\rho \in [-1,1]$. Lastly, assume that the two processes are jointly distributed as a Gaussian process:

$$\begin{bmatrix} D(t,\cdot) \\ X(t,\cdot) \end{bmatrix} \sim \mathcal{GP}\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_d^2 C(\cdot,\cdot;\theta_d) & \rho\sigma_d\sigma_x C(\cdot,\cdot;\theta_{dx}) \\ \rho\sigma_d\sigma_x C(\cdot,\cdot;\theta_{dx}) & \sigma_x^2 C(\cdot,\cdot;\theta_x) \end{bmatrix} \right).$$

Let $\theta$ contain all parameters of the model: $\theta = (\sigma_x^2, \sigma_d^2, \rho, \theta_x, \theta_d, \theta_{dx})$. We first derive the components of Equation (7.4) and then its likelihood.

The model is easily derived by examining the conditional distribution of the model error $D$. Suppose the model error process $D$ is observed at finite spatial locations $S_d \in \mathcal{D}$ and the state process $X$ is observed at finite spatial locations $S_x \in \mathcal{D}$. Let the number of components in $S_d$ and $S_x$ be denoted by $d_d$ and $d_x$, respectively. Then, the joint distribution of the observed processes is

$$\begin{bmatrix} D(t,S_d) \\ X(t,S_x) \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_d^2 C(S_d,S_d;\theta_d) & \rho\sigma_d\sigma_x C(S_d,S_x;\theta_{dx}) \\ \rho\sigma_d\sigma_x C(S_x,S_d;\theta_{dx}) & \sigma_x^2 C(S_x,S_x;\theta_x) \end{bmatrix} \right).$$

The conditional distribution of the model error $D$ is also distributed as a multivariate normal distribution with moments

$$\begin{aligned} &\mathbb{E}[D(t,S_d) \,|\, X(t,S_x)] \\ &\quad = \rho\frac{\sigma_d}{\sigma_x} C(S_d,S_x;\theta_{dx}) C(S_x,S_x;\theta_x)^{-1} X(t,S_x), \\ &\mathbb{V}\text{ar}[D(t,S_d) \,|\, X(t,S_x)] \\ &\quad = \sigma_d^2 C(S_d,S_d;\theta_d) - \rho^2\sigma_d^2 C(S_d,S_x;\theta_{dx}) C(S_x,S_x;\theta_x)^{-1} C(S_x,S_d;\theta_{dx}). \end{aligned} \tag{7.5}$$

These moments of the conditional distribution provide the components in Equation (7.4).

The first moment provides the form of the linear mapping $L$:

$$L(S_d,S_x;\theta_L) = \rho\frac{\sigma_d}{\sigma_x} C(S_d,S_x;\theta_{dx}) C(S_x,S_x;\theta_x)^{-1} \tag{7.6}$$

with parameter $\theta_L = (\rho, \sigma_d, \sigma_x, \theta_x, \theta_{dx})$. The linear mapping is a $d_d \times d_x$ matrix that is a function of the covariance between the model error $D(t,S_d)$ and

the state $X(t, S_x)$ ($C(S_d, S_x; \theta_{dx})$) standardized by the variance of the state ($C(S_x, S_x; \theta_x)$). Notice that the form is very similar to the linear mapping derived by Leith (see Equation (6.9a)): it is a function of the variance of the state $X$ ($\mathbb{V}\text{ar}(X)$) and its covariance with the model error $D$ ($\mathbb{C}\text{ov}[D(X), X]$). However, because of the considerable structure imposed on the random processes, the linear mapping is parametrized by a few parameters and hence we call it a *low-rank linear correction*. This particular form of the linear mapping is important later when we construct the low-rank linear correction for state-space models.

The second moment provides the distribution for the error $e(t, S)$ that is not captured by the state $X$:

$$e(t, S_d) \sim \mathcal{N}(0, C(S_d, S_d; \theta_{d|x})),$$

with variance equal to the expression in Equation (7.5) with parameter $\theta_{d|x} = (\rho, \sigma_d^2, \theta_d, \theta_x, \theta_{dx})$:

$$\overbrace{C(S_d, S_d; \theta_{d|x})}^{\text{variance unexplained}} = \underbrace{\sigma_d^2 C(S_d, S_d; \theta_d)}_{\text{total variance}} - \underbrace{\rho^2 \sigma_d^2 C(S_d, S_x; \theta_{dx}) C(S_x, S_x; \theta_x)^{-1} C(S_x, S_d; \theta_{dx})}_{\text{variance explained}}.$$

This is exactly the variance of the model error that is not captured by the mean model $L(s, \cdot)X(t, \cdot)$. The first term is the total variance of the model error $D(t, S_d)$ and the second term is the variance of the model error $D(t, S_d)$ explained by the state $X(t, S_x)$. The difference between the two terms is the remaining variance in the model error unexplained by the state $X(t, S_x)$.

The model we have presented is often used to model multivariate spatial processes in the spatial statistics literature (Gelfand and Banerjee, 2010). The underlying assumption we made about the relationship between the two spatial processes $X(t, \cdot)$ and $D(t, \cdot)$ that we have presented here is called *separable* and is the most simplifying assumption for these types of models. The model is called separable because it separates the local covariance from the correlation based on distance. Specifically, for all $s, s' \in \mathcal{D}$, the local covariance is

$$\begin{bmatrix} D(t, s) \\ X(t, s) \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_d^2 & \rho\sigma_d\sigma_x \\ \rho\sigma_d\sigma_x & \sigma_x^2 \end{bmatrix} \right)$$

and the correlation based on distance is

$$\begin{bmatrix} D(t, s) \\ X(t, s') \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_d^2 & \rho\sigma_d\sigma_x C(s, s'; \theta_{dx}) \\ \rho\sigma_d\sigma_x C(s', s; \theta_{dx}) & \sigma_x^2 \end{bmatrix} \right).$$

This is a strict assumption that assumes that the relationship between two spatial processes is the same for all locations in the domain. Furthermore, because of the spatial structure imposed on the variances and covariance of the random processes, this model is equivalent to a linear regression with regularization similar to the SVD method proposed by Danforth et al. (2007) (see Equation (6.12)). Unlike SVD, however, this particular regularization respects the spatial structure of the random processes.

To complete the model, the likelihood is derived for Equation (7.4). Let the mean of $D(t, S_d)$ be denoted as $\overline{D(t, S_d)} = L(S_d, S_x; \theta)X(t, S_x)$. Then, the log likelihood at time $t$ is

$$\log p[D(t, S_d), X(t, S_x) \,|\, \theta] = \log p[D(t, S_d) \,|\, X(t, S_x), \theta] + \log p[X(t, S_x) \,|\, \theta],$$

with

$$\begin{aligned}
&\log p[D(t, S_d) \,|\, X(t, S_x), \theta] \\
&\quad = -\frac{d_d}{2} \log(2\pi) - \frac{1}{2} \log |C(S_d, S_d; \theta_{d|x})| \\
&\qquad - \frac{1}{2} \left[ D(t, S_d) - \overline{D(t, S_d)} \right]^T C(S_d, S_d; \theta_{d|x})^{-1} \left[ D(t, S_d) - \overline{D(t, S_d)} \right]
\end{aligned}$$

and

$$\begin{aligned}
&\log p[X(t, S_x) \,|\, \theta] \\
&\quad = -\frac{d_x}{2} \log(2\pi) - \frac{1}{2} \log |C(S_x, S_x; \theta_x)| - \frac{1}{2} X(t, S_x)^T C(S_x, S_x; \theta_x)^{-1} X(t, S_x).
\end{aligned}$$

Since the model error $D$ and the state $X$ is assumed to be temporally independent, the log likelihood for all observations evaluated at times $t = 1, ..., T$ is

$$\log p(\text{data} \,|\, \theta) = \sum_{t=1}^{T} \log p[D(t, S_d) \,|\, X(t, S_x), \theta] + \sum_{t=1}^{T} \log p[X(t, S_x) \,|\, \theta]. \quad (7.7)$$

With the log likelihood, any applicable optimization method can be used to estimate the parameters of the model. We end this section by demonstrating the effectiveness in the low-rank linear correction in improving the imperfect model $W$ of the Lorenz 2005 system from Example 6.1.1.

### 7.3.1 Demonstration: Lorenz 2005

So far, we have introduced three different linear corrections under the assumptions made in Leith (1978):

- the least-squares solution $\hat{L}$ from Equation (6.9a);
- the regularized linear correction $\tilde{L}$ from Equation (6.12);
- the low-rank linear correction $L(\cdot, \cdot; \hat{\theta}_L)$, where $\hat{\theta}_L$ is the subset of the vector $\theta$ that maximizes the likelihood in Equation (7.7).

In this section, we plug these linear corrections into the improved model

$$\widetilde{W}(z(t); \theta_{\widetilde{W}}) = W(z(t); \theta_W) + Lz(t)$$

and evaluate their forecast skill on the Lorenz 2005 system presented in Example 6.1.1. We emphasize that the linear corrections are *not* estimated *while* filtering the state-space model, as done in Section 5.5. For that to happen, we need to make modifications to the linear correction presented in Equation (7.4), which we present in the next section.

Under the assumptions made by Leith, realizations of the true state $Z$ and the model error $D$ must be generated to estimate the above linear corrections. The process to generate the realizations is described at the beginning of Section 7.2: this generated 302,400 realizations from the Lorenz 2005 system. This is a sufficient number of samples to estimate all three linear corrections. In particular, this was enough to obtain a full-rank sample estimate of $\mathbb{V}\mathrm{ar}[Z]$ to calculate $\hat{L}$. Furthermore, the realizations are sufficient to directly calculate the regularized linear correction $\tilde{L}$. Before estimating the optimal $\hat{\theta}_L$ for the low-rank linear correction, we must make a few assumptions regarding the correlation structure of the true state $Z$ and the model error $D$ and the correlation between them. In the section labeled "Estimating parameters of the low-rank linear correction", we discuss these assumptions and describe the estimation procedure to find $\hat{\theta}_L$ to be plugged into the low-rank linear correction.

All three of these linear corrections are then plugged into $\widetilde{W}$ and, with measurements generated from the true model $R$, we filter the state-space model with the improved model $\widetilde{W}$ as its state transition. Similarly, we also filter with the true model $R$ and the imperfect model $W$ as the state transition models and compare the forecast skills of all models at the end of the section.

### Estimating parameters of the low-rank linear correction

In addition to the assumptions of the model used to construct the low-rank linear correction (Equation (7.4)), we further assume that the model error $D$, the true state $Z$, and their correlation are isotropic. The exploratory data analysis in Figure 7.4 indicated strong hole effects in both the covariance of the true state $Z$ and the covariance between the true state $Z$ and model error $D$,

so we choose the form of their covariances to be the Wendland exponentially damped cosine function:

$$\mathbb{C}\text{ov}[Z(t,s), Z(t,s')] = \sigma_z^2 C^{WC}_{q_z,n}(s,s';\theta_z),$$
$$\mathbb{C}\text{ov}[D(t,s), Z(t,s')] = \rho\sigma_z\sigma_d C^{WC}_{q_{dz},n}(s,s';\theta_{dz}),$$

where $\theta_z = (c_z, \tau_z, p_z)$, and $\theta_{dz} = (c_{dz}, \tau_{dz}, p_{dz})$. The covariance of the model error $D$ is not observed to have a hole effect and is simply chosen to be the Wendland function:

$$\mathbb{C}\text{ov}[D(t,s), D(t,s')] = \sigma_d^2 C^{W}_{q_d,n}(s,s';\theta_d),$$

where $\theta_d = (c_d, \tau_d)$. The parameters $q.$, $n$, $c.$, $\tau.$, and $p.$ are the differentiability, maximum wavelength, range, shape, and period fraction parameters, respectively. The model error is evaluated at the locations of the state: $S \equiv S_d = S_z = (0, \frac{1}{d_z} \times 2\pi, \frac{2}{d_z} \times 2\pi, \frac{3}{d_z} \times 2\pi, ..., \frac{d_z-1}{d_z} \times 2\pi)$.

Since a few of the parameters are integer-valued, such as $q.$ and $n$, we cannot directly employ off-the-shelf optimization algorithms to find the parameter values that maximize the likelihood from Equation (7.7). Therefore, we enumerate over a reasonable range of values for the integer-valued parameters, employ Nelder-Mead constrained optimization to estimate the remaining continuous-valued parameters, and report on two sets of parameter values, $\hat{\theta}_L^{EDA}$ and $\hat{\theta}_L^{CV}$.

Our exploratory data analysis indicated that the state $Z$ is a smooth spatial process, so we fix the differentiability parameter of the state $Z$ to be $q_z = 2$. The model error $D$ was not very smooth and similarly we fix $q_d = 0$. Furthermore, both plots in Figure 7.4 showed the presence of at most four "wiggles" and thus a reasonable $n$ is 10. We also explored $n = 0$ and 20. We report on the following parameter values that gave the best overall forecast skill (ES):

$$q_z = q_{dz} = 2, q_d = 0, n = 0,$$
$$\hat{\sigma}_d^2 = 142.58, \hat{\theta}_d = (\tfrac{31.00}{960} \times 2\pi, 46.82).$$
$$\hat{\sigma}_z^2 = 16.03, \hat{\theta}_z = (\tfrac{87.00}{960} \times 2\pi, 49.98), \text{ and}$$
$$\hat{\rho} = -0.30, \hat{\theta}_{dz} = (\tfrac{7.63}{960} \times 2\pi, 6.00),$$

which we denote as $\hat{\theta}_L^{EDA}$.

For the other set of reported parameter values, we used five-fold cross-validation to choose the differentiability parameters $q_d$, $q_z$, and $q_{dz}$ (varied as 0, 1, and 2) and the maximum period parameter $n$ (varied as multiples of 10

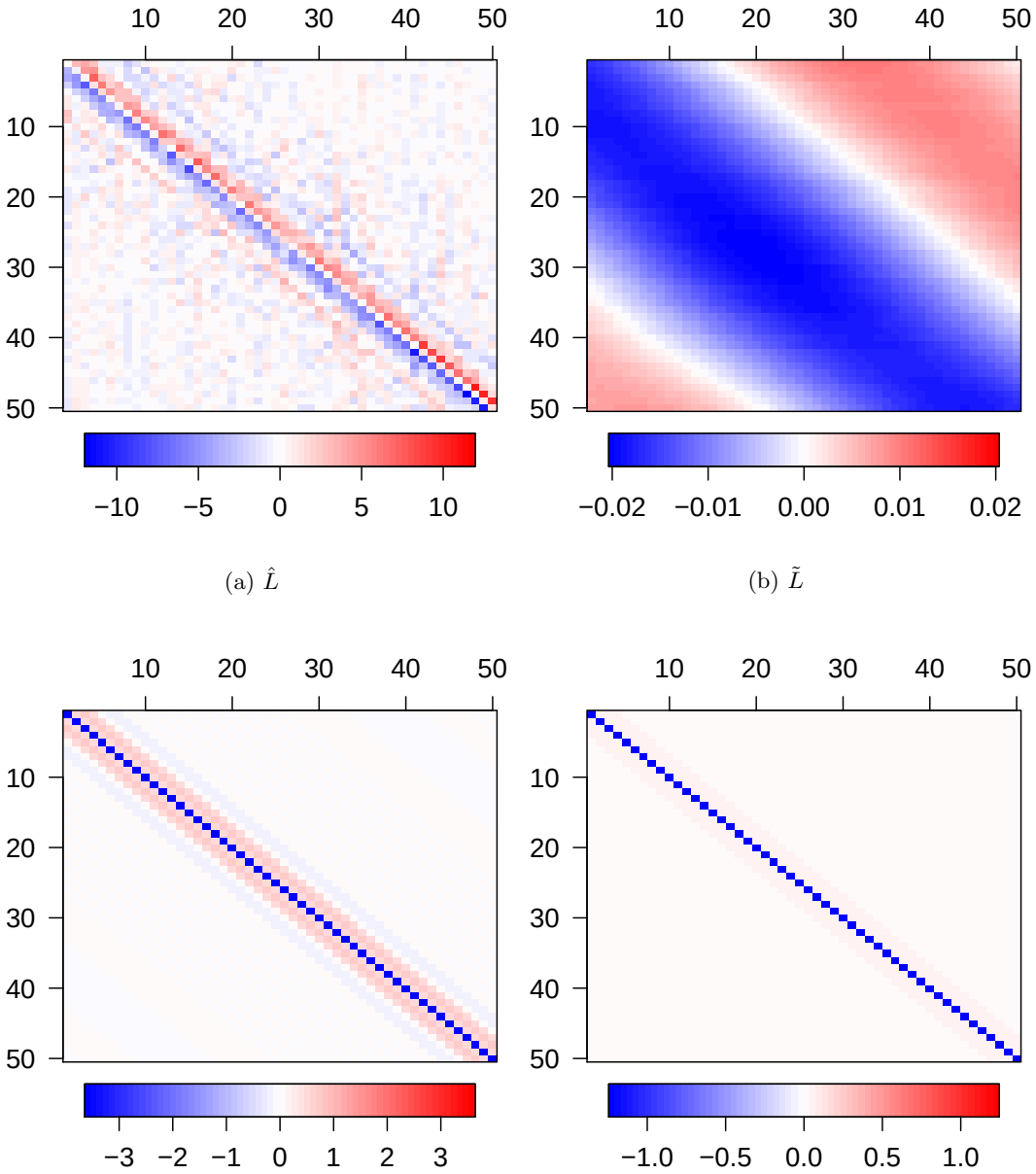
between 0 and 100, inclusive). The following set of parameter values minimized the out-of-sample root-mean-square error (RMSE):

$$\begin{gathered} q_z = q_{dz} = 0, q_d = 2, n = 0, \\ \hat{\sigma}_d^2 = 133.09, \hat{\theta}_d = (\tfrac{11.97}{960} \times 2\pi, 49.99). \\ \hat{\sigma}_z^2 = 2.34, \hat{\theta}_z = (\tfrac{480.00}{960} \times 2\pi, 19.81), \text{ and} \\ \hat{\rho} = -0.068, \hat{\theta}_{dz} = (\tfrac{478.13}{960} \times 2\pi, 50.00); \end{gathered}$$

which we denote as $\hat{\theta}_L^{CV}$.

The out-of-sample RMSE is the smallest for the optimal linear correction given by $\hat{L}$, followed by $\tilde{L}$, and then $L(S, S; \cdot)$ for all combinations of settings. Oddly, even though a strong hole effect is observed (see Figure 7.4), both sets of parameter values do not include the exponentially damped cosine part, i.e., $n = 0$. Upon closer inspection, this result is explained by the results from Figure 7.8: both have the same fitted period parameter value of 0.18 when $n$ is chosen to be 20. Extrapolating this result to the calculation of $L(S, S; \cdot)$, the same fitted value is effectively canceling out the cosine components present in both correlation functions, since the inverse of $C_{q_z,n}^{WC}$ is multiplied by $C_{q_{dz},n}^{WC}$.

Figure 7.9 plots the values of the linear corrections $\hat{L}$, $\tilde{L}$, $L(S, S; \hat{\theta}_L^{EDA})$, and $L(S, S; \hat{\theta}_L^{CV})$; the linear corrections are $960 \times 960$ matrices and the figure only shows the first $50 \times 50$ elements. Compared to $\tilde{L}$, the low-rank linear correction $L(S, S; \cdot)$ better captures the general shape and magnitude of $\hat{L}$. Further comparing $L(S, S; \cdot)$ to $\tilde{L}$, notice that $L(S, S; \cdot)$ better captures the structure of $\hat{L}$. The difference is explained by the choice of regularization: the regularization to obtain $L(S, S; \cdot)$ is cognizant of the spatial structure of the underlying processes and therefore correctly puts higher weights on parts of the state-space that are more important in predicting the model error $D$. In particular, the correlation between the model error $D(t, s)$ and $Z(t, s')$ is large for nearby spatial locations $s, s'$ and therefore nearby states have a larger influence in predicting model error. On the other hand, the regularization to obtain $\tilde{L}$ is oblivious to the spatial structure of the correlation, diluting the weights to all parts of the space. The low-rank linear correction $L(S, S; \cdot)$, however, does not quite correctly capture the sign of the weights in $\hat{L}$, whose superdiagonal entries have large positive weights while the subdiagonal entries have large negative weights. This is not captured by $L(S, S; \cdot)$ because the correlation is assumed to be isotropic and hence depends only on distance, i.e., $\|s - s'\|$ as opposed to $s - s'$.




(a) $\hat{L}$ (b) $\tilde{L}$

(c) $L(S, S; \hat{\theta}_L^{EDA})$ (d) $L(S, S; \hat{\theta}_L^{CV})$

Figure 7.9: *Lorenz 2005: first* $50 \times 50$ *elements of the linear corrections*

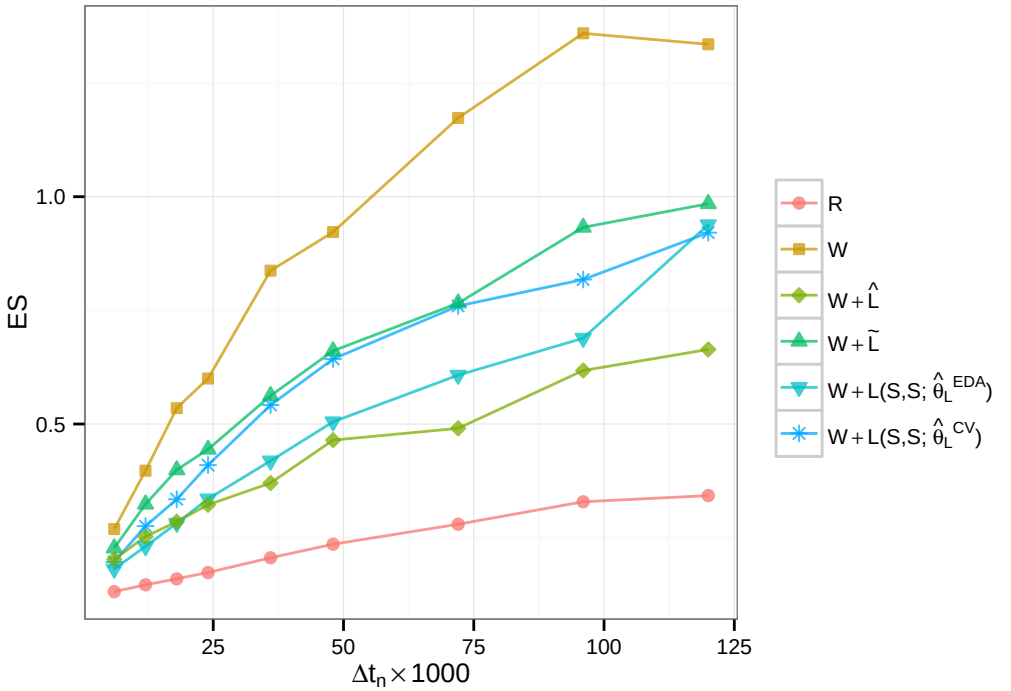



Figure 7.10: *Lorenz 2005: forecast skill of the linear corrections under unrealistic assumptions.* The median ES score of forecasts in the time period where $\sum_n \Delta t_n$ is between $13,140 \times 0.001$ and $17,520 \times 0.01$. The $x$-axis represents the forecast lead times $\Delta t_n$: larger values represent increasing nonlinearities in the state-space model.

### Forecast skill of the linear corrections

We evaluate the forecast skills of the following state transition models: the true model $R$, the wrong model $W$, and the improved model $\widetilde{W}$ with linear corrections $\hat{L}$, $\tilde{L}$, $L(S,S;\hat{\theta}_L^{EDA})$, and $L(S,S;\hat{\theta}_L^{CV})$. Before we do that, we first describe the state-space model and filtering process. We generated measurements from a partially observed system in which every other state location is observed with a standard deviation of one. The forecast lead time $\Delta t_n$ is varied between $6 \times 0.001$ and $120 \times 0.001$ to observe the model improvements as a function of increasing nonlinearities of the state transition model. We apply the deterministic square-root EnKF with localization halfwidth $\frac{5}{960} \times \pi$ and tapering with the Gaspari-Cohn function. The ensemble size is chosen to be $M = 200$. We also estimated an inflation parameter with the method of artificial evolution of parameters; we used the default DART settings as described in Section 5.5.2.

Figure 7.10 reports the median ES for the time period where $\sum_n \Delta t_n$ is between $13,140 \times 0.001$ and $17,520 \times 0.01$. All improved models $\widetilde{W}$ perform

better than the imperfect model $W$ for all forecast lead times $\Delta t_n$. For almost all forecast lead times, Leith's linear correction $\hat{L}$ performed the best, Danforth et al.'s linear correction $\tilde{L}$ performed the worst, and the low-rank linear correction $L(S, S; \cdot)$ performed somewhere in between. This is surprising since the low-rank linear corrections $L(S, S; \cdot)$ had larger out-of-sample RMSE than $\tilde{L}$. Even more surprisingly, the low-rank linear correction with parameter values $\hat{\theta}_L^{EDA}$ performed better the one chosen via cross-validation ($\hat{\theta}_L^{CV}$) for all forecast lead times and better than $\hat{L}$ for forecast lead times $6 \times 0.001$, $12 \times 0.001$, and $18 \times 0.001$.

There are many possible reasons for the conflicting result. The out-of-sample RMSE is calculated under the assumption of temporal independence, which is an assumption that need not apply to state-space models. Therefore, the best model for prediction under the assumptions made by Leith (1978) does not necessarily mean that it is also the model with the best forecast skill. Furthermore, the parameter value found could be a local maxima instead of a global maxima since the existence of a unique maximum likelihood estimator is not guaranteed for stationary Gaussian processes (Zimmerman, 2010)). Another possibility is the choice of the maximization procedure used: the Nelder-Mead method is known to converge to a non-stationary point (McKinnon, 1998).

The forecasts from the low-rank linear correction are not sensitive to the choice of the integer-valued parameters, i.e., differentiability parameter $q$ and the maximum period parameter $n$. To test our hypothesis, we filtered the model with other estimated parameters $\hat{\theta}_L$ fixed at different values of $q$ and $n$. Provided that the value of $n$ is "reasonable", i.e., had values of up to 50, many of these low-rank linear corrections have better forecast skill (lower ES values) than the model with the linear correction $\tilde{L}$ and many more are better than the imperfect model $W$. This is reassuring: the massive effort that we employed in this section to choose the integer-valued parameters is not required to improve forecast skill over the imperfect model $W$.

## 7.4 Low-rank linear correction for state-space models

In the last section, we demonstrated the effectiveness of the low-rank linear correction in improving nonlinear model error. Unfortunately, that low-rank linear correction cannot be directly applied due to the unrealistic assumptions made in Leith (1978): the true state cannot measured without error and the

model error cannot be obtained because it is impossible to evaluate the true model $R$. The form of the low-rank linear correction from the last section does, however, provide guidance on how to construct a low-rank linear correction that is applicable to state-space models.

Instead of approximating the continuous state transition model $R[x(t); \theta_R] \approx W[x(t); \theta_W]$, suppose that a better approximation can be made with a linear correction:

$$R[x(t); \theta_R] \approx W[x(t); \theta_W] + L(\theta_t)x(t),$$

where $L(\cdot)$ is a $d_x \times d_x$ matrix that depends on the parameter $\theta_t$. Unlike in the last section, the parameter of the linear mapping is allowed to vary in time and thus temporal independence is not assumed. It is possible to assume temporal independence by replacing the time-varying parameter $\theta_t$ with a static parameter $\theta$, but that assumption is not made here for more flexibility. To parametrize a low-rank linear correction $L(\cdot)$, let's examine the form of the low-rank linear correction from the previous section (see Equation (7.6)). In a state-space model, the model error is to be corrected at the same locations of the state, so $S_d = S_x$ can be replaced with a single vector of spatial locations $S$. Equation (7.6) provides one possible form for $L(\cdot)$:

$$L(\theta_t) = \rho_t \frac{\sigma_{t,d}}{\sigma_{t,x}} C(S, S; \theta_{t,dx}) C(S, S; \theta_{t,x})^{-1}$$

with parameter $\theta_t = (\rho_t, \sigma_{t,d}, \sigma_{t,x}, \theta_{t,x}, \theta_{t,dx})$. The problem with this particular linear correction is identifiability of the three parameters $\rho_t, \sigma_{t,d}$, and $\sigma_{t,x}$. This is easily remedied by replacing the scalar term $\rho_t \frac{\sigma_{t,d}}{\sigma_{t,x}}$ with one parameter $\beta_t$, providing another form for $L(\cdot)$:

$$L(\theta_t) = \beta_t C(S, S; \theta_{t,dx}) C(S, S; \theta_{t,x})^{-1}$$

with parameter $\theta_t = (\beta_t, \theta_{t,x}, \theta_{t,dx})$. Not only does this form eliminate identifiability issues, it also reduces the number of parameters to be estimated. We end this section by demonstrating the effectiveness in the low-rank linear correction in improving the imperfect model $W$ of the Lorenz 2005 system from Example 6.1.1. Unlike the previous section, the parameters of the low-rank linear correction are estimated while filtering the state-space model with the improved model $\widetilde{W}$ as the state transition model.

### 7.4.1 Demonstration: Lorenz 2005

For the same reasons discussed in Section 7.3.1, the state and its relationship to model error are assumed to be second-order isotropic stationary. The forms

of their covariances are both chosen to be the Wendland exponentially damped cosine function. These choices lead to the following linear correction:

$$L(\theta_t) = \beta_t C^{WC}_{q_{t,dx},n}(S, S; \theta_{t,dx}) C^{WC}_{q_{t,x},n}(S, S; \theta_{t,x})^{-1}, \tag{7.8}$$

where $\theta_t = (\beta_t, \theta_{t,dx}, \theta_{t,x})$, $\theta_{t,dx} = (c_{t,dx}, \tau_{t,dx}, p_{t,dx})$, and $\theta_{t,x} = (c_{t,x}, \tau_{t,x}, p_{t,x})$. The parameters $q_{t,\cdot}$, $n$, $c_{t,\cdot}$, $\tau_{t,\cdot}$, and $p_{t,\cdot}$ are the differentiability, maximum wavelength, range, shape, and period fraction parameters, respectively. If $n$ is fixed to a value greater than zero, the maximum number of parameters to be estimated is 9. If $n$ is fixed to zero, there is no need to estimate the period fraction parameters $p_{t,\cdot}$, this reduces the number of parameters to be estimated to 6. This number can be further reduced by fixing the differentiability parameters $q_{t,\cdot}$ or shape parameters $\tau_{t,\cdot}$. No matter which values are fixed, the number of parameters to be estimated for this low-rank linear correction is far fewer than $O(d_z^2)$ required by Leith's linear correction.

The same state-space model settings are used as outlined in Section 7.3.1. The parameters of the linear correction are estimated with and without the assumption of temporal independence; the latter assumes that the parameters are static. In both cases, we apply the EnKF-APF algorithm developed in Chapter 5 to estimate the parameters. When not assuming temporal independence, the parameters are estimated in an online manner by simply applying artificial evolution of parameters. When assuming that the parameters are static, the parameters are estimated with 25 runs of the iterated filter (IF) (Ionides et al., 2015).

We use different parameter settings to test the robustness of the low-rank linear correction. The maximum period parameter $n$ is fixed and varied among values of 0, 10, and 20, and 100. The differentiability parameters $q_{t,\cdot}$ are both estimated and fixed to the values guided by the results from Section 7.3.1. In the last section, the shape parameters $\tau_{t,\cdot}$ were generally difficult to estimate and thus we test both estimating them and fixing them to be the minimum value that maintains the validity of the Wendland exponentially damped cosine function. The rest of the parameters, including the inflation parameter $\lambda_t$, are estimated with the EnKF-APF algorithm and artificial evolution of parameters or IF, depending on the assumptions of the model. The perturbation density is chosen to be a product of truncated normal distributions with initial standard deviations and constraints as listed in Table 7.1.

Figure 7.11 reports the median ES during the time period $\sum_n \Delta t_n$ between $13,140 \times 0.001$ and $17,520 \times 0.01$ under the best parameter settings. Allowing the parameters to vary with time, the best performing set of parameters (denoted as $\hat{\theta}_t$) has values $n = 0$, $q_{dz} = q_z = 0$, and fixed $\tau_{dz}$ and $\tau_z$. When assuming temporal stationarity, the best performing set of parameters (denoted

| parameter | initial sd | minimum | maximum |
|---|---|---|---|
| $\beta_t$ | 0.1 | -50 | 50 |
| $c_{t,dz}, c_{t,z}$ | $\frac{1}{d_z}\pi$ | 0 | $\pi$ |
| $\tau_{t,dz}, \tau_{t,z}$ | 0.1 | $2(q_{t,\cdot}+1)$ | 10 |
| $q_{t,dz}, q_{t,z}$ | 0.1 | 0 | 2 |
| $p_{t,dz}, p_{t,z}$ | 0.01 | 0 | 1 |
| $\lambda_t$ | 0.6 | 1 | 20 |

Table 7.1: *Lorenz 2005 parameter estimation settings.* The perturbation density $q(\theta_{t_n} \,|\, \theta_{t_{n-1}})$ is a product of truncated normal distributions with the above parameters.

as $\hat{\theta}$) has values $n = 0$ and $q_{dz} = q_z = 0$. Notice that both sets have similar fixed values. The figure also includes Leith's optimal linear correction $\hat{L}$ from Figure 7.10 for comparison.

The improved models $\widetilde{W}$ have better forecast skill than the imperfect model $W$ for all forecast lead times $\Delta t_n$. In fact, both improved models, $L(\hat{\theta}_t)$ and $L(\hat{\theta})$, have similar skill to each other and to Leith's optimal linear correction $\hat{L}$. Since the model error comes from missing the small-scale dynamics of the system, we suspect the comparable performance is due to the insensitivity of the forecasts to relatively small changes in the values of the parameters that govern the model error. This suspicion is corroborated by our preliminary studies on the EnKF-APF: when estimating the parameters of the Lorenz 2005-III system, the parameters that govern the large-scale dynamics of the system (i.e., $K$ and $F$) showed clear convergence while the ones that govern the small-scale dynamics (i.e., $I$, $b$, and $c$) did not. Therefore, for this particular example and similar atmospheric models, a practitioner can reduce the computational expense of estimating parameters by assuming temporal independence: the parameters can be estimated with collected measurements and these estimated parameters can subsequently be plugged into the improved model for future forecasts.

When parameters are fixed by our exploratory data analysis, we obtain poor results. For example, when we fix the differentiability parameters $q_{dz}$ and $q_z$ to be 2, the forecast skill deteriorated: the "improved" model $\widetilde{W}$ consistently has higher ES values than the imperfect model $W$ for all forecast lead times. Since the cross-validation procedure fixed the differentiability pa-

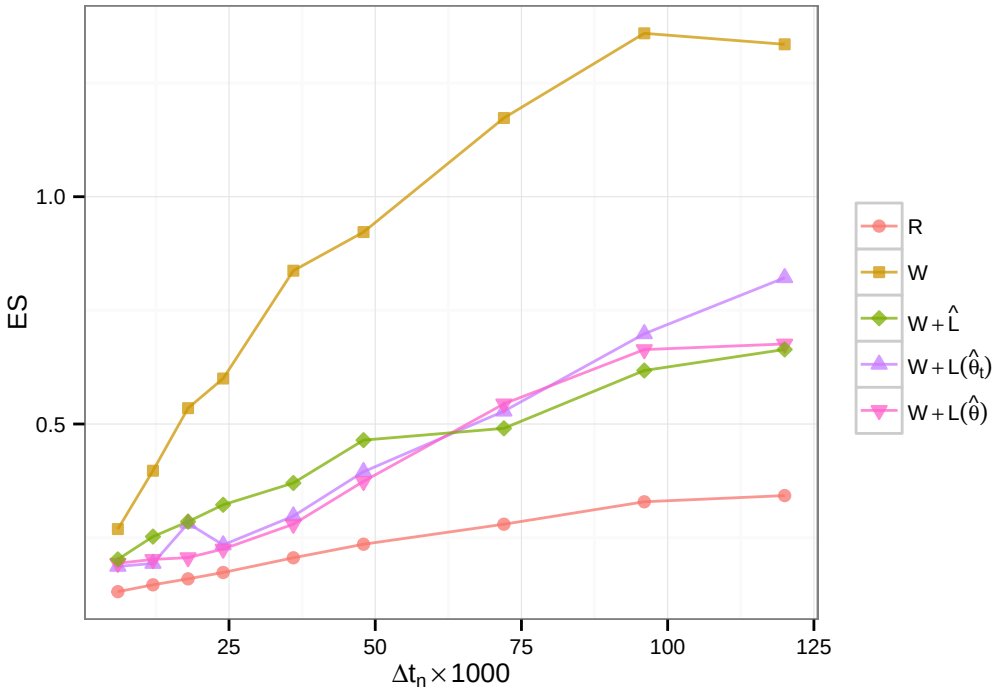



Figure 7.11: *Lorenz 2005: forecast skill of the linear corrections for state-space models.* The median ES score of forecasts in the time period where $\sum_n \Delta t_n$ is between $13,140 \times 0.001$ and $17,520 \times 0.01$. The $x$-axis represents the forecast lead times $\Delta t_n$: larger values represent increasing nonlinearities in the state-space model.

rameters at zero, perhaps our exploratory data analysis from the last section was misguided. Though this is the case, the low-rank linear correction is effective at improving nonlinear model error with a caveat: the practitioner should be careful about how the parameter values are fixed—it is generally safer to estimate the parameters.

## 7.5 Future work

In this section, we developed a method to correct for nonlinear model error for high-dimensional state spaces. Even though the correction is linear, the correction has the ability to capture more complicated phenomenon such as the quadratic nature of the model error of the Lorenz 2005 system. However, even though the Lorenz 2005 system has characteristics of real atmospheric models, it is not clear that the principles learned from the Lorenz 2005 system directly carry over to real high-dimensional applications. In particular, the parametrization of the low-rank linear correction in Equation (7.8) is guided

by our exploratory data analysis of the true state and the model error, both of which are impossible to obtain in practice. Nevertheless, because of the flexibility of the Wendland exponentially cosine function, we believe that the parametrization is applicable to many real atmospheric models. To be sure, it is worth applying this particular parametrization of the low-rank linear correction to a real atmospheric model.

If it turns out that the parametrization is not flexible enough, it is worth further exploring the set of approximations introduced by DelSole and Hou (1999) (discussed in Section 6.3) as a tool for exploratory data analysis. If their approximations reflect the structure of the true state and model error, these approximations can be used to investigate the correlation structure of the true state and the model error, which can then be used to parametrize the low-rank linear correction. Furthermore, the tool may also help understand the extent of the model error and if it is worthwhile endeavour to correct it.

# Bibliography

M U Altaf, T Butler, Xiaodong Luo, C Dawson, T Mayo, and Ibrahim Hoteit. Improving Short-Range Ensemble Kalman Storm Surge Forecasting Using Robust Adaptive Inflation. Monthly Weather Review, 141(8):2705–2720, 2013. ISSN 0027-0644. doi: 10.1175/MWR-D-12-00310.1. URL http://journals.ametsoc.org/doi/abs/10.1175/MWR-D-12-00310.1.

Jeffrey L Anderson. An Ensemble Adjustment Kalman Filter for Data Assimilation. Monthly Weather Review, 129(12):2884–2903, 2001. ISSN 0027-0644. doi: 10.1175/1520-0493(2001)129⟨2884:AEAKFF⟩2.0.CO;2.

Jeffrey L Anderson. An adaptive covariance inflation error correction algorithm for ensemble filters. Tellus, Series A: Dynamic Meteorology and Oceanography, 59(2):210–224, mar 2007. ISSN 0280-6495. doi: 10.1111/j.1600-0870.2006.00216.x. URL http://tellusa.net/index.php/tellusa/article/view/14925.

Jeffrey L Anderson. Spatially and temporally varying adaptive covariance inflation for ensemble filters. Tellus, Series A: Dynamic Meteorology and Oceanography, 61(1):72–83, 2009. ISSN 02806495. doi: 10.1111/j.1600-0870.2008.00361.x. URL http://tellusa.net/index.php/tellusa/article/view/15524.

Jeffrey L Anderson and Stephen L Anderson. A Monte Carlo Implementation of the Nonlinear Filtering Problem to Produce Ensemble Assimilations and Forecasts. Monthly Weather Review, 127(12):2741–2758, 1999. ISSN 0027-0644. doi: 10.1175/1520-0493(1999)127⟨2741:AMCIOT⟩2.0.CO;2.

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(3):269–342, jun 2010. ISSN 13697412. doi:

10.1111/j.1467-9868.2009.00736.x. URL http://doi.wiley.com/10.1111/j.1467-9868.2009.00736.x.

Thomas Bengtsson, Chris Snyder, and Douglas W Nychka. Toward a nonlinear ensemble filter for high-dimensional systems. Journal of Geophysical Research, 108, 2003. ISSN 0148-0227. doi: 10.1029/2002JD002900.

Thomas Bengtsson, Peter J Bickel, and Bo Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In Probability and statistics: Essays in honor of David A. Freedman, volume 2 of Institute of Mathematical Statistics Collections, pages 316–334. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2008. ISBN 978-0-940600-74-4. doi: 10.1214/193940307000000518.

Peter J Bickel and Elizaveta Levina. Regularized estimation of large covariance matrices. The Annals of Statistics, 36(1):199–227, feb 2008. ISSN 0090-5364. doi: 10.1214/009053607000000758. URL http://projecteuclid.org/euclid.aos/1201877299.

Christopher M Bishop. Pattern recognition and machine learning. Springer, New York, 2006.

Richard S Bucy and Rudolph Emil Kalman. New Results in Linear Filtering and Prediction Theory. Journal of Basic Engineering, 83(1):95–108, 1961.

Gerrit Burgers, Peter Jan van Leeuwen, and Geir Evensen. Analysis Scheme in the Ensemble Kalman Filter. Monthly Weather Review, 126(6):1719–1725, 1998.

Olivier Cappé, Eric Moulines, and Tobias Rydén. Inference in hidden Markov models. Springer Science & Business Media, 2006.

Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. Proceedings of the IEEE, 95(5):899–924, 2007. ISSN 00189219. doi: 10.1109/JPROC.2007.893250.

George Casella and Roger L Berger. Statistical inference. Duxbury, Pacific Grove, CA, 2nd edition, 2002.

Zhe Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. Technical report,

2003. URL http://www2.ee.kuas.edu.tw/{~}lwang/WWW/BayesianFilteringFromKalmanFiltersToParticleFiltersAndBeyond.pdf.

Jean-Paul Chiles and Pierre Delfiner. Geostatistics: Modeling spatial uncertainty. John Wiley & Sons, Inc, New York, 1st edition, 1999.

Dan Crisan and Arnaud Doucet. A Survey of Convergence Results on Particle Filtering Methods for Practitioners. IEEE Transactions on Signal Processing, 50(3):736–746, 2002.

Dan Crisan and Joaquin Miguez. Particle-Kernel Estimation of the Filter Density in State-Space Models. Bernoulli, 20(4):1879–1929, 2014. ISSN 13507265. doi: 10.3150/13-BEJ545. URL http://www.bernoulli-society.org/index.php/publications/bernoulli-journal/bernoulli-journal-papers.

P Cvitanović, R Artuso, R Mainieri, G Tanner, and G Vattay. Chaos: Classical and Quantum. 2015.

Christopher M Danforth and Eugenia Kalnay. Impact of online empirical model correction on nonlinear error growth. Geophysical Research Letters, 35(24):L24805, dec 2008. ISSN 0094-8276. doi: 10.1029/2008GL036239. URL http://doi.wiley.com/10.1029/2008GL036239.

Christopher M Danforth, Eugenia Kalnay, and Takemasa Miyoshi. Estimating and Correcting Global Weather Model Error. Monthly Weather Review, 135 (2):281–299, feb 2007. ISSN 0027-0644. doi: 10.1175/MWR3289.1. URL http://journals.ametsoc.org/doi/abs/10.1175/MWR3289.1.

Dick P Dee and Arlindo M da Silva. Data assimilation in the presence of forecast bias. Quarterly Journal of the Royal Meteorological Society, 124: 269–295, 1998.

Pierre Del Moral. Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications. Springer-Verlag, New York, 2004.

Pierre Del Moral. Mean field simulation for Monte Carlo integration. CRC Press, 2013.

Timothy DelSole and Arthur Y Hou. Empirical Correction of a Dynamical Model. Part I: Fundamental Issues. Monthly Weather Review, 127(11): 2533–2545, 1999.

Timothy Delsole and Xiaosong Yang. State and parameter estimation in stochastic dynamical models. Physica D: Nonlinear Phenomena, 239(18): 1781–1788, 2010. ISSN 01672789. doi: 10.1016/j.physd.2010.06.001. URL http://dx.doi.org/10.1016/j.physd.2010.06.001.

Randal Douc, Olivier Cappé, and Eric Moulines. Comparison of Resampling Schemes for Particle Filtering. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, pages 64–69, 2005.

Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fiteen years later. In Dan Crisan and Boris Rozovskii, editors, Handbook of Nonlinear Filtering, chapter 24, pages 656–704. Oxford University Press, 2011. ISBN 978-0199532902. doi: 10.1.1.157.772. URL http://webcat.warwick.ac.uk/record=b2490036{~}S1.

Arnaud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russell. Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence, pages 176–183, 2000.

Arnaud Doucet, Nando De Freitas, and Neil J Gordon. An Introduction to Sequential Monte Carlo Methods. In Arnaud Doucet, Nando De Freitas, and Neil Gordon, editors, Sequential Monte Carlo methods in practice, chapter 1, pages 3–14. Springer, New York, 2001.

Arnaud Doucet, Michael K Pitt, G Deligiannidis, and Robert Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. Biometrika, pages 1–19, 2014. doi: 10.1093/biomet/asu075. URL http://www.newton.ac.uk/programmes/MCM/seminars/2014042311051.pdfhttp://arxiv.org/abs/1404.6909.

Garry Einicke. Nonlinear Prediction, Filtering and Smoothing. In Smoothing, Filtering and Prediction - Estimating The Past, Present and Future, chapter 10. InTech, 2012. URL http://www.intechopen.com/books/smoothing-filtering-and-prediction-estimating-the-past-present-and-future/nonlinear-prediction-filtering-and-smoothing.

Geir Evensen. Using the Extended Kalman Filter with a Multilayer Quasi-Geostrophic Ocean Model. Journal of Geophysical Research, 97(C11): 17905–17924, 1992. doi: Doi10.1029/92jc01972. URL <GotoISI>://WOS:A1992JZ33900016.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. Journal of Geophysical Research, 99(C5):10143–10162, 1994.

Geir Evensen. The Ensemble Kalman Filter: theoretical formulation and practical implementation. Ocean Dynamics, 53(4):343–367, nov 2003. ISSN 1616-7341. doi: 10.1007/s10236-003-0036-9. URL http://link.springer.com/10.1007/s10236-003-0036-9.

Geir Evensen. Data Assimilation: The Ensemble Kalman Filter. Springer-Verlag, Berlin, 2nd edition, 2009.

Marco Luca Flavio Frei and Hans R Künsch. Sequential State and Observation Noise Covariance Estimation Using Combined Ensemble Kalman and Particle Filters. Monthly Weather Review, 140(2007):1476–1495, 2012. ISSN 0027-0644. doi: 10.1175/MWR-D-10-05088.1.

Marco Luca Flavio Frei and Hans R Künsch. Bridging the ensemble Kalman and particle filters. Biometrika, 100(4):781–800, jul 2013. ISSN 0006-3444. doi: 10.1093/biomet/ast020. URL http://biomet.oxfordjournals.org/cgi/doi/10.1093/biomet/ast020.

Bernard Friedland. Treatment of Bias in Recursive Filtering. IEEE Transactions on Automatic Control, 14(4):359–367, 1969.

Bernard Friedland. Notes on Separate-Bias. IEEE Transactions on Automatic Control, 23(4):735–738, 1978.

Reinhard Furrer and Thomas Bengtsson. Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants. Journal of Multivariate Analysis, 98(2):227–255, feb 2007. ISSN 0047259X. doi: 10.1016/j.jmva.2006.08.003. URL http://linkinghub.elsevier.com/retrieve/pii/S0047259X06001187.

Gregory Gaspari and Stephen E Cohn. Construction of correlation functions in two and three dimensions. Quarterly Journal of the Royal Meteorological Society, 125(April 1998):723–757, 1999.

Alan E Gelfand and Sudipto Banerjee. Multivariate spatial process models. In Alan E Gelfand, Peter J Diggle, Montserrat Fuentes, and Peter Guttorp, editors, Handbook of Spatial Statistics, chapter 28. Boca Raton, FL, 2010.

Tilmann Gneiting. Strictly and non-strictly positive definite functions on spheres. Bernoulli, 19(4):1327–1349, 2013. ISSN 1350-7265. doi: 10.3150/12-BEJSP06. URL http://projecteuclid.org/euclid.bj/1377612854.

Tilmann Gneiting and Peter Guttorp. Continuous-parameter stochastic process theory. In Alan E Gelfand, Peter J. Diggle, Montserrat Fuentes, and Peter Guttorp, editors, Handbook of Spatial Statistics, chapter 2, pages 17–28. Taylor and Francis/CRC, Boca Raton, FL, 2010.

Tilmann Gneiting and Adrian E Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. Journal of the American Statistical Association, 102(477):359–378, mar 2007. ISSN 0162-1459. doi: 10.1198/016214506000001437. URL http://www.tandfonline.com/doi/abs/10.1198/016214506000001437.

Tilmann Gneiting, Larissa I Stanberry, Eric Grimit, Leonhard Held, and Nicholas A Johnson. Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. Test, 17:211–235, 2008. ISSN 11330686. doi: 10.1007/s11749-008-0114-x.

Neil J Gordon, D J Salmond, and A F M Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings F (Radar and Signal Processing), 140(2):107–113, 1993.

Georg A Gottwald and Andrew J Majda. A mechanism for catastrophic filter divergence in data assimilation for sparse observation networks. Nonlinear Processes in Geophysics, 20(5):705–712, 2013. ISSN 1607-7946. doi: 10.5194/npg-20-705-2013. URL http://www.nonlin-processes-geophys.net/20/705/2013/.

Thomas M Hamill, Jeffrey S Whitaker, and Chris Snyder. Distance-Dependent Filtering of Background Error Covariance Estimates in an Ensemble Kalman Filter. Monthly Weather Review, 129(11):2776–2790, 2001.

John Harlim and Andrew J Majda. Catastrophic filter divergence in filtering nonlinear dissipative systems. Communications in Mathematical Sciences, 8(1):27–43, 2010. ISSN 15396746. doi: 10.4310/CMS.2010.v8.n1.a3.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2nd edition, 2009.

Jeroen D. Hol, Thomas B. Schön, and Fredrik Gustafsson. On resampling algorithms for particle filters. NSSPW - Nonlinear Statistical Signal Processing Workshop 2006, 2006. doi: 10.1109/NSSPW.2006.4378824.

P L Houtekamer and Herschel L Mitchell. Data Assimilation Using an Ensemble Kalman Filter Technique. Monthly Weather Review, 126(3):796–811, 1998.

Markus Hürzeler and Hans R Künsch. Monte Carlo Approximations for General State-Space Models. Journal of Computational and Graphical Statistics, 7(2):175–193, jun 1998. ISSN 1061-8600. doi: 10.1080/10618600.1998.10474769. URL http://www.tandfonline.com/doi/abs/10.1080/10618600.1998.10474769.

M B Ignagni. An Alternate Derivation and Extension of Friedland's Two-Stage Kalman Estimator. IEEE Transactions on Automatic Control, 26(3):520–523, 1981.

M B Ignagni. Separate-Bias Kalman Estimator with Bias State Noise. IEEE Transactions on Automatic Control, 35(3):338–341, 1990.

Edward L Ionides, C Bretó, and Aaron A King. Inference for nonlinear dynamical systems. Proceedings of the National Academy of Sciences, 103 (49):18438–43, dec 2006. ISSN 0027-8424. doi: 10.1073/pnas.0603181103. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3020138{&}tool=pmcentrez{&}rendertype=abstract.

Edward L Ionides, Anindya Bhadra, Yves Atchadé, and Aaron A King. Iterated filtering. The Annals of Statistics, 39(3):1776–1802, jun 2011. ISSN 0090-5364. doi: 10.1214/11-AOS886. URL http://projecteuclid.org/euclid.aos/1311600283.

Edward L Ionides, Dao Nguyen, Yves Atchadé, Stilian Stoev, and Aaron A King. Inference for dynamic and latent variable models via iterated, perturbed Bayes maps. Proceedings of the National Academy of Sciences, 112 (3):719–724, 2015. ISSN 0027-8424. doi: 10.1073/pnas.1410597112. URL http://www.pnas.org/lookup/doi/10.1073/pnas.1410597112.

Rudolph Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(1):35–45, 1960.

Nikolas Kantas, Arnaud Doucet, Sumeetpal S Singh, Jan M Maciejowski, and Nicolas Chopin. On Particle Methods for Parameter Estimation in General State-Space Models. Statistical Science, pages 1–31, 2015.

Genshiro Kitagawa. A Self-Organizing State-Space Model. Journal of the American Statistical Association, 93(443):1203–1215, 1998.

Francois Le Gland and Nadia Oudjane. Stability and uniform approximation of nonlinear filters using the Hilbert metric and application to particle filters. Annals of Applied Probability, 14(1):144–187, 2004. ISSN 1050-5164. URL http://www.jstor.org/stable/4140493.

Francois Le Gland, Christian Musso, and Nadia Oudjane. An Analysis of Regularized Interacting Particle Methods for Nonlinear Filtering. In Preprints of the 3rd IEEE European Workshop on Computer (Intensive Methods in Control and Data Processing), pages 167–174, Prague, 1998.

Francois Le Gland, Valerie Monbet, and Vu-Duc Tran. Large Sample Asymptotics for the Ensemble Kalman Filter. Technical report, 2009.

Jing Lei, Peter J Bickel, and Chris Snyder. Comparison of Ensemble Kalman Filters under Non-Gaussianity. Monthly Weather Review, 138(4):1293–1306, 2010. ISSN 0027-0644. doi: 10.1175/2009MWR3133.1.

C E Leith. Objective Methods for Weather Prediction. Annual Review of Fluid Mechanics, 10(1):107–128, 1978.

Jane Liu and Mike West. Combined Parameter and State Estimation in Simulation-Based Filtering. In Arnaud Doucet, Nando De Freitas, and Neil Gordon, editors, Sequential Monte Carlo methods in practice, chapter 10, pages 197–223. Springer, New York, 2001.

David M Livings, Sarah L Dance, and Nancy K Nichols. Unbiased ensemble square root filters. Physica D: Nonlinear Phenomena, 237(8):1021–1028, jun 2008. ISSN 01672789. doi: 10.1016/j.physd.2008.01.005. URL http://linkinghub.elsevier.com/retrieve/pii/S0167278908000122.

Edward N Lorenz. Deterministic Nonperiodic Flow. Journal of Atmospheric Science, 20:130–141, 1963.

Edward N Lorenz. Irregularity: a fundamental property of the atmosphere. Tellus, Series A: Dynamic Meteorology and Oceanography, pages 98–110, 1984. ISSN 0280-6495. doi: 10.3402/tellusa.v36i2.11473.

Edward N Lorenz. Predictability–A problem partly solved. In Tim Palmer and Renate Hagedorn, editors, Predictability of Weather and Climate, pages 40–58. Cambridge University Press, 1996.

Edward N Lorenz. Designing Chaotic Models. Journal of the Atmospheric Sciences, 62(5):1574–1587, may 2005. ISSN 0022-4928. doi: 10.1175/JAS3430.1. URL http://journals.ametsoc.org/doi/abs/10.1175/JAS3430.1.

Edward N Lorenz and Kerry A Emanuel. Optimal Sites for Supplementary Weather Observations: Simulation with a Small Model. Journal of the Atmospheric Sciences, 55(3):399–414, 1998. ISSN 0022-4928. doi: 10.1175/1520-0469(1998)055⟨0399:OSFSWO⟩2.0.CO;2.

Xiaodong Luo and Ibrahim Hoteit. Ensemble Kalman Filtering with Residual Nudging: An Extension to State Estimation Problems with Nonlinear Observation Operators. Monthly Weather Review, 142(10):3696–3712, 2014. ISSN 0027-0644. doi: 10.1175/MWR-D-13-00328.1. URL http://journals.ametsoc.org/doi/abs/10.1175/MWR-D-13-00328.1.

Jan Mandel, Loren Cobb, and Jonathan D Beezley. On the Convergence of the Ensemble Kalman Filter. Applications of Mathematics, 56(6):533–541, jan 2009. URL http://arxiv.org/abs/0901.2951v2.

James E Matheson and Robert L Winkler. Scoring Rules for Continuous Probability Distributions. Management Science, 22(10):1087–1096, 1976. ISSN 0025-1909. doi: 10.1287/mnsc.22.10.1087.

K I M McKinnon. Convergence of the Nelder-Mead simplex method to a nostationary point. SIAM Journal on Optimization, 9(1):148–158, 1998.

Meteorological Assimilation Data Ingest System (MADIS). MADIS Meteorological Surface Station Counts, oct 2014. URL https://madis.ncep.noaa.gov/sfc{_}station{_}counts.shtml.

R N Miller, M Ghil, and F Gauthiez. Advanced Data Assimilation in Strongly Nonlinear Dynamical-Systems, 1994. ISSN 0022-4928.

Takemasa Miyoshi. The Gaussian Approach to Adaptive Covariance Inflation and Its Implementation with the Local Ensemble Transform Kalman Filter. Monthly Weather Review, 139(5):1519–1535, 2011. ISSN 0027-0644. doi: 10.1175/2010MWR3570.1.

Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. Journal of the American Statistical Association, 94(446):590–599, 1999. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1999.10474153.

Carl Edward Rasmussen and Christopher K I Williams. Gaussian processes in machine learning. MIT Press, 2006. URL http://www.springerlink.com/index/3xevuky2p4272n75.pdf.

Håvard Rue and Leonhard Held. Gaussian Markov Random Fields: Theory and Applications. Chapman & Hall/CRC, Boca Raton, FL, 1st edition, 2005.

Bernard W Silverman. Density estimation for statistics and data analysis. CRC Press, Boca Raton, FL, 1986.

Chris Snyder, Thomas Bengtsson, Peter J Bickel, and Jeffrey L Anderson. Obstacles to High-Dimensional Particle Filtering. Monthly Weather Review, 136(12):4629–4640, dec 2008. ISSN 0027-0644. doi: 10.1175/2008MWR2529.1. URL http://journals.ametsoc.org/doi/abs/10.1175/2008MWR2529.1.

Jonathan R Stroud and Thomas Bengtsson. Sequential State and Variance Estimation within the Ensemble Kalman Filter. Monthly Weather Review, 135(9):3194–3208, 2007. ISSN 0027-0644. doi: 10.1175/MWR3460.1.

Michael K Tippett, Jeffrey L Anderson, Craig H Bishop, Thomas M Hamill, and Jeffrey S Whitaker. Ensemble Square Root Filters. Monthly Weather Review, 131(7):1485–1490, jul 2003. ISSN 0027-0644. doi: 10.1175/1520-0493(2003)131⟨1485:ESRF⟩2.0.CO;2. URL http://journals.ametsoc.org/doi/abs/10.1175/1520-0493{%}282003{%}29131{%}3C1485{%}3AESRF{%}3E2.0.CO{%}3B2.

Rudolph van der Merwe. Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models. PhD thesis, 2004.

M P Wand. Error analysis for general multivariate kernel estimators. Journal of Nonparametric Statistics, 2(1):1–15, jan 1992. ISSN 1048-5252. doi: 10.1080/10485259208832538. URL http://www.tandfonline.com/doi/abs/10.1080/10485259208832538.

M P Wand and M C Jones. Kernel Smoothing. CRC Press, New York, 1995.

Daniel S Wilks. Forecast verification. In Statistical methods in the atmospheric sciences, chapter 8. Academic Press, 3rd edition, 2011.

Xiaosong Yang and Timothy Delsole. Using the ensemble Kalman filter to estimate multiplicative model parameters. Tellus, Series A: Dynamic

Meteorology and Oceanography, 61(5):601–609, 2009. ISSN 02806495. doi: 10.1111/j.1600-0870.2009.00407.x.

V P Zastavnyi. On Positive Definiteness of Some Functions. Journal of Multivariate Analysis, 81(73):55–81, 2000. ISSN 0047259X. doi: 10.1006/jmva.1999.1864.

Dale L Zimmerman. Likelihood-Based Methods. In Alan E Gelfand, Peter J Diggle, Montserrat Fuentes, and Peter Guttorp, editors, Handbook of Spatial Statistics, chapter 4. Boca Raton, FL, 2010.

Dusanka Zupanski and Milija Zupanski. Model Error Estimation Employing an Ensemble Data Assimilation Approach. Monthly Weather Review, 134 (May):1337–1354, 2006.