Master Thesis: My Guess is Better Than Yours

Filip Tronarp

5th January 2016

Notation

ΔX_n	:	The forward difference of X_n, i.e $\Delta X_n = X_{n+1} - X_n$.
$p_X(x)$	:	The probability density of the stochastic variable X, the subscript is usually omitted.
$\mathbb{E}\{X\}$	:	The expected value of X, brackets are omitted when the meaning is obvious.
$\mathbb{V}\{X\}$	:	The variance of X, brackets are omitted when the meaning is obvious.
$\mathbb{C}\{X, Y\}$	:	The covariance between X and Y.
$X \perp Y$	:	X and Y are independent.
$X \sim Y$	:	X and Y are equal in sense of distribution.
$X \sim p(x)$	:	X is drawn from the distribution corresponding to the probability density $p(x)$
$\{x_{n':N}\}$	:	shorthand for $\{x_n\}_{n=n'}^{N}$.
I	:	The identity matrix.
A^T	:	The transpose of A.
$A_{i,j}$	:	The element at row i and column j of the matrix A.
$A_{:,j}$	:	The j:th column vector of the matrix A.
$A_{i,:}$	:	The i:th row vector of the matrix A.
$\frac{\partial}{\partial X}$	:	Either the partial derivative with respect to X or the jacobian with respect to X.
∇_X	:	The gradient with respect to X.
$\mathrm{Tr}\{A\}$	:	The trace of A.

Contents

Chapter 1

Introduction

Modeling of and inference in dynamical systems is an issue that frequently arises in a wide array of disciplines across science and engineering. The biologist may want to understand how the environment and the interaction of different species causes the population sizes to change over time. The financial engineer is interested in understanding how the prices of financial products evolve over time. In automatic control the interest lies in accurate descriptions of processes so that suitable control schemes can be developed. What all these people have in common is that they're interested in descriptions of dynamical systems for the purpose of predicting the future behaviour of said systems. There are different approaches to describing dynamical systems, the system can be thought of as operating in continuous or discrete time and it can be thought of as deterministic or stochastic. The aim of this thesis is to survey methods for dealing with dynamical systems in the case when they evolve stochastically in continuous-time. Though more specifically models for continuous-time stochastic processes that are measured at a collection of discrete instants are considered.

1.1 Overview

The stochastic differential equation will in this text be fundamental to the development of a framework in which the behaviour of a continuous-time stochastic process can be expressed intuitively and compactly. Chapter 2 offers a very brief overview of the theoretical aspects of defining the stochastic differential equation. The Wiener process is defined and what it means to integrate a function with respect to a Wiener process is discussed. This makes it possible to construct stochastic differential equations, of particular interest is Itō's interpretation and a modest selection of the more important results are presented. Discrete time approximations will also be discussed as it is essential when dealing with stochastic differential equations numerically.

Chapter 3 deals with extending the stochastic differential equation model by letting it be, perhaps indirectly, measured at a collection of discrete instants under noisy conditions. This raises questions such as "Given the measurements I have up until now, what can I say about the current state of the underlying process?". This is the filtering problem which will briefly be discussed in general terms and a situation for when there is an exact filtering algorithm will be presented. Though most of the emphasis will be on approximate filtering algorithms so that more general classes of state space models can be dealt with.

Chapter 4 considers the case when the behaviour of the stochastic process generated by the stochastic differential equation as well as the measurements depend on a set of unknown parameters. The obvious question is "So I have a bunch of measurements, what are the parameters?". This is the problem of estimating parameters and a brief discussion of what constitutes a good estimator is followed by a discussion of Maximum Likelihood and Quasi Maximum Likelihood for the very frequently occurring situations when the true likelihood is unavailable. Strategies for maximising the likelihood is also considered with particular emphasis on a recent method, Gradient-Based Adaptive Stochastic Search, which is based on stochastic approximation.

In order to evaluate the different methods presented in the previous chapters the results of some simulated experiments are presented in Chapter 5. More precisely the performance of some of the filtering algorithms from Chapter 3 is evaluated. There is also a section evaluating the merits of using the Gradient-Based Adaptive Stochastic Search to maximise the (Quasi) Likelihood.

This thesis ends with Chapter 6 which features a discussion on how the results should be interpreted, potential flaws and rooms for improvement.

Chapter 2

Continuous-Time Stochastic Processes

This chapter outlines the theoretical foundations of the stochastic differential equation which is the fundamental building for the models considered in this thesis. Section 2.1.1 offers a short description of the Wiener process and its' properties. Section 2.1 motivates the need of stochastic differential equations (SDEs) and uses the material from Section 2.1.1 in order to define what is meant by a SDE. A few of it's more fundamental properties are subsequently presented. Section 2.1.3 discusses discretisation of stochastic differential equations. The methods presented here are essential to developing the approximate algorithms for state space models in Chapter 3.

2.1 Stochastic Differential Equations

While the classical methods of time series analysis are often sufficient to model dynamic systems there are some drawbacks. Non-uniformly sampled data may for example cause problems. Since the conditional mean is linear and the conditional variance is constant in linear time series models heteroscedatic data will be a bad experience for the modeler. This can be handled employing non-linear time series models though these models can be hard to interpret. Furthermore it's cumbersome to incorporate domain knowledge, coming from e.g physics, in the modeling procedure for these types of models. Hence the goal of developing modeling techniques with stochastic differential equations is to create a framework in which the modeler can intuitively express partial knowledge of the system being studied as well as making it easy to incorporate additional hypothesis of the systems behaviour. Since the solution to a SDE is a continuous-time process the issue of irregularly sampled data is in principle solved. Though everything has a cost, the theory of SDEs is a lot more complicated, the same goes for implementing filters/smoothers and consequently parameter estimation. The idea behind stochastic differential equations is to mimic the generalisation of deterministic signals governed by a recursive relation to stochastic signals governed by a recursive relation with noise input in the context of differential equations.

Deterministic	**Stochastic**	
$X_t = f(t, X_{t-1})$	$X_t = f(t, X_{t-1}) + g(t, X_{t-1})E_t$	(2.1)
$\frac{\mathrm{d}X_t}{\mathrm{d}t} = f(t, X_t)$	$\frac{\mathrm{d}X_t}{\mathrm{d}t} = f(t, X_t) + g(t, X_t)\frac{\mathrm{d}W_t}{\mathrm{d}t}$	(2.2)

Though since there's no suitable way to define the derivative of W_t stochastic differential equations are usually expressed in integral or differential form,

Integral form:

$$X_t = X_{t_0} + \int_{t_0}^{t} f(s, X_s)\mathrm{d}s + \int_{t_0}^{t} g(s, X_s)\mathrm{d}W_s \tag{2.3}$$

Differential form:

$$\mathrm{d}X_t = f(t, X_t)\mathrm{d}t + g(t, X_t)\mathrm{d}W_t. \tag{2.4}$$

2.1.1 The Wiener Process

Before stochastic differential equation in (2.3) can make any sense the properties of the driving noise W_t need to be discussed. Usually W_t is taken to be the Wiener process though other alternatives are possible, e.g compound poisson.

Definition 1. *The Wiener process.*
A stochastic process $\{W_t\}_{t>0}$ is said to be a Wiener process if the following conditions are satisfied.
1) $W_0 = 0$ with probability one.
2) For any non-overlapping intervals $[t', t]$ and $[s', s]$, $\quad W_t - W_{t'} \perp W_s - W_{s'}$.
3) For any t, t' such that $t > t' > 0$, $\quad W_t - W_{t'} \sim \mathcal{N}(0, t - t')$.
4) The path is continuous.

It is obvious from the definition that the Wiener process can be simulated on a grid $\{t_n\}_{n=0}^{N}, t_0 = 0$ by setting $w_0 = 0$ and recursively compute w_{t_n} by $w_{t_n} = w_{t_{n-1}} + E_{t_n}$ where $E_{t_n} \sim \mathcal{N}(0, t_n - t_{n-1})$. The results of ten such simulations are shown in Figure 2.1.

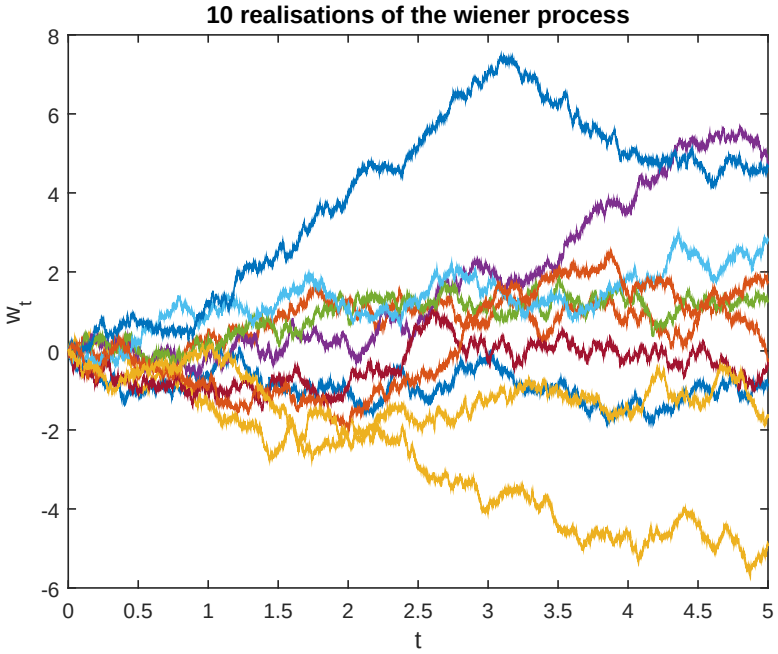


Figure 2.1: Ten simulations of the Wiener process on a uniform grid with $t_N = 5$ and $\Delta t_n = 1/1000$

As the simulation procedure suggests the Wiener process can be thought of as the continuous-time equivalent of the random walk. Since $\mathbb{E}\{W_t^2\} = \mathbb{E}\{(W_t - W_0)^2\} = t < \infty$ and $\lim_{t' \to t} \mathbb{E}\{(W_{t'} - W_t)^2\} = \lim_{t' \to t} |t' - t| = 0$ the Wiener process is continuous almost everywhere in the mean square sense. Though it's nowhere differentiable which motivates why SDEs are thought of in terms of integral equations.

2.1.2 Itō Calculus

The challenging part in making equation (2.3) meaningful is to define what it means to integrate with respect to W_t. Recall that the integral with respect to a real valued variable can be constructed as a Riemann sum,

$$\int_{t_i}^{t_f} f(t)\mathrm{d}t = \lim_{N\to\infty} \sum_{n=0}^{N} f(\tau_n)(t_n - t_{n-1}), \tag{2.5}$$

$$t_i = t_0, \quad t_N = t_f, \quad t_n = t_{n-1} + \frac{t_f - t_i}{N}, \quad \tau_n \in [t_{n-1}, t_n]. \tag{2.6}$$

The idea is to construct the stochastic integral in a similar manner. Though instead of multiplying with the time increments, $t_n - t_{n-1}$, the increments of the Wiener process are used, $W_{t_n} - W_{t_{n-1}}$.

$$\int_{t_i}^{t_f} f(t, X_t)\mathrm{d}W_t = \lim_{N\to\infty} \sum_{n=0}^{N} f(\tau_n, X_{\tau_n})(W_{t_n} - W_{t_{n-1}}), \tag{2.7}$$

$$t_i = t_0, \quad t_N = t_f, \quad t_n = t_{n-1} + \frac{t_f - t_i}{N}, \quad \tau_n \in [t_{n-1}, t_n]. \tag{2.8}$$

In the deterministic integral it doesn't matter where in the interval $[t_{n-1}, t_n]$ the integrand is evaluated. This is however not the case for stochastic integration. In fact when integrating a Wiener process with respect to itself choosing either $\tau_n = t_{n-1}$ or $\tau_n = t_n$ yield two quite different answers, namely $(W_{t_f}^2 - W_{t_i}^2 - (t_f - t_i))/2$ and $(W_{t_f}^2 - W_{t_i}^2 + (t_f - t_i))/2$ respectively. To illustrate this consider the integral of a Wiener process with respect to itself, setting $\tau_n = t_{n-1}$.

$$\int_{t_i}^{t_f} W_s\mathrm{d}W_s = \lim_{N\to\infty} \sum_{n} W_{t_{n-1}}(W_{t_n} - W_{t_{n-1}}) = \tag{2.9}$$

$$\tag{2.10}$$

Expressing the first occurrence of $W_{t_{n-1}}$ as $W_{t_{n-1}} + 0 = W_{t_n} + W_{t_{n-1}} - W_{t_n}$ yields the following,

$$\lim_{N\to\infty} \sum_{n} (W_{t_n} + W_{t_{n-1}} - W_{t_n})(W_{t_n} - W_{t_{n-1}}) = \tag{2.11}$$

$$\lim_{N\to\infty} \sum_{n} (W_{t_n} + W_{t_{n-1}})(W_{t_n} - W_{t_{n-1}}) - \sum_{n} W_{t_n}(W_{t_n} - W_{t_{n-1}}). \tag{2.12}$$

The first in (2.12) is a telescope sum that simply evalutes to $W_{t_f}^2 - W_{t_i}^2$. Now to figure out what the second sum in (2.12) amounts to the same trick is used again, $W_{t_n} + 0 = W_{t_n} - W_{t_{n-1}} + W_{t_{n-1}}$, which results in,

$$-\sum_{n} (W_{t_n} - W_{t_{n-1}} + W_{t_{n-1}})(W_{t_n} - W_{t_{n-1}}) = \tag{2.13}$$

$$-\sum_{n} (W_{t_n} - W_{t_{n-1}})^2 - \sum_{n} W_{t_{n-1}}(W_{t_n} - W_{t_{n-1}}). \tag{2.14}$$

$$\tag{2.15}$$

The second sum in (2.14) equals $-\int_{t_i}^{t_f} W_s\mathrm{d}W_s$ by definition. As a consequence of the properties of the Wiener process the first sum in (2.14) has an expected value of $(t_f - t_i)$, furthermore $\frac{N}{t_f - t_i}(W_{t_n} - W_{t_{n-1}})^2 \sim \mathcal{X}^2(1)$ which implies that the variance of the sum is $\frac{2(t_f - t_i)^2}{N}$, hence it converges to $t_f - t_i$ in mean square sense. It is thus concluded that,

$$\int_{t_i}^{t_f} W_s \mathrm{d}W_s = W_{t_f}^2 - W_{t_i}^2 - (t_f - t_i) - \int_{t_i}^{t_f} W_s \mathrm{d}W_s, \tag{2.16}$$

and hence,

$$\int_{t_i}^{t_f} W_s \mathrm{d}W_s = \frac{W_{t_f}^2 - W_{t_i}^2 - (t_f - t_i)}{2}. \tag{2.17}$$

On the other hand,

$$\lim_{N\to\infty} \sum_n (W_{t_n} + W_{t_{n-1}})(W_{t_n} - W_{t_{n-1}}) = \tag{2.18}$$

$$\lim_{N\to\infty} \sum_n W_{t_n}(W_{t_n} - W_{t_{n-1}}) + \lim_{N\to\infty} \sum_n W_{t_{n-1}}(W_{t_n} - W_{t_{n-1}}) = \tag{2.19}$$

$$W_{t_f}^2 - W_{t_i}^2, \tag{2.20}$$

which implies that

$$\lim_{N\to\infty} \sum W_{t_n}(W_{t_n} - W_{t_{n-1}}) = \frac{W_{t_f}^2 - W_{t_i}^2 + (t_f - t_i)}{2}. \tag{2.21}$$

Clearly the result is different for different choices of τ_n. When the choice is $\tau_n = t_{n-1}$ the result is Itō's integral. Which is the interpretation used for the remainder of this text unless otherwise specified.

The integral in (2.3) can finally be given meaning by interpreting the deterministic integral as a Riemann sum and the stochastic integral as an Itō integral for example. Something peculiar happens to the chain rule when the SDE is interpreted in Itō's sense, this is the Itō Formula given below.

Theorem 1. *The Itō Formula.*
Let $q \in \mathcal{C}^{1,2}(\mathbb{R})$ and X_t be a solution to $\mathrm{d}X_t = f(t, X_t)\mathrm{d}t + g(t, X_t)\mathrm{d}W_t$. Then $Y_t = q(t, X_t)$ satisfies the following SDE.
One dimension:
$\mathrm{d}Y_t = (\frac{\partial q}{\partial t} + f\frac{\partial q}{\partial X_t} + \frac{1}{2}g^2\frac{\partial^2 q}{\partial X_t^2})\mathrm{d}t + g\frac{\partial q}{\partial X_t}\mathrm{d}W_t$
Several Dimensions:
$\mathrm{d}Y_t = (\frac{\partial q}{\partial t} + (\nabla_X q)^T f + \frac{1}{2}\operatorname{Tr}\{g^T\frac{\partial^2 q}{\partial X^2}g\})\mathrm{d}t + (\nabla_X q)^T g\mathrm{d}W_t$

For a motivation why this happens refer to [3]. From Theorem 1 it follows that the state dependence of the diffusion term, $g(t, X_t)$, can in principle be removed by finding a primitive to $\frac{1}{g(t,X_t)}$. This is the Lamperti Transform given below.

Definition 2. *The Lamperti Transform.*
Let X_t be a solution to X_t be a solution to $\mathrm{d}X_t = f(t, X_t)\mathrm{d}t + g(t, X_t)\mathrm{d}W_t$.
Then the Lamperti transform of X_t is given by
$Y_t = q(t, X_t) = \int \frac{1}{g(t,x)}\mathrm{d}x\big|_{x=X_t}$, *resulting in the following SDE for Y_t,*
$\mathrm{d}Y_t = (\frac{\partial q}{\partial t} + \frac{f(t,X_t)}{g(t,X_t)} - \frac{1}{2}\frac{\partial g}{\partial X_t})\mathrm{d}t + \mathrm{d}W_t$

This transform is useful as it's easier to produce approximate transition densities for systems where the diffusion term doesn't depend on the state and it can increase the order of convergence for discrete time approximations of SDEs, see Section 2.1.3.

2.1.3 Discretisation Schemes

Discrete time approximations are needed in order to simulate and develop filters for systems governed by SDEs for the many cases where no closed form solution is known or exists. Typically this is accomplished by stochastic Taylor expansions which serves as the counterpart to the ordinary Taylor expansion. According to Theorem 1 f and g can be expressed as stochastic integrals whose solutions are given by,

$$f_t = f_{t_0} + \int_{t_0}^{t} \frac{\partial f}{\partial s} + f\frac{\partial f}{\partial X_s} + \frac{1}{2}g^2\frac{\partial^2 f}{\partial X_s^2}\mathrm{d}s + \int_{t_0}^{t} g\frac{\partial f}{\partial X_s}\mathrm{d}W_s \tag{2.22}$$

$$g_t = g_{t_0} + \int_{t_0}^{t} \frac{\partial g}{\partial s} + f\frac{\partial g}{\partial X_s} + \frac{1}{2}g^2\frac{\partial^2 g}{\partial X_s^2}\mathrm{d}s + \int_{t_0}^{t} g\frac{\partial g}{\partial X_s}\mathrm{d}W_s. \tag{2.23}$$

This looks a bit messy but is in fact quite useful seeing as the drift and the diffusion are now expressed in terms of a constant initial value with some added integrals. Since integrating with respect to a constant either by Riemann or Itoō is trivial analytically the solution can be divided into a part that can be evaluated exactly and a part containing a bunch of multiple integrals, call the latter R. This results in the following,

$$X_t = X_{t_0} + \int_{t_0}^{t} f_{t_0}\mathrm{d}t + \int_{t_0}^{t} g_{t_0}\mathrm{d}W_s + R \tag{2.24}$$

$$= X_{t_0} + f_{t_0}(t - t_0) + g_{t_0}(W_t - W_{t_0}) + R. \tag{2.25}$$

When R is neglected the result is the Euler-Maruyama scheme (EM scheme). This can be thought of as the SDE counterpart to to Explicit Eulers method from deterministic numerical integration and the complete algorithm is given below.

Algorithm 1. *The Euler-Maruyama Method.*
1) Partition the interval $[0, T]$ *into* N *subintervals equal in length,* $0 = t_0 < t_1 < \cdots < t_N = T$
2) Simulate $X_0 \sim p_{X_0}(x)$
3) For $n = 1, 2, \ldots, N$
$X_{t_n} = X_{t_{n-1}} + f(t_{n-1}, X_{t_{n-1}})\Delta t_{n-1} + g(t_{n-1}, X_{t_{n-1}})E_{t_n}$, *where* $E_{t_n} \sim \mathcal{N}(0, I\Delta t_{n-1})$.

This approximation can be refined by continued application of Itō's Formula to the terms in R. Doing it on the term $g\frac{\partial f}{\partial X}$ from (2.22) results in the Milstein method which is given below.

Algorithm 2. *The Milstein Method.*
1) Partition the interval $[0, T]$ *into* N *subintervals equal in length,* $0 = t_0 < t_1 < \cdots < t_N = T$
2) Simulate $X_0 \sim p_{X_0}(x)$
3) For $n = 1, 2, \ldots, N$
$X_{t_n} = X_{t_{n-1}} + f(t_{n-1}, X_{t_{n-1}})\Delta t_{n-1} + g(t_{n-1}, X_{t_{n-1}})E_{t_n} + \frac{1}{2}g(t_{n-1}, X_{t_{n-1}})\frac{\mathrm{dg}}{\mathrm{dX}}(t_{n-1}, X_{t_{n-1}})(E_{t_n}^2 - \Delta t_{n-1})$,
where $E_{t_n} \sim \mathcal{N}(0, \Delta t_{n-1})$.

In order to assess the quality of an approximation there needs to be a notion of error with respect to the step size, h. If Y_t^h is a discrete time approximation to the stochastic process X_t then there are two interesting notions of error,

$$\varepsilon^{(1)}(h) = \mathbb{E}|X_t - Y_t^h|, \tag{2.26}$$

$$\varepsilon^{(2)}(h) = |\mathbb{E}g(X_t) - \mathbb{E}g(Y_t^h)|, \tag{2.27}$$

where g is chosen from some class of polynomials, C. This leads to two different definitions of convergence corresponding to the errors $\varepsilon^{(1)}$ and $\varepsilon^{(2)}$ respectively.

Definition 3. *convergence.*
A discrete time approximation, Y_t^h, *of* X_t *with maximum step size* h *is said to converge to* X_t *if*
$\lim_{h\to 0}\varepsilon^{(i)}(h) = 0$,
and there exists a positive constant C *that does not depend on* h, *and a* $h_0 > 0$ *such that*
$\varepsilon^{(i)}(h) < Ch^{\alpha}$
for some $\alpha > 0$ *and* $\forall h \in (0, h_0)$.
Y_t^h *is said to converge strongly when* $i = 1$ *and for* $i = 2$ *it is said to converge weakly, with* $\varepsilon^{(i)}$ *being defined in* (2.26) *and* (2.27). *The order of convergence is in both cases given by* α.

As it turns out the Euler-Maruyama scheme attains a strong order of convergence $\alpha = 0.5$ while the Milstein scheme achieves a strong order of convergence $\alpha = 1$. Though whenever $\frac{\partial g}{\partial X} = 0$ these two methods are identical which motivates the transfrom in Definition 2. Both attain weak order $\alpha = 1$. For a more thorough discussion on convergence and discrete time approximations see [3].

Approximations can also be produced by being a bit more clever than mindlessly applying Itōs Formula ad infinitum. For example in [7] Carlos M. Mora develops a collection of exponential schemes for the case when the diffusion term, g, is independent of the state by locally expressing the sought after process, X_t, as

$$X_t = X_{t_0} + \int_{t_0}^{t} A_{t_0} X_t \mathrm{d}t + R_{t,t_0}, \tag{2.28}$$

where R_{t,t_0} is the error process. This procedure results in the Euler Exponential Scheme which achieves weak order 1.

Algorithm 3. *The Euler Exponential Scheme.*
1) Partition the interval $[0,T]$ *into N subintervals equal in length,* $0 = t_0 < t_1 < \cdots < t_N = T$
2) Simulate $X_0 \sim p_{X_0}(x)$
3) For $n = 1, 2, \ldots, N$
$X_{t_n} = \exp\left(\frac{\partial f}{\partial X}(t_{n-1}, X_{t_{n-1}})\Delta t_{n-1}\right)\Big(X_{t_{n-1}} + \Big(f(t_{n-1}, X_{t_{n-1}}) - \frac{\partial f}{\partial X}(t_{n-1}, X_{t_{n-1}})X_{t_{n-1}}\Big)\Delta t_{n-1} + \ldots$
$(\Delta t_{n-1})^{1/2} g(t_{n-1}) E_{t_n}\Big)$,
where E_{t_n} *is distributed according to a symmetric law with variance 1 and moments of any order and* $\frac{\partial f}{\partial X}$ *denotes the jacobian of* f.

Numerical experiments show that the Euler Exponential scheme performs well when Jabobian of f have non-positive eigenvalues with large differences in magnitude and when the eigenvalues are imaginary of large magnitude. The scheme can also be improved upon, leading Mora's Exponential Scheme 4 that achieves weak order 2 by calculating second derivatives of f. Another way to improve the Euler Exponential Scheme is through the application of extrapolation methods [7].

There are of course many other ways to develop approximations, for example [10] uses a method for converting an SDE into a an ODE through a series expansion of the Wiener process with stochastic coefficients.

Chapter 3

State Space Models, Filtering and, Prediction

This chapter is concerned with filtering and prediction in state space models. When a stochastic process is imperfectly measured, e.g noisy or measurements restricted to some subspace, it becomes a partially observed system. From a series of outcomes from the measurement function the goal is to is to infer the state of the underlying SDE as well as predicting future states and measurements. This is flexible way to deal with measurements of a phenomena since it takes cares of a lot of situations, from sensor imperfections to censored observations and perhaps even more complicated relations between the measurements and the state of the underlying SDE arising from e.g the Lamperti Transfrom in Definition 2. Though it's the modelers perogative to assert a measurement function since it's in a very real sense part of the model, a state space model is the aim to model the underlying pheonomena and the measurement of it jointly. More specifically this chapter is concerned with continuous-discrete time filtering, that is the the underlying stochastic process will be governed by an SDE while the measurements are available at a collection of discrete instances. In section 3.1 some general filtering theory is presented as well as the famous Kalman filter which is applicable to linear systems. In section 3.2 Extened Kalman Filters and Unscented Kalman Filters are discussed which are approximate filters for non-linear systems.

3.1 The Filtering Problem

The mathematical setting of continuous-discrete time state space models is that there's a latent process, X_t, governed by a SDE in Itō's sense together with measurements, Y_n, that depend on X_{t_n} and some stochastic variable, V_n, with $V_n \perp V_m$ if $n \neq m$.

$$\mathrm{d}X_t = f(t, X_t)\mathrm{d}t + g(t, X_t)\mathrm{d}W_t, \tag{3.1}$$
$$Y_n = h(t_n, X_{t_n}) + V_n, \quad V_n \sim p_{V_n}(v_n). \tag{3.2}$$

Normally in probabilistic state space models a transition density, $X_t|X_s = x_s \sim p(x_t|x_s), t > s$, and a measurement density, $Y_n|X_{t_n} = x_{t_n} \sim p(y_n|x_{t_n})$, are provided. Here the transition density is implicitly defined by the SDE and the measurement density is defined by h, t, X, and, V. The ambition in the context of filtering such a system is that given a set of observations $Y_{1:n}$ find the so called filtering density of X_{t_n}, that is find the probability density, $p(x_{t_n}|y_{1:n})$. Now Bayes' rule, the fact that $X_{t_n}|X_{t_{n-1}} = x_{t_{n-1}}, Y_{1:n-1} = y_{1:n-1} \sim X_{t_n}|X_{t_{n-1}} = x_{t_{n-1}}$ and $Y_n|X_{t_n} = x_{t_n}, Y_{1:n-1} = y_{1:n-1} \sim Y_n|X_{t_n} = x_{t_n}$ means this density can be expressed in relatively simple terms.

$$p(x_{t_n}|y_{1:n}) = \frac{p(x_{t_n}, y_n|y_{1:n-1})}{p(y_n|y_{1:n-1})} = \frac{p(y_n|x_{t_n}, y_{1:n-1})p(x_{t_n}|y_{1:n-1})}{p(y_n|y_{1:n-1})} \tag{3.3}$$

$$= \frac{p(y_n|x_{t_n})p(x_{t_n}|y_{1:n-1})}{p(y_n|y_{1:n-1})} = \frac{p(y_n|x_{t_n})\int p(x_{t_n}|x_{t_{n-1}}, y_{1:n-1})p(x_{t_{n-1}}|y_{1:n-1})\mathrm{d}x_{t_{n-1}}}{p(y_n|y_{1:n-1})} \tag{3.4}$$

$$= \frac{p(y_n|x_{t_n})\int p(x_{t_n}|x_{t_{n-1}})p(x_{t_{n-1}}|y_{1:n-1})\mathrm{d}x_{t_{n-1}}}{p(y_n|y_{1:n-1})}. \tag{3.5}$$

The above actually contains all information necessary to produce predictions $Y_n|Y_{1:n-k} = y_{1:n-k}$ and $X_n|Y_{n-k} = y_{1:n-k}, k > 0$ as well as the filtering densities provided the aforementioned densities are available and that the integration can be carried out. This is true for Gaussian systems which is the case when the initial distribution $p(x_{t_0})$ is Gaussian and the SDE and measurement are linear in X_t, i.e

$$\mathrm{d}X_t = A(t)x_t\mathrm{d}t + g(t)\mathrm{d}W_t, \tag{3.6}$$

$$Y_n = H(t_n)X_{t_n} + V_n, \quad V_n \sim \mathcal{N}(0, R_{t_n}). \tag{3.7}$$

Only the mean, and covariance needs to be tracked since X_t begins as a Gaussian and stays that way forever. This leads to a pair of ODEs for $\mathbb{E}\{X_t, |X_s\}$ and $\mathbb{C}\{X_t|X_s\}, t > s$ which provides the necessities for the Continuous-Discrete Kalaman Filter given in Algorithm 4. For a more extensive discussion on filtering and prediction in state-space models, though in the time discrete setting, see [8].

Algorithm 4. *The Continuous-Discrete Kalman Filter.*
Prediction:
Solve the following differential equations.
$\frac{\mathrm{d}}{\mathrm{d}t}m_{t|t_{n-1}} = A(t)m_{t|t_{n-1}}$, *and*
$\frac{\mathrm{d}}{\mathrm{d}t}P_{t|t_{n-1}} = A(t)P_{t|t_{n-1}} + P_{t|t_{n-1}}A(t)^T + g(t)g(t)^T$,
on the interval $[t_{n-1}, t_n]$ *with* $m_{t_{n-1}|t_{n-1}} = \mathbb{E}\{X_{t_{n-1}}|y_{t_1:t_{n-1}}\}$ *and* $P_{t_{n-1}|t_{n-1}} = \mathbb{V}\{X_{t_{n-1}}|y_{t_1:t_{n-1}}\}$.
Update:
Compute the following.
$S_{t_n} = H(t_n)P_{t_n|t_{n-1}}H(t_n)^T + R_{t_n}$
$K_{t_n} = P_{t_n|t_{n-1}}H(t_n)^T S_{t_n}^{-1}$
$\mathbb{E}\{X_{t_n}|y_{t_1:t_n}\} = x_{t_n|t_{n-1}} + K_{t_n}(y_{t_n} - H(t_n)x_{t_n|t_{n-1}})$
$\mathbb{V}\{X_{t_n}|y_{t_1:t_n}\} = (I - K_{t_n}H(t_n))P_{t_n|t_{n-1}}$

3.2 Approximate Filters

The world is rarely as simple as one could hope for, it's frequently non-linear and/or non-Gaussian. When the system is non-linear the procedure in Algorithm 4 no longer applies. Though since it only operates on means and covariances a popular approach to approximate filtering is based on approximating these and then applying the kalman filter update anyway. This section will discuss two such approaches, the first one being linearisation of the drift function, $f(t, X_t)$, and the measurement function, $h(t, X_t)$, around the mean of X_t resulting in the Extended Kalman Filter (EKF) which is one of the earlier attempts at filtering non-linear systems. The second approach is based on the Unscented Transform (UT) which involves deterministically sampling points from a distribution and to each point associating a weight such that means and covariances can be computed through a weighted sum, this results in the Unscented Kalman Filter (UKF). Other approaches involve for example particle methods though that is outside the scope of this text.

3.2.1 The Extended Kalman Filter

As mentioned the basic idea of the Extended Kalman Filter is simply to produce a linear approximation to the system and apply the ordinary Kalman Filter procedure to the resulting system,

$$\mathrm{d}X_t \approx \left(f(t, \mathbb{E}X_t) + \frac{\partial f}{\partial X_t}(t, \mathbb{E}X_t)(X_t - \mathbb{E}X_t) \right) \mathrm{d}t + g(t)\mathrm{d}W_t, \tag{3.8}$$

$$Y_n \approx h(t_n, \mathbb{E}X_{t_n}) + \frac{\partial h}{\partial X_{t_n}}(t_n, \mathbb{E}X_{t_n})(X_{t_n} - \mathbb{E}X_{t_n}) + V_n. \tag{3.9}$$

When the necessary expectations have been computed the result is the Continuous-Discrete Extended Kalman Filter.

Algorithm 5. *The Continuous-Discrete Extended Kalman Filter.*
Prediction:
Solve the following differential equations.
$\frac{\mathrm{d}}{\mathrm{d}t} m_{t|t_{n-1}} = f(t, m_{t|t_{n-1}})$, *and*
$\frac{\mathrm{d}}{\mathrm{d}t} P_{t|t_{n-1}} = \frac{\partial f}{\partial X_t}(t, m_{t|t_{n-1}}) P_{t|t_{n-1}} + P_{t|t_{n-1}} \frac{\partial f}{\partial X_t}(t, m_{t|t_{n-1}})^T + g(t)g(t)^T$,
on the interval $[t_{n-1}, t_n]$ *with* $m_{t_{n-1}|t_{n-1}} = \mathbb{E}\{X_{t_{n-1}} | y_{t_1:t_{n-1}}\}$ *and* $P_{t_{n-1}|t_{n-1}} = \mathbb{V}\{X_{t_{n-1}} | y_{t_1:t_{n-1}}\}$.
Update:
Compute the following.
$S_{t_n} = \frac{\partial h}{\partial X_t}(t_n, m_{t_n|t_{n-1}}) P_{t_n|t_{n-1}} \frac{\partial h}{\partial X_t}(t_n, m_{t_n|t_{n-1}})^T + R_{t_n}$
$K_{t_n} = P_{t_n|t_{n-1}} \frac{\partial h}{\partial X_t}(t_n, m_{t_n|t_{n-1}})^T S_{t_n}^{-1}$
$\mathbb{E}\{X_{t_n} | y_{t_1:t_n}\} = m_{t_n|t_{n-1}} + K_{t_n}(y_{t_n} - h(t_n, m_{t_n|t_{n-1}}))$
$\mathbb{V}\{X_{t_n} | y_{t_1:t_n}\} = (I - K_{t_n} \frac{\partial h}{\partial X_t}(t_n, m_{t_n|t_{n-1}})) P_{t_n|t_{n-1}}$

The approximation can of course be refined by keeping higher order terms, when the second derivatives of the Taylor expansions are kept the result is the second order Extended Kalman Filter, to see what happens in the case of a discrete time system refer to [8].

3.2.2 The Unscented Kalman Filter

Another approach to approximate filtering in non-linear systems is based on the Unscented Transform which is inspired by the realisation that it is easier to approximate a probability distribution than an arbitrary non-linear function. The idea is to deterministically sample a set of so-called Sigma Points such that the true mean and covariance are exactly recovered by a weighted sum. This approach, at least in discrete time, guarantees the same performance as a truncated second order filter but with the same computational complexity as the Extended Kalman Filter and without the need to compute any derivatives [4].

Algorithm 6. *The Unscented Transform.*
Let $\mathbb{R}^D \ni X \sim \mathcal{N}(m, P)$ *and* $q(x)$ *be a function that can be evaluated at* X *then* $\mathbb{E}q(X)$ *and* $\mathbb{V}q(X)$ *can be approximated by the following procedure.*
Set the parameters of the algorithm, α, β, κ, *and define the following vectors, called sigma points.*
$\mathcal{X}^{(0)} = m$
$\mathcal{X}^{(d)} = m + (D + \lambda)^{1/2} P^{1/2}_{:,d}$,
$\mathcal{X}^{(d+D)} = m - (D + \lambda)^{1/2} P^{1/2}_{:,d}, \quad d = 1, \ldots, D$
Then $\mathbb{E}q(X)$, $\mathbb{V}q(X)$ *are approximated by the following.*
$\mathbb{E}_{UT} q(X) = \sum_{d=0}^{2D} w_d^{(m)} q(\mathcal{X}^{(d)})$
$\mathbb{V}_{UT} q(X) = \sum_{d=0}^{2D} w_d^{(c)} (q(\mathcal{X}^{(d)}) - \mathbb{E}q(X))(q(\mathcal{X}^{(d)}) - \mathbb{E}q(X))^T$,
where $\lambda = \alpha^2(D + \kappa) - D$ *and the weights are defined by the following.*
$w_0^{(m)} = \frac{\lambda}{\lambda + D}$,
$w_0^{(c)} = \frac{\lambda}{\lambda + D} + (1 - \alpha^2 + \beta)$
$w_d^{(m)} = w_d^{(c)} = \frac{1}{2(D+\lambda)}, \quad d = 1, \ldots, 2D.$

In order to derive a Kalman-type filter for continuous-discrete systems based on the Unscented Transform there are several approaches. The UT can be applied directly to a discretisation of the SDE resulting in Algorithm 8. When taking the limit $\Delta t \to 0$ an ODE for the sigma points can be derived as is done using the EM scheme in [9], this results in Algorithm 7. Another approach is to approximate the SDE by an

ODE with random coefficients and then jointly apply the UT to the state and the random coefficients [10].

Algorithm 7. *The Continuous-Discrete Unscented Kalman Filter.*
Prediction:
Solve the following differential equations.
$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}\{X_t|y_{t_1:t_{n-1}}\} = \mathbb{E}_{UT}\{f(t, X_t)|y_{t_1:t_{n-1}}\}$,
$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{V}\{X_t|y_{t_1:t_{n-1}}\} = \mathbb{C}_{UT}\{X_t, f(t, X_t)|y_{t_1:t_{n-1}}\} + \mathbb{C}_{UT}\{f(t, X_t), X_t|y_{t_1:t_{n-1}}\} + g(t)g(t)^T$,
on the interval $[t_{n-1}, t_n]$.
Update:
Compute the following.
$K_{t_n} = \mathbb{C}\{X_{t_n}, Y_{t_n}|y_{t_1:t_{n-1}}\}\mathbb{V}\{Y_{t_n}|y_{t_1:t_{n-1}}\}^{-1}$,
$\mathbb{E}\{X_{t_n}|y_{t_1:t_n}\} = \mathbb{E}\{X_{t_n}|y_{t_1:t_{n-1}}\} + K_{t_n}(y_{t_n} - \mathbb{E}\{Y_{t_n}|y_{t_1:t_{n-1}}\})$,
$\mathbb{V}\{X_{t_n}|y_{t_1:t_n}\} = \mathbb{V}\{X_{t_n}|y_{t_1:t_{n-1}}\} - K_{t_n}\mathbb{V}\{h(t_n, X_{t_n})|y_{t_1:t_{n-1}}\}K_{t_n}^T$.
Where the expectation, variances and, covariances are computed with respect to the unscented transform.

Now the ODE in Algorithm 7 may need some additional clarification. Let $\mathcal{S} : \mathbb{R}^D \times \mathbb{R}^{D\times D} \to \mathbb{R}^{D\times(2D+1)}$ denote the function that transforms the mean, m, and covariance P into a matrix with the sigma points in the columns, i.e $(\mathcal{S}(m, P))_{:,d} = \mathcal{X}^{(d)}$. Additionally let $m_t = \mathbb{E}\{X_t|y_{t_1:t_{n-1}}\}$, $f_t = \sum_{d=0}^{2D} w_d^{(m)} f(t, (\mathcal{S}(m_t, P_t))_{:,d})$ and, $P_t = \mathbb{V}\{X_t|y_{t_1:t_{n-1}}\}$ to reduce clutter. The ODE is then given by,

$$\frac{\mathrm{d}}{\mathrm{d}t}m_t = \sum_{d=0}^{2D} w_d^{(m)} f(t, (\mathcal{S}(m_t, P_t))_{:,d}), \tag{3.10}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}P_t = \sum_{d=0}^{2D} w_d^{(c)}((\mathcal{S}(m_t, P_t))_{:,d} - m_t)(f(t, (\mathcal{S}(m_t, P_t))_{:,d}) - f_t)^T \tag{3.11}$$

$$+ \sum_{d=0}^{2D} w_d^{(c)}(f(t, (\mathcal{S}(m_t, P_t))_{:,d}) - f_t)((\mathcal{S}(m_t, P_t))_{:,d} - m_t)^T) + g(t)g(t)^T, \tag{3.12}$$

with the derivatives of m_t and P_t now expressed solely in terms of functions with m_t and P_t as arguments it is now straight forward to apply a suitable numerical scheme for ODEs.

Algorithm 8. *The Discretised Unscented Kalman Filter.*
Choose your favourite explicit SDE discretisation scheme, $X_{t+h} = F(X_t, E_{t+h}, h)$*, where* h *is the step size.*
Predict:
Set $s_{n-1} = t_{n-1}$ *and perform the unscented transform on the augmented vector,* $Z_{s_{n-1}} = \begin{pmatrix} X_{s_{n-1}}^T & E_{s_{n-1}+h}^T \end{pmatrix}^T$*, and compute*
$\mathbb{E}\{X_{s_{n-1}+h}|y_{t_1:t_{n-1}}\} = \mathbb{E}_{UT}\{F(Z_{s_{n-1}}^X, Z_{s_{n-1}}^E, h)|y_{t_1:t_{n-1}}\}$,
$\mathbb{V}\{X_{s_{n-1}+h}|y_{t_1:t_{n-1}}\} = \mathbb{V}_{UT}\{F(Z_{s_{n-1}}^X, Z_{s_{n-1}}^E, h)|y_{t_1:t_{n-1}}\}$,
if $s_{n-1} + h = t_n$ *continue to the update step, otherwise repeat the above computation for* $s_n = s_{n-1} + h$*. The expectations are taken with respect to the unscented transform and* Z_t^X, Z_t^E *are the sub-vectors of* Z_t *corresponding to* X_t *and* E_{t+h} *respectively.*
Update:
Use the update procedure from Algorithm 7.

In principle a semi-implicit or implicit scheme can be used in Algorithm 8 though the author makes no guarantees with regards to theory nor experience that this would be sensible in practice.

Chapter 4

Parameter Estimation

This chapter deals with parameter estimation in state space models driven by a SDE. Section 4.1 features a very brief discussion on estimators in general and what should be expected from a good estimator. In Section 4.2 the maximum likelihood estimator is presented along with it's ugly stepsister, the quasi maximum likelihood estimator. Section 4.3 ponders different strategies for maximising the likelihood when a closed form solution is unavailable with particular attention given to the Gradient-Based Adaptive Stochastic Search algorithm.

4.1 A very brief discussion on estimators in general

This chapter discusses parameter estimation in state space models driven by a SDE. Hence the assumption is that the system under consideration is described by the following model,

$$\mathrm{d}X_t = f(t, X_t; \theta)\mathrm{d}t + g(t; \theta)\mathrm{d}W_t, \tag{4.1}$$

$$Y_n = h(t_n, X_{t_n}; \theta) + V_n, \quad V_n \sim \mathcal{N}(0, , R_{t_n}(\theta)), \tag{4.2}$$

where f, g, h and, R_{t_n} all depend on the parameter θ, taken from some parameter space, $\theta \in \Theta$. The goal is to given a set of measurements, $y_{1:N}$, find the parameter θ^* that in some sense is the best fit to the measurements of all possible choices of parameters from Θ. This parameter is obviously a function of the measurements, i.e $\theta^* = \theta^*(Y_{1:N})$. There are a few conditions put on θ^* to guarantee that it's a reasonable estimator of the true parameter, let's call it θ',

$$(1) \quad \lim_{N\to\infty} \mathbb{E}\theta^* = \theta', \tag{4.3}$$

$$(2) \quad \lim_{N\to\infty} \mathbb{V}\theta^* = 0. \tag{4.4}$$

An estimator for which the first condition holds is called asymptotically unbiased and an estimator for which the first and the second condition holds is called asymptotically consistent. Together they ensure that when there's an unlimited amount of data available the true value, θ', is recovered. It's also preferable that θ^* is efficient which means that among all unbiased estimators θ^* has the smallest variance [6].

4.2 Maximum Likelihood

Suppose the joint density of $Y_{1:N}$ is available then the likelihood function is given by,

$$L(\theta, Y_{1:N}) = p(y_{1:N}; \theta) = \left(\prod_{n=2}^{N} p(y_n | y_{1:n-1}; \theta) \right) p(y_1; \theta). \tag{4.5}$$

Though it's often more convenient to work with the log-likelihood function,

$$\ell(\theta, Y_{1:N}) = \log L(\theta) = \log p(y_1; \theta) + \sum_{n=2}^{N} \log p(y_n | y_{1:n-1}; \theta). \tag{4.6}$$

Maximising the log-likelihood function results in the Maximum Likelihood Estimator (MLE),

$$\theta_{MLE} = \underset{\theta \in \Theta}{\operatorname{argmax}}\ \ell(\theta, Y_{1:N}). \tag{4.7}$$

The MLE has some nice properties when the data set grows,

$$I_N(\theta')^{1/2}(\theta_{MLE} - \theta') \xrightarrow{d} \mathcal{N}(0, I),\ N \to \infty, \tag{4.8}$$

where $I_N(\theta)$ is an approximation to the fisher information matrix given by

$$I_N(\theta) = \sum_{n=0}^{N} \mathbb{E}\Big\{ \Big(\nabla_\theta \ell(\theta, y_{1:n}) - \nabla_\theta \ell(\theta, y_{1:n-1})\Big)\Big(\nabla_\theta \ell(\theta, y_{1:n}) - \nabla_\theta \ell(\theta, y_{1:n-1})\Big)^T \Big| y_{1:n-1} \Big\}, \tag{4.9}$$

where the first term in the sum is defined as $\mathbb{E}\big\{\nabla_\theta \ell(\theta, y_1) \nabla_\theta \ell(\theta, y_1)^T\big\}$. For handwavy argument as to why this is the case see [6].

In the case of state space models driven by a SDE the problem arises that it's difficult or impossible to arrive at a closed form expression for the likelihood function. In linear models it can be approximated by numerical integration using the Kalman Filter in Algorithm 4. However the non-linear case requires the delusion that the one-step prediction errors are normally distributed, i.e

$$\mathbb{E}\{Y_n | y_{1:n-1}\} - Y_n \sim \mathcal{N}(0, S_{t_n}), \tag{4.10}$$

where the prediction mean and covariance are computed using for example Algorithm 5, 7, or 8. The result of maximising these quasi-likelihoods does not necessarily have the good properties of the MLE, but it's typically consistent, see Chapter 5 in [2].

4.3 Optimisation

The conclusion of the previous section is that many times in order to produce an estimator a maximisation/minimisation problem needs to be solved and when this is done on the likelihood function the result is the MLE but it can also be done on a quasi-likelihood which results in the Quasi Maximum Likelihood Estimator (QMLE).

The classical methods for finding the MLE (or the maximum of any function) are the gradient methods, i.e gradient ascent or Newtons method. Though these kinds of algorithms are usually not suitable for parameter estimation in continuous-time state space models since the derivatives are too difficult to find. The options are to either approximate the derivatives numerically or to use a derivative free method. A recent method belonging to the latter category is the Gradient-Based Adaptive Stochastic Search algorithm that was developed in [1].

4.3.1 Gradient-Based Stochastic Search

The idea behing the Gradient-Based Adaptive Stochastic Search algorithm is to use a parametrised family of probability distributions, $p(\theta; \gamma), \gamma \in \Gamma$, to search for θ_{MLE} through sampling. The samples are then used in updating γ so that the next sampling step is more successful in finding promising candidates for θ_{MLE}. The method was originally not developed for finding the MLE and so there is an assumption that $\ell(\theta, y_{1:N})$ is bounded, i.e $\ell(\theta, y_{1:N}) \in [\ell_{lb}, \ell_{ub}]$, which is not necessarily true for the log-likelihood. Though in practice this is of little consequence since the optimisation can be performed on some subset of Θ, $\hat{\Theta} \subset \Theta$ where this assumption holds. Now let $\hat{\ell} = \ell(\hat{\theta}, y_{1:N})$ be the maximum value of $\ell(\theta, y_{1:N})$ with $\theta \in \hat{\Theta}$. The problem of maximising $\ell(\theta, y_{1:N})$ is then transformed into the following problem,

$$\gamma^* = \underset{\gamma\in\Gamma}{\operatorname{argmax}} \int_{\theta\in\hat{\Theta}} \ell(\theta, y_{1:N}) p(\theta;\gamma)\mathrm{d}\theta \tag{4.11}$$

$$= \underset{\gamma\in\Gamma}{\operatorname{argmax}}\ \mathbb{E}_{p(\theta;\gamma)}\{\ell(\theta, y_{1:N})\}. \tag{4.12}$$

Though in order to develop this further $\ell(\theta, y_{1:N})$ needs to be transformed by a shape function, $S_\gamma(\theta)$, that is bounded and makes sure the maximisation objective is positive as it will be used to define a probability density. Obviously $S_\gamma(\theta)$ also needs to be non-decreasing so as to ensure the maxima does not change. The shape function, $S_\gamma(\theta)$, can also be used to prune some of the least promising samples by choosing it as a (soft) indicator function and setting the threshold to some quantile of $\ell(\theta, y_{1:N})$. Though this quantile will have to be estimated by the samples of $p(\theta;\gamma)$ in which case the shape function will be referred to as $\hat{S}$ if it's of technical relevancy. the Now the following function can be defined,

$$L(\gamma, \gamma') = \int_{\theta\in\hat{\Theta}} S_{\gamma'}(\ell(\theta, y_{1:N})) p(\theta;\gamma)\mathrm{d}\theta. \tag{4.13}$$

Finally a Newton-type procedure can be carried out by finding an expression for the gradient and the hessian of $L(\gamma, \gamma_i)$ which can then be estimated from the samples of $p(\theta;\gamma_i)$. This becomes especially simple when p is chosen as an exponential distribution, $p(\theta;\gamma) = \exp(\gamma^T T(\theta) - \phi(\gamma))$, where $T(\theta)$ is the vector of sufficient statistics. Subsequently the following probability distribution is defined,

$$p'(\theta;\gamma) \propto S_{\gamma'}(\ell(\theta, y_{1:N})) p(\theta;\gamma), \tag{4.14}$$

which results in Algorithm 9.

Algorithm 9. *Gradient-Based Adaptive Stochastic Search.*
Choose an exponential density $p(\theta;\gamma)$, a family of step sizes $\{\alpha_i\}$, a sequence of sample sizes $\{N_i\}$, a tolerance δ, a small constant ε, an initial parameter γ_0 and, a maximum number of iterations I.
Sampling:
Generate N_i samples from $p(\theta;\gamma_i)$,
Update:
$\gamma_{i+1} = \gamma_i + \alpha_i V_i^{-1}(\mathbb{E}'_i\{T(\theta)\} - \mathbb{E}_i\{T(\theta)\})$.
If $||\mathbb{E}'_i\{T(\theta)\} - \mathbb{E}_i\{T(\theta)\}|| < \delta$ stop, otherwise increment i and go back to the sampling step.
V_i is given by $V_i = \mathbb{V}T(\theta) + \varepsilon I$. $\mathbb{E}_i$ and $\mathbb{E}'_i$ are the expectations with respect to $p(\theta;\gamma_i)$ and $p'(\theta;\gamma_i)$ respectively, the latter can be evaluated through importance weights.

Obviously Algorithm 9 will not converge for any set of parameters fed into it though it is proven in [1] that it does indeed converge if the following conditions are satisfied.

$$(1)\ \alpha_i > 0\ \forall i, \quad \alpha_i \to 0 \text{ as } i \to \infty, \quad \sum_{i=0}^{\infty} \alpha_i = \infty. \tag{4.15}$$

$$(2)\ N_i = N_0 i^\zeta \text{ for some } \zeta > 0, \quad \frac{\alpha_i}{N_i} \in O(i^{-\beta}) \text{ for some } \beta > 0 \tag{4.16}$$

$$(3)\ T(\theta) \text{ is bounded in } \hat{\Theta}. \tag{4.17}$$

$$(4)\ \forall\theta,\ |\hat{S_\gamma} - S_\gamma| \to 0 \text{ with probability 1 as } N_i \to \infty. \tag{4.18}$$

Furthermore, the convergence rate of Algorithm 9 can be increased by performing Polyak averaging according to,

$$\gamma_{i+1} = \gamma_i + \alpha_i V_i^{-1}(\mathbb{E}'_i\{T(\theta)\} - \mathbb{E}_i\{T(\theta)\}) + \alpha_i c(\bar{\gamma_i} - \gamma_i), \tag{4.19}$$

$$\bar{\gamma_i} = \frac{i-1}{i}\bar{\gamma}_{i-1} + \frac{1}{i}\gamma_i, \quad \bar{\gamma}_0 = 0. \tag{4.20}$$

Chapter 5

Results

This chapter presents the results of a series of simulation studies used to evaluate the approximate filtering methods of section 3.2 as well as the optimisation method for finding the (Quasi) MLE of section 4.3. In section 5.1 the continuous-time UKF, the discretised UKF based on the Euler-Exponential scheme and the EKF are evaluated on stochastic versions of the Lotka-Volterra-, Lorenz63- and, Lorenz96 systems. In section 5.3 the Gradient-Based Adaptive Stochastic Search algorithm is evaluated on the Ornstein-Uhlenbeck process and the Lorenz63 system.

5.1 Filter Performance Evaluation

The performance measures used to assess the relative quality of the approximate filters is in this text based on the one-step prediction error, $y_{n|n-1} - y_n$, and the filtering error, $x_{n|n} - x_n$, according to the following,

$$\varepsilon_y = \left(\frac{1}{N} \sum_{n=1}^{N} ||y_n - y_{n|n-1}||^2 \right)^{1/2}, \tag{5.1}$$

$$\varepsilon_x = \left(\frac{1}{N} \sum_{n=1}^{N} ||x_n - x_{n|n}||^2 \right)^{1/2}. \tag{5.2}$$

In the case of one-dimensional signals this corresponds to the root mean square error (RMSE).

5.1.1 The Lotka-Volterra System

The Lotka-Volterra system is a model intended to capture the behaviour of systems in population ecology where different species are either classified as prey or predator. The population size of the predator species ought to increase when there's an abundance of food available and decrease when there's food scarcity, i.e it's related to the population size of the prey species. On the other hand the population size of the prey species ought to decrease with the population size of the predator species as it increases their survival rate. A stochastic version of the system is given below,

$$\mathrm{d}X_t = \begin{pmatrix} (b_1 - a_1 X_t^{(2)}) X_t^{(1)} \\ (b_2 - a_2 X_t^{(1)}) X_t^{(2)} \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 X_t^{(1)} & 0 \\ 0 & \sigma_2 X_t^{(2)} \end{pmatrix} \mathrm{d}W_t \tag{5.3}$$

$$Y_{t_n} = X_{t_n} + V_n, \quad V_n \sim \mathcal{N}(0, R) \tag{5.4}$$

where $h(t, X_t) = X_t$ is the function through which the system is observed and E_{t_n} is a measurement error. Though one can also envision just measuring the prey species or the predator species corresponding to $h(t, X) = \begin{pmatrix} 1 & 0 \end{pmatrix} X$ and $h(t, X) = \begin{pmatrix} 0 & 1 \end{pmatrix} X$ respectively.

In order to apply the filtering algorithms of section 3.2 the Lamperti Transform needs to be applied to remove the state dependent diffusion. Choosing $q^{(1)}(t, X^{(1)}, X^{(2)}) = \sigma_1^{-1} \log(X^{(1)})$ and choosing $q^{(2)}(t, X^{(1)}, X^{(2)}) = \sigma_2^{-1} \log(X^{(2)})$ yields the following system,

$$\mathrm{d}Z_t = \begin{pmatrix} \frac{b_1 - a_1 \exp(\sigma_2 Z_t^{(2)})}{\sigma_1} - \frac{\sigma_1}{2} \\ \frac{b_2 - a_2 \exp(\sigma_1 Z_t^{(1)}))}{\sigma_2} - \frac{\sigma_2}{2} \end{pmatrix} \mathrm{d}t + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \mathrm{d}W_t \tag{5.5}$$

$$Y_{t_n} = \begin{pmatrix} \exp(\sigma_1 Z_{t_n}^{(1)}) \\ \exp(\sigma_2 Z_{t_n}^{(2)}) \end{pmatrix} + V_n, \quad V_n \sim \mathcal{N}(0, R). \tag{5.6}$$

The system given by this form is simulated 1000 times on the interval $[0, 3.5]$ using the Euler-Maruyama scheme and a step size of $\Delta t = 1/10000$ with parameters the following parameters

$$a_1 = b_1 = 3, \quad a_2 = b_2 = -15, \tag{5.7}$$

$$\sigma_1 = 0.25, \quad \sigma_2 = 0.20, \tag{5.8}$$

$$R = \begin{pmatrix} 0.1^2 & 0 \\ 0 & 0.1^2/2 \end{pmatrix}. \tag{5.9}$$

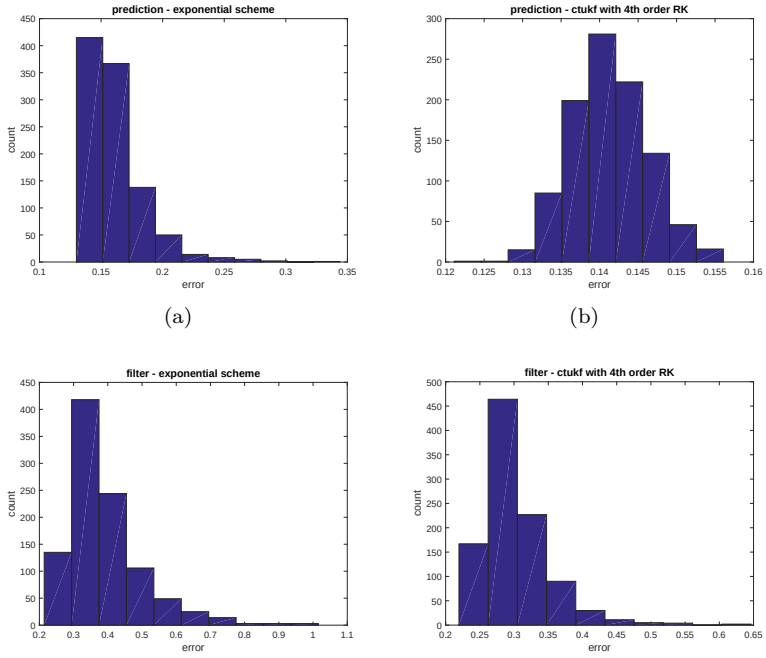
The data size was also reduced by throwing away all measurements between every $100th$ measurement. The UKF based on the exponential scheme was tested using $\Delta t_1 = 1/100$ while the continuous-time UKF was tested using $\Delta t_2 = 1/1000$ and integrated using a fourth order Runge-Kutta scheme (RK4) and their parameters were set to $\alpha = 10^{-3},\ \beta = 2,\ \kappa = 0$. The step size for the continous-time was chosen as a lower to be smaller since the covariance had a tendency to stop being positive definite otherwise. The Extended Kalman Filter (integrated with a RK4 scheme) had a tendency to either crash or be on it's way toward crashing under these circumstances so due to the authors' lack of patience it was omitted from the comparison. This is consistent with previous experience of the EKF, see for example [6].

As Figure 5.1 illustrates the both the continuous-time UKF and the exponential UKF appear to have rather similar performance in both the filtering and the prediction tasks. Though the exponential UKF has slightly worse performance on average, see Table 5.1, and it deviates from the mean performance with higher frequency. Keep in mind that the step size was 10 times larger for the exponential UKF.

Table 5.1: The mean of the performance measures, ε_y and ε_x, taken over the 1000 simulations of the Lotka-Volterra system.

	Continuous-Time UKF	Exponential UKF
ε_y (mean)	0.1413	0.1615
ε_x (mean)	0.3016	0.3921

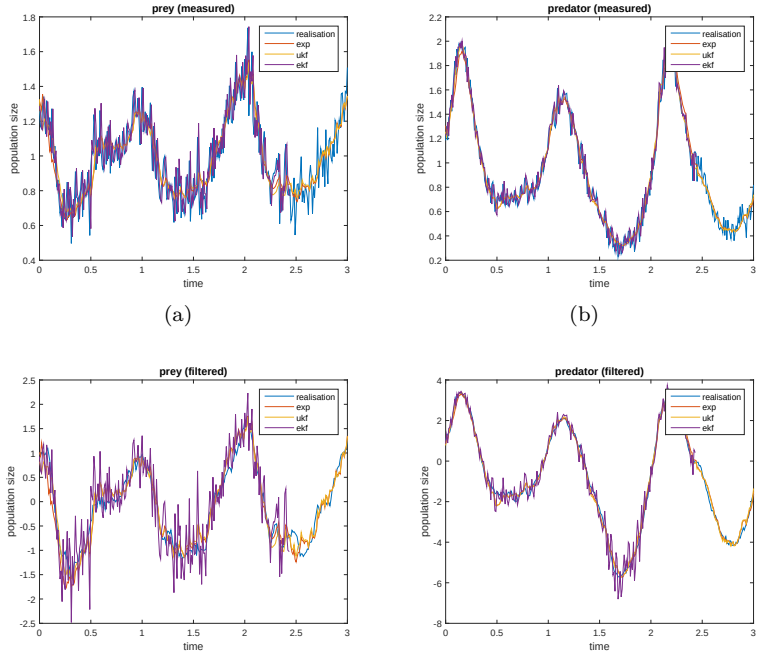
Furthermore, to illustrate the systems behaviour graphically a realisation of X_t and Y_n are plotted along with the filter estimate and the one-step prediction respectively. This time the EKF is included and it is integrated using a RK4 scheme with a step size of 1/1000 (the same as for the continuous-time UKF). From figure 5.2 it is clear that the EKF has trouble keeping up - it oscillates wildly around the true value until the covariance matrix goes singular/non-positive definite and at around $t = 2.5$ when it starts outputting `NaN`. The two UT filters behave reasonably though but it can be hard to see since the EKF output is blocking the view.



(a) (b)

(c) (d)

Figure 5.1: Histogram of the square root mean norm errors for the Lotka-Volterra system
a) one-step predictions of the exponential scheme b) one-step predictions of the continuous-discrete UKF
c) filter estimates of the exponential scheme d) filter estimates of the continuous-discrete UKF.



(a) (b)

(c) (d)

Figure 5.2: The continuous-time UKF, exponential UKF and the EKF compared against the true value for a) prediction of $Y_n^{(1)}$ b) prediction of $Y_n^{(2)}$ c) filter estimate of $X_{t_n}^{(1)}$ d) filter estimate of $X_{t_n}^{(2)}$.

5.1.2 The Lorenz63 System

The Lorenz 63 system is a simplified model for atmospheric convection proposed by Edward N. Lorenz in [5]. This is one of the earlier and certainly most famous examples of a system that may exhibit chaotic behaviour. A stochastic version with uncorrelated Brownian perturbations is given below.

$$\mathrm{d}X_t = \begin{pmatrix} \sigma(X_t^{(2)} - X_t^{(1)}) \\ X_t^{(1)}(\rho - X_t^{(3)}) - X_t^{(2)} \\ X_t^{(1)} X_t^{(2)} - \beta X_t^{(3)} \end{pmatrix} \mathrm{d}t + \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \mathrm{d}W_t, \tag{5.10}$$

$$Y_{t_n} = h(t, X_t) + V_n, \quad V_n \sim \mathcal{N}(0, R) \tag{5.11}$$

When the parameters are chosen as $(\sigma, \beta, \rho) = (10, 8/3, 28)$ the system becomes chaotic. The system given by this form is simulated 1000 times on the interval $[0, 3.5]$ using the Euler-Maruyama scheme and a step size of $\Delta t = 1/10000$ with parameters the following parameters,

$$\sigma = 10, \quad \beta = 8/3, \quad \rho = 28, \tag{5.12}$$

$$\sigma_1 = \sigma_2 = \sigma_3 = 4.5, \tag{5.13}$$

$$h(t, X_t) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} X_t, \tag{5.14}$$

$$R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{5.15}$$

The data size was again reduced by keeping every $100th$ measurement. The UKF based on the exponential scheme was tested using $\Delta t_1 = 1/100$ while the continuous-time UKF was tested using $\Delta t_2 = 2/10000$ and integrated using a fourth order Runge-Kutta scheme (RK4) and their parameters were set to $\alpha = 10^{-3},\ \beta = 2,\ \kappa = 0$. The EKF was once again excluded from the comparison.

Figure 5.3 reveal that the continuous-time UKF and the exponential UKF have rather similar performance again in both prediction and filtering though this time the exponential UKF performs slightly better on average, see Table 5.2. It's important to note that this time the step size of the exponential UKF is 50 times bigger than that of the continuous-time UKF.

Table 5.2: The mean of the performance measures, ε_y and ε_x, taken over the 1000 simulations of the Lorenz63 system.

	Continuous-Time UKF	Exponential UKF
ε_y (mean)	1.7657	1.7648
ε_x (mean)	1.5652	1.5626

A graphic demonstration of the filer performance can be found in Figure 5.4. Though since the exponential UKF and the continuous-time UKF are so similar in their estimates and predictions the former is hard to see unless this thesis is read in PDF-format on a computer where the reader can zoom in, a lot.

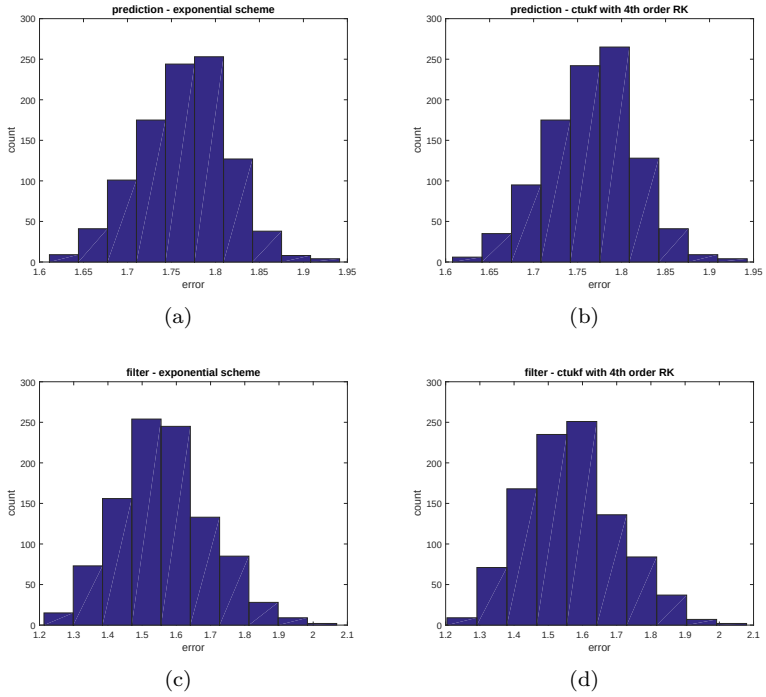


Figure 5.3: Histogram of the square root mean norm errors for a) one-step predictions of the exponential scheme b) one-step predictions of the continuous-discrete UKF c) filter estimates of the exponential scheme d) filter estimates of the continuous-discrete UKF.

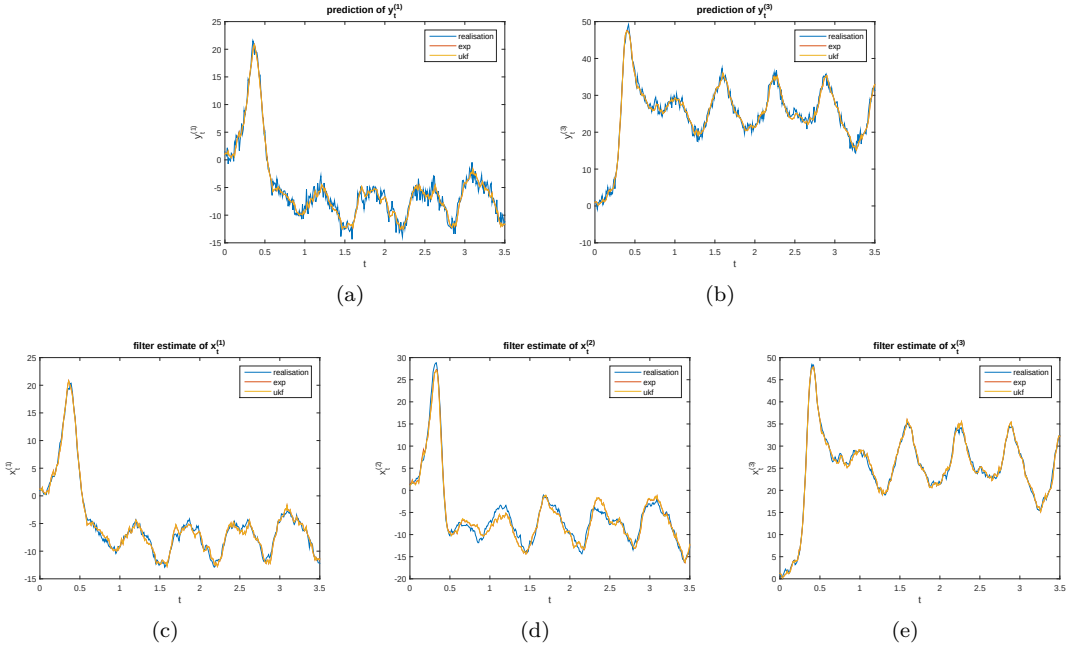


Figure 5.4: The continuous-time UKF, exponential UKF and compared against the true value for a) prediction of $Y_n^{(1)}$ b) prediction of $Y_n^{(3)}$ c) filter estimate of $X_{t_n}^{(1)}$ d) filter estimate of $X_{t_n}^{(2)}$ e) filter estimate of $X_{t_n}^{(3)}$.

5.1.3 The Lorenz96 System

The Lorenz 96 system is a model proposed by Lorenz in 1996 that is intended to describe the dynamics of some atmospheric variable over a single latitude circle. It is thus a spatiotemporal model, continuous in time and discrete in space, given by,

$$\mathrm{d}X_t^{(m)} = (-X_t^{(m-2)}X_t^{(m-1)} + X_t^{(m-1)}X_t^{(m+1)} - X_t^{(m)} + F)\mathrm{d}t, \tag{5.16}$$

$$m = 1, \ldots, M, \quad X_t^{(-2)} = X_t^{(M-1)}, X_t^{(-1)} = X_t^{(M)}, X_t^{(M+1)} = X_t^{(1)}. \tag{5.17}$$

A simple stochastic version is given by,

$$\mathrm{d}X_t^{(m)} = -X_t^{(m-2)}X_t^{(m-1)} + X_t^{(m-1)}X_t^{(m+1)} - X_t^{(m)} + F + \sigma \mathrm{d}W_t^{(m)}, \tag{5.18}$$

$$Y_n = X_{t_n} + V_n \quad V_n \sim \mathcal{N}(0, I\sigma_V^2), \tag{5.19}$$

$$m = 1, \ldots, M, \quad X_t^{(-2)} = X_t^{(M-1)}, X_t^{(-1)} = X_t^{(M)}, X_t^{(M+1)} = X_t^{(1)}. \tag{5.20}$$

This system is simulated 1000 times on the interval $[0, 5]$ using the EM scheme and a step size of $\Delta t = 1/10000$ with parameters $(F, \sigma, \sigma_V, N) = (8, 10, 2, 2^4)$. As per tradition only every $100th$ sample is kept and the step size for the exponential UKF and the continuous-time UKF were both chosen as $1/100$ and their parameters were set to $\alpha = 10^{-3},\ \beta = 2,\ \kappa = 0$.

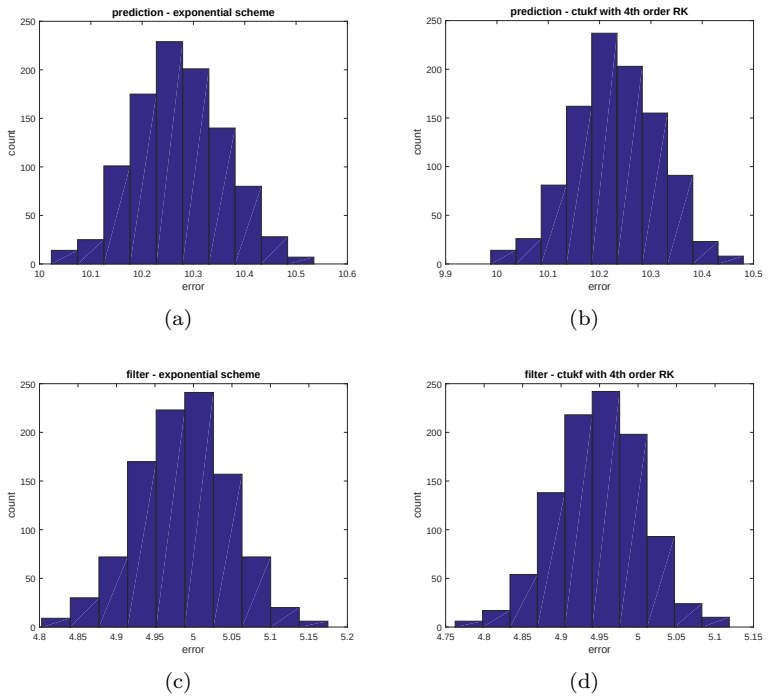


Figure 5.5: Histogram of the square root mean norm errors for a) one-step predictions of the exponential scheme b) one-step predictions of the continuous-discrete UKF c) filter estimates of the exponential scheme d) filter estimates of the continuous-discrete UKF.

The histograms of Figure 5.5 show that the exponential UKF and the continuous-time UKF have fairly similar performance though latter comes out slightly ahead which is easier to see when studying the mean of the performance measures presented in Table 5.3.

In order to get a visual idea of how the Lorenz96 system behaves the system is run with the same parameters except the number of states is set to $M = 4$, the resulting filter estimates and one-step predictions are shown in Figure 5.6.

Table 5.3: The mean of the performance measures, ε_y and ε_x, taken over the 1000 simulations of the Lorenz96 system.

	Continuous-Time UKF	Exponential UKF
ε_y (mean)	10.2311	10.2713
ε_x (mean)	4.9489	4.9863

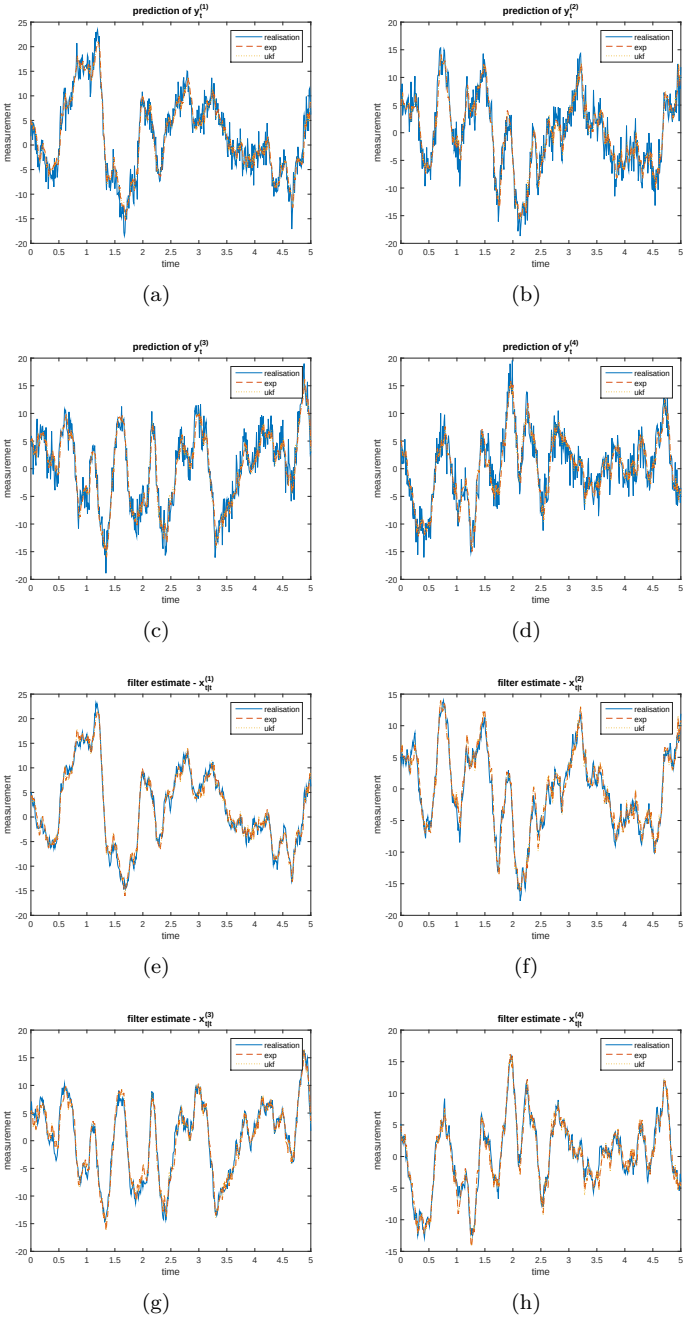


Figure 5.6: The realisation compared to the filter output of the continuous-time UKF and the exponential UKF a)-d) corresponds to the one-step prediction of $Y_n^{(1)}, Y_n^{(2)}, Y_n^{(3)}$ and $Y_n^{(4)}$, e)-h) corresponds to the filter estimate of $X_n^{(1)}, X_n^{(2)}, X_n^{(3)}$ and $X_n^{(4)}$.

5.2 Continuous-Time UKF vs Exponential UKF In Terms of Computational Speed

The experience from the previous sections of the chapter indicates that the Exponential UKF can sometimes tolerate much larger step-sizes than the Continuous-Time UKF though it is not necessarily the case that this translates into much faster run-time. If D_X and D_W are the dimensions of X_t and W_t respectively then for non-linear systems the Exponential UKF does an unscented transform on $Z_t = \begin{pmatrix} X_t & W_{t+h} - W_t \end{pmatrix}^T$ but since,

$$\mathbb{C}\{Z_t, Z_t\} = \begin{pmatrix} \mathbb{C}\{X_t, X_t\} & 0 \\ 0 & \mathbb{C}\{W_{t+h} - W_t, W_{t+h} - W_t\} \end{pmatrix}, \tag{5.21}$$

which implies that $\mathbb{C}\{Z_t, Z_t\}^{1/2}$ can be retrieved from computing $\mathbb{C}\{X_t, X_t\}^{1/2}$ and $\mathbb{C}\{W_{t+h} - W_t, W_{t+h} - W_t\}^{1/2}$. It means that instead of a computing a $(D_X + D_W) \times (D_X + D_W)$ matrix square root two matrix square roots are computed, one of $D_X \times D_X$ and $D_W \times D_W$ so the generation of sigma points is more expensive though not quite as expensive as doing it on a generic vector in $\mathbb{R}^{D_X + D_W}$. It also needs to compute an UT-mean and an UT-covariance containing $2D_X + 2D_W + 1$ terms. Furthermore In order to compute these moments $f(t, X_t)$, $\frac{\partial f}{\partial X_t}, \exp(\frac{\partial f}{\partial X_t} h)$ and $g(t)g(t)^T$ needs to be computed $2D_X + 2D_W + 1$ times.

The Continuous-Time UKF on the other hand, when integrated using a s-stage Runge-Kutta scheme, requires s $D_X \times D_X$ matrix square roots, it also needs to compute s UT-means and UT-covariances that contain $2D_X + 1$ terms each. It also needs to evaluate $f(t, X_t)$ and $g(t)g(t)^T$ in order to compute the UT-moments. With this in mind the exponential UKF, provided it is allowed a much larger step-size, is expected to offer a speed-up for low-dimensional systems but as the dimensions of X_t and W_t grow it will be outrun by the Continuous-Time UKF at some point.

5.3 Parameter Estimation Experiments

5.3.1 The Ornstein Uhlenbeck Process

The Gradient-Based Adaptive Stochastic Search algorithm is tested on the Ornstein Uhlenbeck process given by

$$\mathrm{d}X_t = \lambda(X_t - \mu)\mathrm{d}t + \sigma_X \mathrm{d}W_t, \tag{5.22}$$

$$Y_n = X_{t_n} + V_n, \quad V_n \sim \mathcal{N}(0, \sigma_V^2). \tag{5.23}$$

The process is simulated 500 times on the interval $[0, 10]$ with a step size of $\Delta t = 1/1000$, the time between measurements is $t_n - t_{n-1} = 1/10$ initial value $X_0 \sim \mathcal{N}(m, 1)$ and, the parameters are given below.

$$m = 1,\ \lambda = 2,\ \mu = 1,\ \sigma_X = 0.3,\ \sigma_V = 0.01. \tag{5.24}$$

The GASS algorithm with Polyak averaging is run on each realisation using the following parameters,

$$\rho = 0.3, N_i = 50 + i,\ \alpha_i = (\frac{0.6}{i})^{0.01},\ \varepsilon = 10^{-3}, \tag{5.25}$$

$$I = 40,\ \delta = 0.1,\ c = 0.1 \tag{5.26}$$

$$p(\theta; \gamma) = p(\theta_1; \gamma_1) \dots p(\theta_5; \gamma_5), \tag{5.27}$$

$$\theta_1 \sim \mathcal{N}(0, 10),\ \theta_i \sim \Gamma(3, 1), \quad i = 2, \dots, 5, \tag{5.28}$$

where $\theta = (m, \sigma_V, \lambda, \mu, \sigma_X)$ and the likelihood is computed using the Exponential UKF with a step size of $1/10$ and parameters $\alpha = 10^{-3},\ \beta = 2,\ \kappa = 0$. In order to assess the quality of the estimates the following quantities are computed for each simulation,

$$l(\theta^{GASS}) - l(\theta^{TRUE}),\ \theta_i^{GASS} - \theta_i^{TRUE},\ i = 1, \ldots, 5. \tag{5.29}$$

The histograms of the errors are shown in Figure 5.7 where it can be seen that the likelihood of the parameters from GASS tends to be higher than the likelihood of the true parameters which agrees with common sense since it's unlikely that the optima of the log-likelihood should coincide with the true parameters in the case of a finite sample. Though there appear to be some bias in the parameter estimates, most noticeably for σ_V and μ though a little bias should be expected from the integration error. The mean of the difference in log-likelihood and parameter errors are presented in Table 5.4.

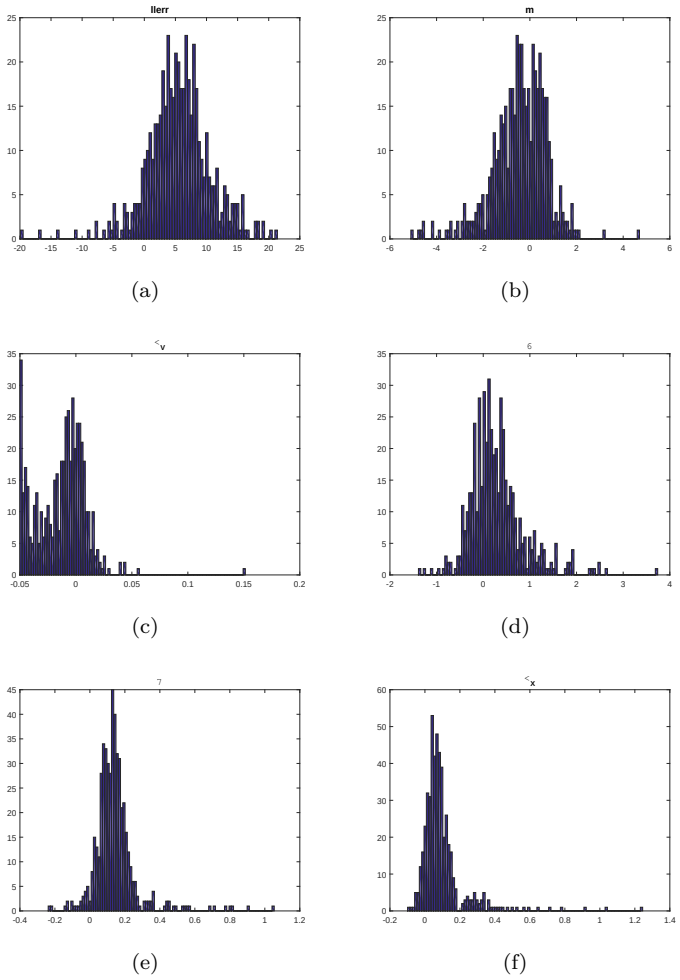


Figure 5.7: Histograms of a) $l(\theta^{GASS}) - l(\theta^{TRUE})$, b) $m^{GASS} - m^{TRUE}$, c) $\sigma_V^{GASS} - \sigma_V^{TRUE}$, d) $\lambda^{GASS} - \lambda^{TRUE}$, e) $\mu^{GASS} - \mu^{TRUE}$ and, f) $\sigma_X^{GASS} - \sigma_X^{TRUE}$.

Table 5.4: The mean of the difference in likelihood and parameter errors for 500 simulations of the OU process.

$l(\theta^{GASS}) - l(\theta^{TRUE})$	5.5361
$m^{GASS} - m^{TRUE}$	-0.4385
$\sigma_V^{GASS} - \sigma_V^{TRUE}$	-0.0132
$\lambda^{GASS} - \lambda^{TRUE}$	0.2914
$\mu^{GASS} - \mu^{TRUE}$	0.1422
$\sigma_X^{GASS} - \sigma_X^{TRUE}$	0.0977

In order to make a comparative assessment the experiment is repeated under the same conditions except the native MATLAB-function `fmincon` is used to find the QMLE. The solver is set to the interior-point algorithm and the optimisation is constrained according to the following,

$$m \in [-5, 5],\ \lambda \in [0, 3],\ \mu \in [-5, 5],\ \sigma_X \in [10^{-6}, 3],\ \sigma_V \in [10^{-6}, 3], \tag{5.30}$$

and the initial solution is chosen at random according to a uniform distribution over the constraints. Histograms of the error are presented in Figure 5.8.

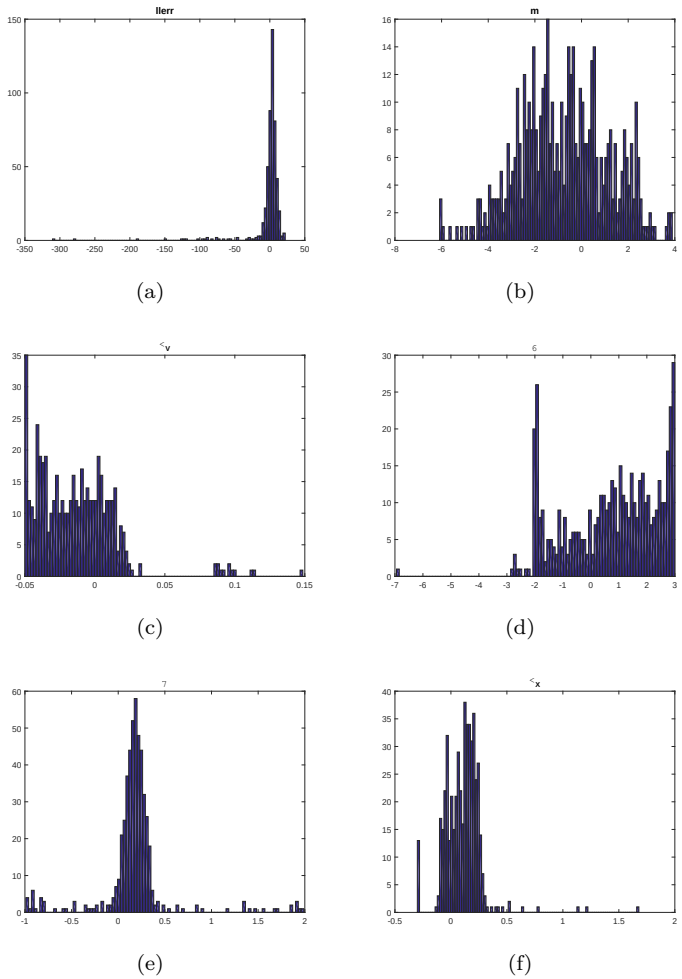


Figure 5.8: Histograms of a) $l(\theta^{IP}) - l(\theta^{TRUE})$, b) $m^{IP} - m^{TRUE}$, c) $\sigma_V^{IP} - \sigma_V^{TRUE}$, d) $\lambda^{IP} - \lambda^{TRUE}$, e) $\mu^{IP} - \mu^{TRUE}$ and, f) $\sigma_X^{IP} - \sigma_X^{TRUE}$.

Clearly the interior-point algorithm has some problem of getting stuck at bad solutions which translates into a higher variation in the parameter estimates. The bias is also higher, see Table 5.5.

Table 5.5: The mean of the difference in likelihood and parameter errors for 500 simulations of the OU process.

$l(\theta^{IP}) - l(\theta^{TRUE})$	-2.0052
$m^{IP} - m^{TRUE}$	-0.7580
$\sigma_V^{IP} - \sigma_V^{TRUE}$	-0.0139
$\lambda^{IP} - \lambda^{TRUE}$	0.7282
$\mu^{IP} - \mu^{TRUE}$	0.1758
$\sigma_X^{IP} - \sigma_X^{TRUE}$	0.1066

5.3.2 The Lorenz63 system

The experiment in Section 5.3.1 is repeated for the following version of the Lorenz63 system,

$$\mathrm{d}X_t = \begin{pmatrix} \sigma(X_t^{(2)} - X_t^{(1)}) \\ X_t^{(1)}(\rho - X_t^{(3)}) - X_t^{(2)} \\ X_t^{(1)} X_t^{(2)} - \beta X_t^{(3)} \end{pmatrix} \mathrm{d}t + \sigma_X I_{3\times 3} \mathrm{d}W_t, \tag{5.31}$$

$$Y_{t_n} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} X_{t_n} + V_n, \quad V_n \sim \mathcal{N}(0, R), \quad R = \begin{pmatrix} \sigma_{V_1} & 0 \\ 0 & \sigma_{V_2} \end{pmatrix}. \tag{5.32}$$

The system is simulated 500 times on the interval $[0, 10]$ with a step size of $\Delta t = 1/1000$, the time between measurements is $t_n - t_{n-1} = 1/100$ initial value $X_0 \sim \mathcal{N}(m, I_{3\times 3})$ and, the parameters are given by,

$$m_1 = m_2 = m_3 = 1,\ \sigma_{V_1} = \sigma_{V_2} = 1,\ \sigma = 10,\ \rho = 28,\ \beta = 8/3,\ \sigma_X = 4.5. \tag{5.33}$$

The likelihood is approximated using the Exponential UKF with a step size of $1/10$ and $\alpha = 10^{-3},\ \beta = 2,\ \kappa = 0$. In order to speed up the experiment GASS is run for 40 iterations with the following starting parameters,

$$\rho = 0.2, N_i = 50 + i,\ \alpha_i = (\frac{0.5}{i})^{0.05},\ \varepsilon = 10^{-3}, \tag{5.34}$$

$$I = 40,\ \delta = 0.1,\ c = 0.3 \tag{5.35}$$

$$p(\theta; \gamma) = p(\theta_1; \gamma_1) \dots p(\theta_9; \gamma_9), \tag{5.36}$$

$$\theta_1, \theta_2, \theta_3 \sim \mathcal{N}(0, 10),\ \theta_i \sim \Gamma(3, 1), \quad i = 4, \dots, 9, \tag{5.37}$$

the resulting distributions is given by,

$$\theta_1 \sim \mathcal{N}(-1.2500, 0.0417), \quad \theta_2 \sim \mathcal{N}(0.6667, 0.0833), \quad \theta_3 \sim \mathcal{N}(1.9091, 0.0455), \tag{5.38}$$

$$\theta_4 \sim \Gamma(68.000, 0.0147), \quad \theta_5 \sim \Gamma(62.000, 0.0149), \quad \theta_6 \sim \Gamma(51.000, 0.2000), \tag{5.39}$$

$$\theta_7 \sim \Gamma(119.00, 0.2500), \quad \theta_8 \sim \Gamma(73.000, 0.0435), \quad \theta_9 \sim \Gamma(79.000, 0.0769), \tag{5.40}$$

which was chosen as the starting distribution for every of the 500 realisations. The remaining parameters were not altered. Histograms of the errors are shown in Figure 5.9 and once again GASS finds parameters that yields a higher likelihood than the true parameters on average. Though it appears there is some bias in the estimates, most prominent in m, s_X, ρ and, β, see Table 5.6.

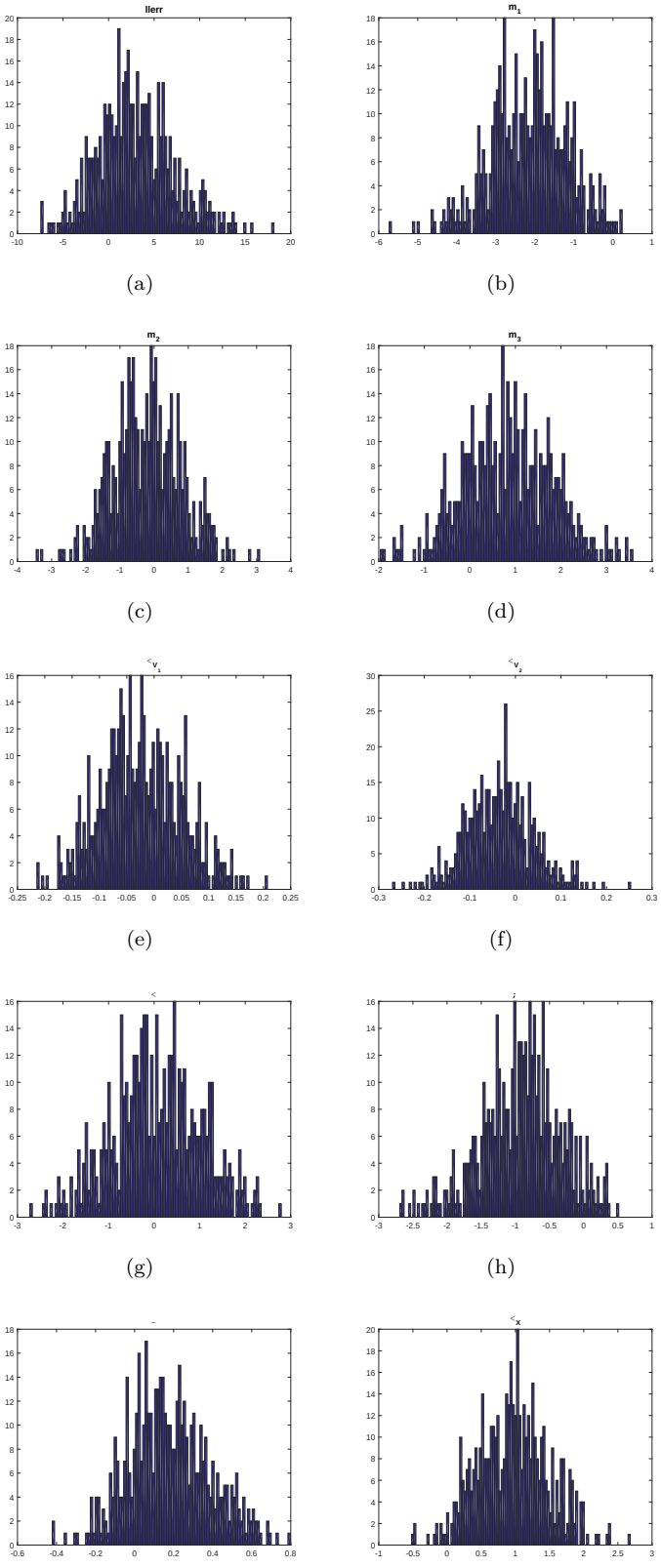


Figure 5.9: Histograms of a) $l(\theta^{GASS}) - l(\theta^{TRUE})$, b)-d) $m_{1:3}^{GASS} - m_{1:3}^{TRUE}$, e)-f) $\sigma_{V_{1:2}}^{GASS} - \sigma_{V_{1:2}}^{TRUE}$, g) $\sigma^{GASS} - \sigma^{TRUE}$, h) $\rho^{GASS} - \rho^{TRUE}$, i) $\beta^{GASS} - \beta^{TRUE}$ and, j) $\sigma_X^{GASS} - \sigma_X^{TRUE}$.

Table 5.6: The mean of the difference in likelihood and parameter errors for 500 simulations of the Lorenz63 system.

$l(\theta^{GASS}) - l(\theta^{TRUE})$	2.9829
$m_1^{GASS} - m_1^{TRUE}$	-2.1537
$m_2^{GASS} - m_2^{TRUE}$	-0.2069
$m_3^{GASS} - m_3^{TRUE}$	0.8337
$\sigma_{V_1}^{GASS} - \sigma_{V_1}^{TRUE}$	-0.0240
$\sigma_{V_2}^{GASS} - \sigma_{V_2}^{TRUE}$	-0.0341
$\sigma^{GASS} - \sigma^{TRUE}$	0.0564
$\rho^{GASS} - \rho^{TRUE}$	-0.9180
$\beta^{GASS} - \beta^{TRUE}$	0.1762
$\sigma_X^{GASS} - \sigma_X^{TRUE}$	0.9629

The experiment is repeated with `fmincon` used to find the QMLE. The solver is set to the interior-point algorithm and the optimisation is constrained according to the following,

$$m \in [-5,5]^3,\ \sigma_{V_1} \in [10^{-6},3],\ \sigma_{V_2} \in [10^{-6},3],\ \sigma \in [0,20], \tag{5.41}$$

$$\rho \in [10,35],\ \beta \in [0.5,10],\ \sigma_X \in [10^{-6},10], \tag{5.42}$$

the algorithm is initialised at a point drawn from a uniform distribution over the constraints. The Histograms of the errors are presented in Figure 5.10 and once again it is discovered that the interior-point algorithm has some trouble with getting stuck at less feasible solutions (mind the scale!). Furthermore the histograms show that the parameter estimates of m, σ and σ_X are not even close to normal distributions which suggests that the interior-point algorithm is a lousy maximiser of the quasi-likelihood in this case, maybe due to occurrence of more than one local maxima. As Table 5.7 shows the bias of the inter-point algorithm is higher than that of GASS almost across the board. A notable exception is β, for which the interior-point algorithm has a bias that is less than that of GASS by several orders of magnitude.

Table 5.7: The mean of the difference in likelihood and parameter errors for 500 simulations of the Lorenz63 system.

$l(\theta^{IP}) - l(\theta^{TRUE})$	-532.6439
$m_1^{IP} - m_1^{TRUE}$	-1.3609
$m_2^{IP} - m_2^{TRUE}$	-1.4833
$m_3^{IP} - m_3^{TRUE}$	-1.5872
$\sigma_{V_1}^{IP} - \sigma_{V_1}^{TRUE}$	0.0746
$\sigma_{V_2}^{IP} - \sigma_{V_2}^{TRUE}$	-0.0723
$\sigma^{IP} - \sigma^{TRUE}$	0.0996
$\rho^{IP} - \rho^{TRUE}$	-1.3757
$\beta^{IP} - \beta^{TRUE}$	0.0089
$\sigma_X^{IP} - \sigma_X^{TRUE}$	3.2630

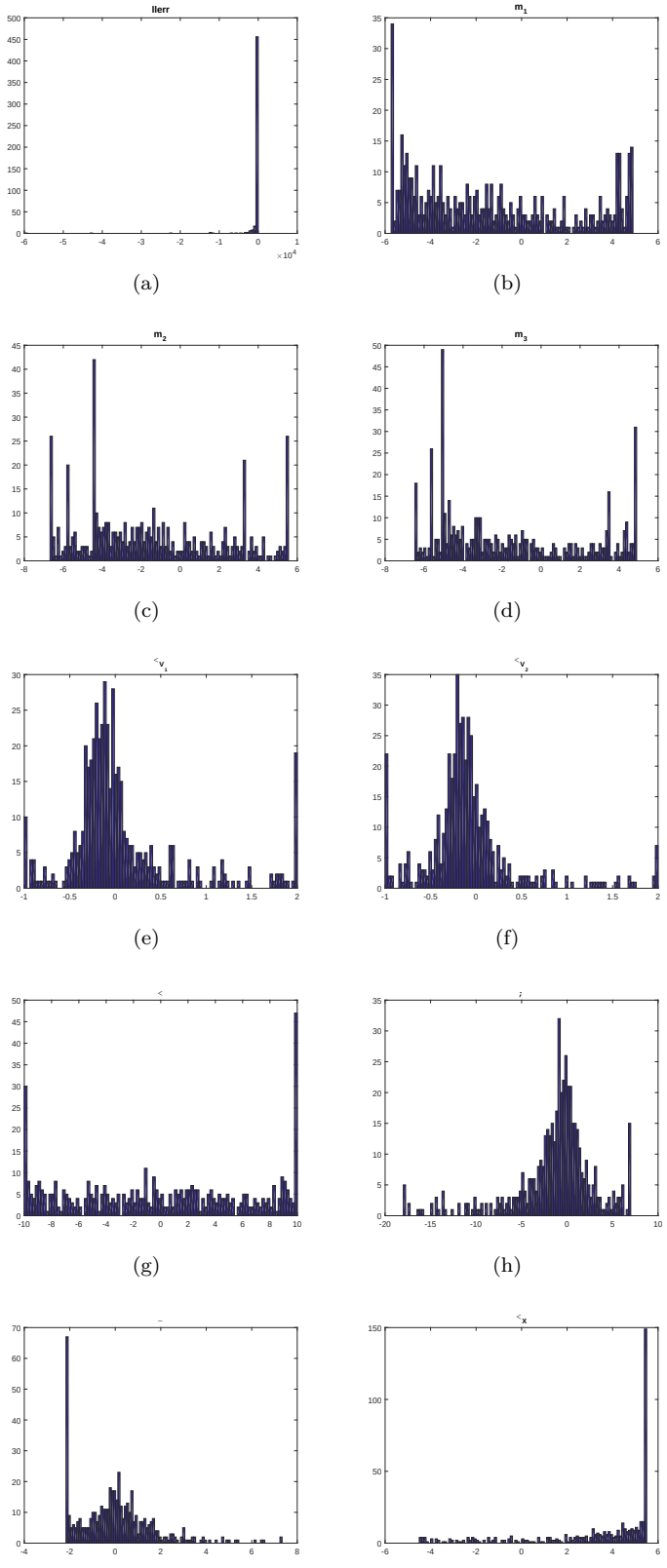


Figure 5.10: Histograms of a) $l(\theta^{IP}) - l(\theta^{IP})$, b)-d) $m_{1:3}^{IP} - m_{1:3}^{TRUE}$, e)-f) $\sigma_{V_{1:2}}^{IP} - \sigma_{V_{1:2}}^{TRUE}$, g) $\sigma^{IP} - \sigma^{TRUE}$, h) $\rho^{IP} - \rho^{TRUE}$, i) $\beta^{IP} - \beta^{TRUE}$ and, j) $\sigma_X^{IP} - \sigma_X^{TRUE}$.

Chapter 6

Conclusion

This thesis discusses some of the basic theory behind stochastic differential equations and how they're incorporated into the framework of state space models by having the states evolve continuously in time but only having measurement at a set of discrete instances. The merits behind driving a state space model with and SDE is that one can side step the issue of irregularly sampled data and that it's fairly straight forward to incorporate partial knowledge of the system in such models. The issue of approximate filtering in non-linear state space models driven by SDEs was also considered in terms of Kalman-type filters such as the Extended Kalman Filter and different ways of producing Unscented Kalman Filters. The conclusion is that it's more likely better performance is achieved by using an UKF rather than the EKF. Furthermore the two different UKFs that were presented, the continuous-time UKF and the Exponential UKF, were compared and the Exponential UKF has the advantage of being able to take larger time steps in many cases which in some situations make it computationally faster without necessarily sacrificing too much in performance.

The issue of parameter estimation was also considered with a brief overview of (Quasi) Maximum Likelihood Estimation and the Gradient-Based Adaptive Stochastic Search algorithm was tested as a method for maximising the likelihood and compared against the MATLAB implementation of the interior-point algorithm. It is concluded that GASS is often better at finding a good candidate for the QMLE. The issue of speed is a little more complicated though a fairly casual stopping-criterion along with the maximum number of iterations set fairly low meant that GASS would run between roughly for the same amount of time as the interior-point algorithm and several minutes shorter. This suggests that GASS is likely to provide a better estimate for the same amount of time spent. The author would like to point out that the stopping-criterion of GASS in itself is quite dodgy as it measures how concentrated the auxiliary distribution is around the maxima of the likelihood. Perhaps it would be wiser, if possible, to have a measure of how likely the algorithm is to make a significant improvement in the solution by continuing. Furthermore, since the parameters of the UKF, α, β and κ, affect the the higher order terms, i.e the error, it would probably be a good idea to estimate these as well rather than fixing them arbitrarily as was done here.

Bibliography

[1] Jiaqiao Hu Enlu Zhou. Gradient-based adaptive stochastic search for non-differentiable optimization. *IEEE TRANSACTION ON AUTOMATIC CONTROL*, VOL 59(NO. 7), July 2014.

[2] Henrik Madsen Erik Lindström and Jan Nygaard Nielsen. *Statistics for Finance*. Chapman and Hall/CRC, 2015.

[3] Henrik Madsen Erik Lindström and Jan Nygaard Nielsen. *Statistics for Finance (Pre-Print)*. Chapman & Hall, 2015.

[4] Simon J. Julier. The scaled unscented transform. *American Control Conference. Proceedings of the 2002 (Volume:6)*, 2002.

[5] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, January 1963.

[6] Henrik Madsen and Jan Holst. *Modelling Non-Linear and Non-Stationary Time Series*. IMM, 2000.

[7] Carlos M. Mora. Weak exponential schemes for stochastic differential equations with additive noise. *IMA Journal of Numerical Analysis*, July 2005.

[8] Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[9] Simo Särkkä. On unscented kalman filtering for state estimation of continuous-time nonlinear systems. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, October 2007.

[10] Simo Särkkä Simon Lyons and Amos Storkey. Series expansion approximations of brownian motion for non-linear kalman filtering of diffusion processes. *IEEE Transactions on Signal Processing, Volume 62, Issue 6, pages 1514-1524*, 2014.