Approximate Bayesian techniques for inference in stochastic dynamical systems

MICHAIL D. VRETTAS

Doctor of Philosophy

– ASTON UNIVERSITY –

September 2010

ASTON UNIVERSITY

Approximate Bayesian techniques for inference in stochastic dynamical systems

MICHAIL D. VRETTAS

Doctor of Philosophy, 2010

Thesis Summary

This thesis is concerned with approximate inference in dynamical systems, from a variational Bayesian perspective. When modelling real world dynamical systems, stochastic differential equations appear as a natural choice, mainly because of their ability to model the noise of the system by adding a variant of some stochastic process to the deterministic dynamics. Hence, inference in such processes has drawn much attention. Here two new extended frameworks are derived and presented that are based on basis function expansions and local polynomial approximations of a recently proposed variational Bayesian algorithm. It is shown that the new extensions converge to the original variational algorithm and can be used for state estimation (smoothing). However, the main focus is on estimating the (hyper-) parameters of these systems (i.e. drift parameters and diffusion coefficients). The new methods are numerically validated on a range of different systems which vary in dimensionality and non-linearity. These are the Ornstein-Uhlenbeck process, for which the exact likelihood can be computed analytically, the univariate and highly non-linear, stochastic double well and the multivariate chaotic stochastic Lorenz ’63 (3-dimensional model). The algorithms are also applied to the 40 dimensional stochastic Lorenz ’96 system. In this investigation these new approaches are compared with a variety of other well known methods such as the ensemble Kalman filter / smoother, a hybrid Monte Carlo sampler, the dual unscented Kalman filter (for jointly estimating the systems states and model parameters) and full weak-constraint 4D-Var. Empirical analysis of their asymptotic behaviour as a function of observation density or length of time window increases is provided.

Keywords: Bayesian inference, variational techniques, dynamical systems, stochastic differential equations, parameter estimation

Contents

List of Figures

List of Tables

To my family

Διονύσιος , Γεωργία και Ελευθερία

Acknowledgements

Undoubtedly, the person to whom I owe the most in completing this PhD thesis is my supervisor Dan Cornford. First, for giving me the opportunity to pursue one of my dreams, by offering me this PhD position and second because throughout all these four years he was very supportive, inspiring and helpful.

I am particularly thankful to professor Manfred Opper which as part of the variational inference in stochastic dynamic environmental models (VISDEM) project, helped me the most in understanding the variational methodology as described in Chapter 4 in this thesis. I also feel very privileged for giving me the opportunity (twice) to visit him in Berlin and work together on different extensions of the proposed variational algorithm. Also special thanks deserve Cedric Archambeau, Yuan Shen and the rest of the VISDEM project team.

An important factor that is often neglected when someone is looking back in time to resume his path is the environment in which that path was taken. I can proudly say that I was a member of the Non-linearity and Complexity Research Group (NCRG)[1] at Aston University. During the first three months of my PhD the pattern analysis and neural networks (PANN) courses, that I had to undertake as part of my training, was a true “crash test” and proved an invaluable tool for the rest of my PhD. I also want to grasp the opportunity and say a big thank you to all members of the NCRG for exchanging ideas and providing insightful comments on various occasions through individual talks and group seminars. In addition I want to thank Vicky Bond for all the administrative assistance that she provided, relieving all the bureaucratic burden.

However a PhD life is not only studying and doing research. For the moments outside the lab I thank in particular my friends Erik Casagrande, Jack Raymond, Remi Barillec, Alexis Boukouvalas, George Lychnos and Patrick McGuire. Special thanks to Erik, Jack and George for being also my flatmates for almost three years.

My PhD was funded for three years from the Engineering and Physical Sciences Research Council (EPSRC), via the VISDEM project, and partially from other sources found by my supervisor. That helped critically in maintaining my focus and energy in doing my research.

Last but not least I want to express my sincere gratitude to my professors from Alexander Technological Educational Institute of Thessaloniki in Greece, Panagiotis Adamidis and Konstantinos Katopodis who encouraged me and supported me in the beginning of my research career.

[1] Formerly known as Neural Computing Research Group.

Glossary & Mathematical Notation

The need for a unified notation in the field of data assimilation has been well established (Ide et al., 1997). In order to assist the reader with the mathematical notation and glossary, used throughout this thesis, the following tables summarize the most commonly found symbols and expressions. Each term will be defined properly, when first appeared and further definitions and clarifications will be provided when necessary. As a general rule bold fonts are used for vectors or matrices, while normal fonts for scalars. Lower-case Latin letters will denote scalars or vectors, whilst upper-case matrices. Greek letters will denote model parameters.

For better presentation the notation has been organised in tables. First are given some commonly found mathematical symbols.

Mathematical symbols and expressions	Description
$\sim$	is distributed as
$\propto$	is proportional to
$\approx$	approximately equal
∂_a	partial derivative with respect to scalar a
$\nabla_{\mathbf{a}}$	gradient with respect to vector $\mathbf{a}$
ln	natural logarithm
$O(n)$	of order n
pdf	probability density function
w.r.t.	with respect to

Next are considered the sets of numbers. In this thesis the most frequently used set is the one of real numbers. However, the set of natural numbers is used when indexing the elements of vectors or matrices, with the asterisk symbol ($*$) denoting exclusion of the zero number.

Sets	Description
$\Re$	set of real numbers
$N^{(*)}$	set of natural numbers (* excluding zero)
$\Re^D$	D-dimensional set of real numbers

To avoid confusion the vectors are considered column-wise unless transposed. When a vector has no index is assumed to be a (continuous) random variable. The most common index is 't' and denotes (continuous) time dependence (e.g. the state vector $\mathbf{x}_t$). For discrete time dependence the index 'k' is more favourable.

Vectors	Description
$\mathbf{x} \in \Re^D$	real valued column vector
$x_i \in \Re$	i'th element of the vector $\mathbf{x}$
$\mathbf{x}_t \in \Re^D$	(continuous) time dependent state vector
$\mathbf{x}_k \in \Re^D$	(discrete) time dependent state vector, i.e. $\mathbf{x}_k = \mathbf{x}_{t=t_k}$
$\mathbf{y}_k \in \Re^D$	(discrete) time dependent observation vector

Matrices follow as a natural extension of vectors. Only upper-case letters are used and unless otherwise stated they are considered square (D×D), where 'D' is the number of rows /columns. If every element of a matrix is time dependent, then for notational convenience this dependency will be denoted as subscript on the whole matrix rather than on each individual element (see Appendix D).

Matrices	Description
$\mathbf{K} \in \Re^{D \times D}$	real valued matrix
$K_{rc} \in \Re$	r'th row c'th column scalar element of $\mathbf{K}$
$\mathbf{K}^{\top}$	transposed matrix
$\mathbf{K}^{-1}$	inverted matrix
$\text{tr}\{\mathbf{K}\}$	trace of matrix
$\lvert\mathbf{K}\rvert$	determinant of matrix
$\text{diag}(\mathbf{K})$	diagonal elements of matrix $\mathbf{K}$
$\mathbf{I} \in \Re^{D \times D}$	Identity matrix

To identify a specific class of distribution, calligraphic capital letters are used, such as $\mathcal{N}$ormal or $\mathcal{G}$amma distribution. The terms 'distribution' and 'density' are used interchangeably and the letter 'p' is used for a general type of distribution, with the type of it (e.g. prior, posterior or likelihood) given individually, at each occurrence. Although an abuse of mathematical notation, this approach is more compact and commonly used.

Distributions	Description
$\mathcal{N}(\mu, \Lambda)$	Normal (Gaussian) distribution
$\mathcal{G}(\alpha, \beta)$	Gamma distribution
$\mathcal{G}^{-1}(a, b)$	Inverse Gamma distribution
$p(\mathbf{x})$	true marginal distribution
$q(\mathbf{x})$	approximate marginal distribution
$p(\mathbf{y}\lvert\mathbf{x})$	conditional distribution of $\mathbf{y}$ given $\mathbf{x}$
$p(\mathbf{y}, \mathbf{x})$	joint distribution of $\mathbf{y}$ and $\mathbf{x}$
$p(\mathbf{x}_{0:N})$	shorthand notation of $p(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N)$

Some special notation that is used to describe the variational framework in Chapter (4) is given in the following table. Proper definitions of the vectors, matrices and functions are also given in the same section.

Special notation	Description
$\mathbf{f}(\mathbf{x}_t) \in \Re^D$	drift function
$\mathbf{g}_L(\mathbf{x}_t) \in \Re^D$	(linear) approximation function
$\boldsymbol{\theta} \in \Re^D$	drift parameter vector
$\mathbf{w}_t \in \Re^D$	Wiener process
$\boldsymbol{\Sigma} \in \Re^{D \times D}$	system noise covariance matrix
$\mathbf{R} \in \Re^{D \times D}$	measurement error covariance matrix
$\mathbf{H} \in \Re^{D \times D}$	(linear) observation operator
$E\{\mathbf{x}\}_{q(\mathbf{x})}$	expectation of $\mathbf{x}$ w.r.t. $q(\mathbf{x})$
$\langle\mathbf{x}\rangle_q$	shorthand notation of $E\{\mathbf{x}\}_{q(\mathbf{x})}$

1 Introduction

CONTENTS

"I cannot believe that God plays dice with the cosmos."
— Albert Einstein, German physicist.

"Consideration of black holes suggests, not only that God does play dice, but that he sometimes confuses us by throwing them where they can't be seen."
— Stephen W. Hawking, British physicist.

1.1 How random are phenomena?

One of the main characteristics that distinguish the human species from the rest of the species on this planet is its intrinsic curiosity to better understand the world that surrounds them. Unlike the other animals, humans are not content to satisfy only their basic needs and instincts, like thirst, hunger, self-preservation, breeding, etc. What is more interesting, is the human ability to create questions that themselves, cannot answer.

In the early beginnings of civilisation, humans were faced with a lot of queries concerning mostly the natural phenomena. The answer, at that time, was simple; for everything was responsible a "God". A God was raising the sun every morning and took it back in the night, another God was responsible for bringing rain, someone else was the one "punishing" humans with natural disasters, when they were misbehaving and so on[1]. However, the advanced ability of humans (compared to the other animals) to observe and draw conclusions helped in finding patterns and creating physical "laws" that describe the observed phenomena. Ultimately the goal of science is to understand how things work and if possible to make predictions about them (Orrell, 2007).

Pierre Simon Laplace, the famous French mathematician and physicist, laid the foundations of deterministic science. He believed that there exists a set of well defined equations that predict everything in the universe (even human behaviour) given an accurate initial condition of the system; i.e. if it was possible to specify the exact position and momentum of every particle, at a single time instant, then the evolution of the universe could be uniquely determined.

The scientific belief that the whole cosmos is completely and uniquely determined by a set of equations that describe everything was very strong, until the beginning of the 20'th century, when the work of two German physicists, Max Planck with the *quantum principle* (early 1900) and Werner Heisenberg with the *uncertainty principle* (1926), laid the foundations of what is known today as *Quantum Theory* or *Quantum Mechanics*. The uncertainty principle, roughly states that the more accurately the position of a particle is measured, the greater the uncertainty in its momentum and vice versa. Therefore, even if Laplace's belief was right and a single mathematical equation, given an infinitesimally accurate initial condition can predict everything, then the *uncertainty principle*, if accepted, makes sure that this cannot happen because we would never be able

[1] Some people still have the same beliefs about the world that surrounds us today.

to measure the initial conditions infinitesimally accurate. Therefore, nature itself limits human curiosity to perform predictions.

Even though quantum mechanics imply that matter is by definition indeterministic and can be described only in a probabilistic way, someone can argue that what appears random to us is because we are still unable to understand the underlying dynamics that pushes the electron from one quantum state to the other. Albert Einstein, one of the most recognized scientists of the past century, contributed a lot to the development of quantum theory (in fact he was awarded a Nobel prize), but was a deeply religious person and refused to accept that randomness exists in the universe and believed, until the very end of his life, that the universe operates under complete *Law and Order*.

Nevertheless, there is room for both scientific beliefs (deterministic and random) to co-exist. Even though quantum phenomena are mostly observed in microscopic level when averaging over a huge number of particles, phenomena can still be adequately described by deterministic laws. After all, following Ockhams' Razor: “a theory should be no more complicated than necessary”.

1.2 From ODEs to SDEs

Consider a system whose macroscopic behaviour (i.e. state of the system x_t) can be described by an ordinary differential equation (ODE) such as:

$$dx_t = f(x_t)dt \; . \tag{1.1}$$

This describes, roughly, that the change in the state of the system (x_t), during the time interval dt is proportional to that time increment dt, with a coupling coefficient $f(x_t)$ that depends on the state of the system at each time[1]. In the deterministic case, given an initial state of the system x_0 there will be a unique solution of Equation (1.1). Another way to see Equation (1.1) is in a form of an *integral equation*. That is:

$$x_t = x_0 + \int_0^t f(x_s)ds \; . \tag{1.2}$$

In reality, however, systems most often incorporate unknown forces, or known but very complex to be represented, that influence their macroscopic behaviour (Honerkamp, 1993). Often the term *noise*, is used to describe these unknown components that cause the system to fluctuate. To capture these fluctuations a random (stochastic) term is introduced to the previous model (Equation 1.1). Hence, the evolution of the system can be better described by an equation of the following form:

$$dx_t = f(x_t)dt + \sigma(x_t)dz_t \; , \tag{1.3}$$

[1] To ease the notation, this section considers only univariate examples.

where $f(x_t)$ is the *drift* function characterising the local trend, $\sigma(x_t)$ is the *diffusion* function, which influences the average size of fluctuations of x_t, and z_t is the *noise process* which often models the effect of faster dynamical modes not explicitly represented in the drift function but present in the real system.

The corresponding integral equation is:

$$x_t = x_0 + \int_0^t f(x_s)ds + \int_0^t \sigma(x_s)dz_s \tag{1.4}$$

The question that now arises is, *since there is no knowledge about the noise term z_t and its effect on the evolution of the system (i.e. $\sigma(x_t)dz_t$), how can this equation be solved and determine the evolution of the system*?

The classical theory of stochastic differential equations is based on the assumption of *Gaussian white noise* (Penland, 2003) and its *"parent"*, the *Wiener process*. As described in Chapter 2, the Wiener process is "almost everywhere" non-differentiable. Therefore, strictly mathematically, it is not permissible to write down the following expression: $\frac{dw_t}{dt}$. However, in a more loose sense it is assumed that this time derivative exists (in a general way) and that is equal to the Gaussian white noise. Hence:

$$\frac{dw_t}{dt} = \xi_t \Rightarrow dw_t = \xi_t dt \ , \tag{1.5}$$

where $\xi_t \in \Re$ is the time dependent Gaussian white noise. Therefore, by substituting the noise process z_t with a Wiener process w_t and the above result (Eq. 1.5) into Equation (1.3), yields:

$$\begin{align} dx_t &= f(x_t)dt + \sigma(x_t)dw_t \tag{1.6} \\ &= f(x_t)dt + \sigma(x_t)\xi_t dt \ , \tag{1.7} \end{align}$$

which is assumed here to provide a general expression for a stochastic differential equation (SDE).

Example

To give an example of the above discussion a simple univariate system is considered, with dynamics described by the following ODE (Eq. 1.8). This system is driven by a force $f(x_t) = \theta \sin(x_t)$, and an example simulation (trajectory), on a five time unit interval, $T = [0,5]$, is shown in Figure 1.1 (left panel, dashed black line). The corresponding SDE is given by Equation (1.9), and a realisation with an additive noise process (i.e. σ is independent of the state x_t), is illustrated in Figure 1.1 (left panel, solid blue line).

$$\begin{align} \frac{dx_t}{dt} &= \theta \sin(x_t) \quad \text{and} \tag{1.8} \\ \frac{dx_t}{dt} &= \theta \sin(x_t) + \sigma \, \frac{dw_t}{dt} \tag{1.9} \end{align}$$

In practice, however, the continuous time equations are transformed to their discrete time counterparts, as shown in Equations (1.10) and (1.11) respectively. Here a simple Euler scheme was chosen for the discretisation of both ODE and SDE (Kloeden and Platen, 1999), which imposed a relatively small time step $\delta t \equiv \delta t_{k+1} - \delta t_k = 0.001$. For this example the drift parameter was set to $\theta = 4$ and the system noise to $\sigma = 1$.

$$\delta x_k = \theta \sin(x_k)\ \delta t \quad \text{and} \tag{1.10}$$

$$\delta x_k = \theta \sin(x_k)\ \delta t + \sigma\sqrt{\delta t}\ \varepsilon_k\ , \tag{1.11}$$

where $\delta x_k = x_{k+1} - x_k$ and $\varepsilon_k \sim \mathcal{N}(0,1)$.

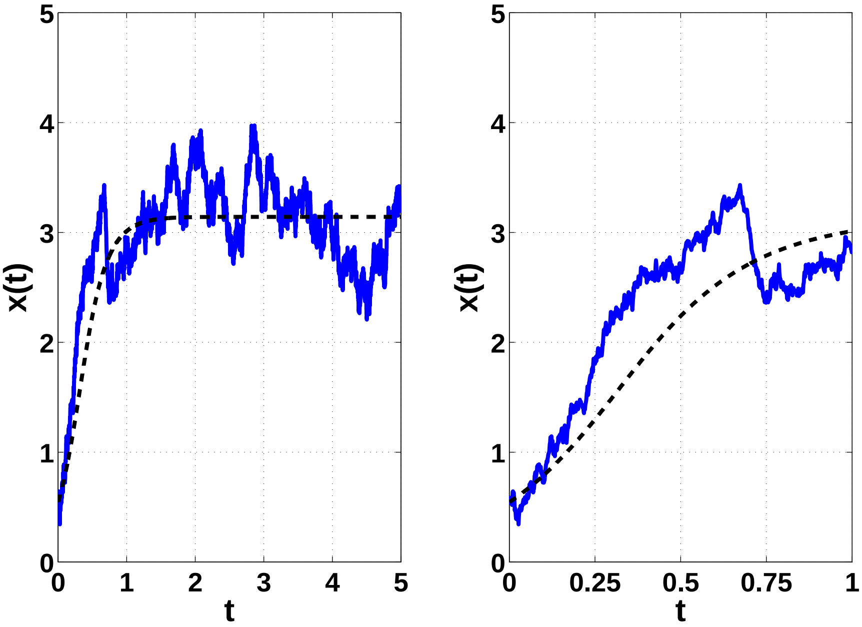


Figure 1.1: **Left panel:** An example of an ordinary differential equation (dashed black line) versus the corresponding stochastic differential equation (solid blue line), simulated on a five time units interval (i.e. $T = [0,5]$). Both trajectories share the same initial state condition ($\mathbf{x}_0$) and have the same setting for the drift parameter $\theta = 4$. The effect of the added random process ($\mathbf{w}_t$) is obvious even from early times in the simulating window. **Right panel:** shows only the first time unit of the simulation, to emphasise how fast the SDE deviates from the ODE even though they start from the exact same point.

Figure 1.1 shows that the solution for the ODE (dashed black line), is smooth and given a fixed initial condition x_0, is unique. On the contrary, the solution for the SDE (solid blue line) is very rough and even though both solutions start with the same initial conditions, it deviates from the deterministic evolution in a random way. Moreover, every time that the SDE is solved the trajectory is different, as a result of the influence of the random noise process w_t.

1.3 Bayesian inference

As implied earlier in Section 1.1, phenomena that appear to evolve in a random manner can be described in a probabilistic way. Shafer (1992), argues that probability can mean many things. The two most prevalent approaches, of probability theory, are the *frequentist* and the *Bayesian* interpretations. Roughly speaking, the frequentist approach interprets Kolmogorov's Axioms for probability, as frequencies. That means the probability of an event is the long-run frequency with which the event occurs with a specific experimental setup. Figure 1.2, shows an example of the probability of appearing "Heads" (blue 'x' symbol) or "Tails" (red circles), when tossing a fair coin. When the number of experimental trials (coin tosses) increases the probabilities of both events tend to $1/2$ (horizontal dashed line), as expected for a "fair" coin.

However, the frequentist approach to probabilities requests not only for an event to have happened, but also to repeat many times (infinite in theory), in order to apply a probability on that event. On the contrary, the Bayesian approach interprets the axioms as degrees of belief (i.e. probabilities can be assigned to quantify beliefs on events that have not yet happened). It is not the intention here to get involved into philosophical discussions on which probabilistic interpretation is correct. Within this thesis the Bayesian approach is adopted and the methods described later are developed in a Bayesian inference framework. The main reason is because within the Bayesian paradigm uncertainty, in making inference, is quantified directly by probabilities based on statistical data analysis, therefore it provides a more principled framework for its treatment.

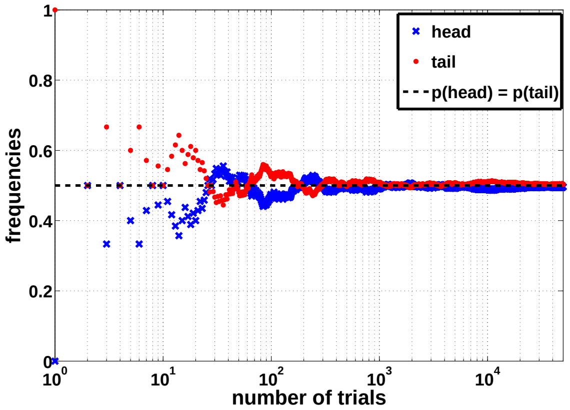


Figure 1.2: Frequency of a "fair" coin, after $50,000$ tosses. As the number of trials increases the frequency that the heads (blue 'x' marks) and tails (red circles) appear tends to the true probability of $1/2$ (horizontal dashed line).

In a Bayesian inference framework (Gelman et al., 1995), everything is expressed with probability distributions. First the problem must be formulated with a "full probability model", which is basically the joint probability density of all the quantities of interest (observed and unobserved). Then, after the prior beliefs have been quantified, in terms of prior probability density functions (pdfs), inference can be characterised as estimating the conditional density of the quantities of interest, given the available observations. In practice, this can be achieved using Bayes' rule[1]:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \,, \quad \text{with} \quad 0 < p(\mathbf{y}) < \infty \,, \tag{1.12}$$

$$\propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) \,. \tag{1.13}$$

In Equation (1.13), $p(\mathbf{x})$ is the *prior* density function, which incorporates all prior beliefs about the quantity $\mathbf{x}$ before seeing any data, $p(\mathbf{y}|\mathbf{x})$ is the *likelihood* of the observed values $\mathbf{y}$ given the current estimates of $\mathbf{x}$, $p(\mathbf{y})$ is the *marginal likelihood* (or evidence) which must be bounded and $p(\mathbf{x}|\mathbf{y})$ is the *posterior* density of the quantities of primary interest conditional on the available observations.

Bayesian inference is very popular in the areas of data assimilation (Wikle and Berliner, 2007) and machine learning (Tipping, 2006), mainly because it provides a natural way to update the current estimates in the light of new observations, by iterating the Bayes rule, Eq. (1.13).

1.4 Thesis outline

Chapter 1 begins with a general discussion about the source of stochasticity (or randomness) found in real world systems and provides a small discussion, including a simple example, on the difference between a deterministic system described by an ODE and a stochastic system defined by an SDE. The viewpoint on why the Bayesian paradigm is appropriate if one wants to make inference about dynamical systems is also highlighted.

Chapter 2 gives some necessary theoretical definitions of stochastic processes, including some properties, to make the rest of the thesis more self-contained. The emphasis is on the *Markov processes* and some characteristic examples are illustrated. Diffusion processes follow and the notion of discrete time observation is clarified. The problem of optimal estimation of the system state and model parameters, given a discrete set of noisy observations is defined. Although an exhaustive review of the methodologies that deal with this problem cannot be claimed an effort is made to collect and present the basic methods on this subject.

Chapter 3 summarizes and briefly reviews the dynamical models that are used in the following chapters to validate the new approximation algorithms that developed. The univariate linear Ornstein- Uhlenbeck process (OU) is introduced and the non-linear Double-Well (DW) follows.

[1] Named after the English mathematician Thomas Bayes (1702 - 1761). Bayes theorem as described in his work, "An Essay towards solving a Problem in the Doctrine of Chances", was published after his death at the '*Philosophical Transactions of the Royal Society of London (1763).*

To test how the methods developed scale to multivariate systems a stochastic version of the three dimensional chaotic Lorenz ’63 system (L3D) is implemented. The last system considered is the forty dimensional stochastic Lorenz ’96 (L40D).

Chapter 4 reviews in detail the variational Gaussian process approximation (VGPA) algorithm, for partially observed diffusions that was first introduced in Archambeau et al. (2007). This is essential because the VGPA algorithm provides the backbone of both extensions that will follow in the next chapters. The state estimation (smoothing) framework is examined first, with two approaches to estimating the model (hyper) parameters following.

Chapter 5 presents an extension of the aforementioned VGPA algorithm in terms of basis function expansions defined globally over the whole time domain of the inference window. Initially, the main characteristics and benefits of using RBFs are highlighted and then the general multivariate framework is derived. Numerical simulations test its convergence properties comparing to the original VGPA algorithm and results of estimating (hyper-) parameters are also included.

Chapter 6 provides an alternative re-parametrisation of the original VGPA framework by using polynomial approximation defined locally between each pair of observations. This approach although similar to the one of the basis function expansions, as presented in Chapter 5, gives a more appropriate approximation framework with beneficial characteristics.

Chapter 7 compares the previously derived extensions with a variety of well known methods of state and parameter estimation. The algorithms are briefly described and the comparison results are presented separately for state and parameter estimation. The asymptotic properties of the local polynomial approximation (as defined in Chapter 6), as the number of observations or length of time window increases, is empirically thoroughly analysed.

Chapter 8 summarizes the work and provides possible future research directions.

1.5 Disclaimer

This thesis is submitted for the degree of Doctor of Philosophy (Ph.D). The work presented here is original and has not been submitted previously for a degree, diploma or qualification at another university. However, parts of the work have been published and presented in the following papers, conferences and seminars (in chronological order):

- Appendix A, which contains the full derivations of the original VGPA framework, has been submitted as a Non-linearity and Complexity Research Group (NCRG) technical report in Vrettas et al. (2008).

- Early theoretical work on both extensions, as described in Chapters 5 and 6, has been accepted and presented at the Bayesian Inference for Stochastic Processes (BISP) workshop, June, 2009.

- The complete theoretical framework (for the univariate case) of the basis function expansion (Chapter 5), along with some preliminary results on the univariate DW system have been presented at the European Symposium on Artificial Neural Networks (ESANN) and published in the conference proceedings (Vrettas et al., 2009). In addition, an extended version of the paper containing the full multivariate RBF framework and results on higher-dimensional systems has been published in Neurocomputing (Vrettas et al., 2010b).

- Comparison results, mainly on state estimation, of the VGPA algorithm with methods implemented in Chapter 7, have been presented at the European Geosciences Union (EGU) conference, April 2010.

- The local polynomial extension along with many results included in Chapters 6 and 7, have been submitted as a journal paper to Physica D (Vrettas et al., 2010a).

- Finally, many views and approaches presented here have been discussed in the Non-linearity and Complexity Research Group (NCRG, Aston University) seminars, on several occasions.

2 Problem statement and existing methodologies

CONTENTS

> "*Probable is what usually happens.*"
> — Aristotle, Greek philosopher.

2.1 Foreword

Chapter 2 introduces the reader to the problem this thesis addresses, as well as the main categories of methodologies that have been developed to solve it. In order to do that it is necessary to first give a review of the main mathematical elements that are used later to built the machinery of the approximation methods that developed. A basic level of probability theory is assumed (e.g. events, sample spaces, probabilities, etc.). Instead of rigorous definitions, intuitive ways of presenting the essential building blocks are preferred. A more detailed presentation on the subject of probability theory and stochastic processes is given by Papoulis (1984).

2.1.1 Chapter outline

The chapter is organised as follows. Initially a definition of a stochastic process is given, including some useful properties. Emphasis is on so-called *Markov processes* and some characteristic examples such as the Gaussian and the Wiener processes are illustrated. The important class of diffusion processes follows and the notion of discrete time observation is clarified. After the basic elements are introduced, the inference problem (from a Bayesian perspective) that provides the focus of this work is properly defined. A review of the methods that address inference in partially observed diffusion processes is given. The chapter concludes with a discussion.

2.2 Stochastic processes

Stochastic processes (also known as random processes) arise naturally in range of different contexts from financial modelling (e.g. the stock market, exchange rate fluctuations), biological modelling (e.g. a patient's EEG) to environmental modelling (e.g. the temperature at a point). It can be seen intuitively as a physical phenomenon which evolves in time in a random or, in a loose sense, probabilistic way. In this section a definition of a stochastic process will be given, highlighting some important properties as well as providing an intuitive view, based on some characteristic examples. It is not the intention to reproduce all the theory around stochastic processes (which would require proper Itō calculus). Instead it only provides the basic definitions and properties that are necessary for the rest of the thesis. An informal and short introduction to stochastic processes can be found in Miller (2007). For a more complete and detailed study of the subject there are excellent textbooks such as Honerkamp (1993); Gardiner (2003), and Kloeden and Platen (1999), where all the concepts are provided in a formal mathematical manner.

Definition 2.2.1 *A stochastic process is a collection of random variables,* $(\mathbf{x}_t)$*, indexed by a set, which here is interpreted as time. Hence if* $T \subset \Re$*, is the time set under consideration and* (Ω, A, P) *a common probability space, then* $\{\mathbf{x}_t\}_{t \in T}$ *is a* ***stochastic process****.*

Thus, it can be seen as a function of two variables $T \times \Omega \to \Re$ such that:

- $\mathbf{x}(t, \cdot) : \Omega \to \Re$ is a random variable $\forall\, t \in T$.
- $\mathbf{x}(\cdot, \omega) : T \to \Re$ is a realization $\forall\, \omega \in \Omega$.

If T is a countable set (discrete case) the stochastic process is called *discrete in time*, otherwise if T is an interval (continuous case) the stochastic process is known as *continuous in time*.

Some important properties, that can characterise whole classes of stochastic processes are:

Property 2.2.1 *Given a partition of time,* $T = \{t_1 < t_2 < \cdots < t_n\}$ *and a positive quantity* $d > 0$*, a stochastic process is* ***strictly stationary*** *if,* $\forall\, t \in T$ *the joint distributions* $(\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_n})$ *and* $(\mathbf{x}_{t_1+d}, \mathbf{x}_{t_2+d}, \ldots, \mathbf{x}_{t_n+d})$ *are identically distributed. That is, time displacements leave the joint distributions unchanged.*

Property 2.2.2 *Given a partition of time,* $T = \{t_1 < t_2 < \cdots < t_n\}$*, a stochastic process is said to have* ***independent increments*** *if,* $\forall\, t \in T$ *the random variables (*$\mathbf{x}_{t_{j+1}} - \mathbf{x}_{t_j}$*), with* $j = 1, 2, \ldots, n-1$ *are independent for any finite combination of time instants.*

Property 2.2.3 *If, for any* $t > s$ *and* $d > 0$*, the distribution of (*$\mathbf{x}_{t+d} - \mathbf{x}_{s+d}$*) is the same as the distribution of (*$\mathbf{x}_t - \mathbf{x}_s$*), then the process is said to have* ***stationary independent increments****.*

Property 2.2.4 *A stochastic process in which if one wants to make a prediction about the state of the system, at a future time '*t_{n+1}*', the only information necessary is the state of the system at the present time '*t_n*', is called a* ***Markov process****.*

Any knowledge about the past (of a Markov process) is redundant. More accurately this is called a "first order" Markov process. This can be generalised to "m'th order" by allowing the process to "remember" the $m-1$ past states. However for the rest of this thesis emphasis is only on "first order" Markov processes unless stated otherwise.

2.2.1 Examples

Gaussian Processes

One of the most well known classes of stochastic processes is the Gaussian process. Here the index set is (often) not considered as the time. A thorough treatment of Gaussian processes can be found in Rasmussen and Williams (2006). Here the formal definition as given in Rasmussen and Williams (2006, Ch.2) is adopted.

Definition 2.2.2 *"A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution."*

The Gaussian process can be fully characterised by its first two moments. For a multivariate process that is:

- means : $\boldsymbol{\mu}_t = \langle \mathbf{x}_t \rangle, \forall\, t \in T.$
- variances : $\boldsymbol{\sigma}_t^2 = \left\langle (\mathbf{x}_t - \boldsymbol{\mu}_t)(\mathbf{x}_t - \boldsymbol{\mu}_t)^\top \right\rangle, \forall\, t \in T.$
- (two-point) covariances : $cov(s,t) = \left\langle (\mathbf{x}_s - \boldsymbol{\mu}_s)(\mathbf{x}_t - \boldsymbol{\mu}_t)^\top \right\rangle, \forall\, t, s \in T$ with $t \neq s$.

Wiener Process

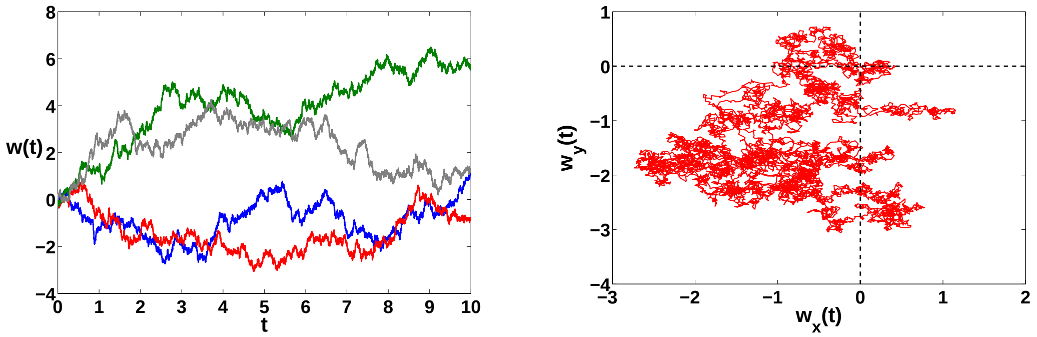
A well studied Gaussian process is the *Wiener process*[1]. It is a continuous-time stochastic process which was proposed to describe the arbitrary movement of a particle pollen on the surface of water, due to the continuous collisions with many water molecules, and is also known as *Brownian motion* or a *continuous random walk.*

Definition 2.2.3 *A Wiener process is a continuous-time Gaussian process that satisfies the Markov property, with independent increments for which:*

- $\mathbf{w}_0 \;=\; 0$ *, with probability 1*
- $\langle \mathbf{w}_t \rangle \;=\; 0$
- $\mathbf{w}_t - \mathbf{w}_s \sim \mathcal{N}(0, t-s)$
- $\left\langle \mathbf{w}_t \cdot \mathbf{w}_s^\top \right\rangle \;=\; \mathbf{I} \cdot min(t,s)$
- $\left\langle d\mathbf{w}_t \cdot d\mathbf{w}_s^\top \right\rangle \;=\; dt \cdot \mathbf{I} \cdot \delta(t-s) \;\;, \forall\, (0 \leq s \leq t) \in T.$

Figure 2.1(a) shows four sample paths, or trajectories, from the standard univariate Wiener process. Notice that although a Wiener sample path is a continuous function of time almost surely, it is not differentiable with probability one; this is called a *rough process*.

[1] Named after the American mathematician Norbert Wiener 1894 - 1964.



(a) 1D Wiener process (b) 2D Wiener process

Figure 2.1: (a) Four different standard Wiener paths are simulated, each one presented with different colour. (b) An illustration of a two dimensional Wiener process. Note that all sample paths start at $\mathbf{w}_0 = 0$.

2.3 Partially observed diffusions

Diffusion processes are a special class of continuous time Markov processes with continuous sample paths, (Kloeden and Platen, 1999). The time evolution of a general, D dimensional, diffusion process $\{\mathbf{x}_t\}_{t\in T}$ can be described by a stochastic differential equation (here to be interpreted in the Itō sense):

$$d\mathbf{x}_t = \mathbf{f}(t,\mathbf{x}_t;\boldsymbol{\theta})\, dt + \boldsymbol{\Sigma}(t,\mathbf{x}_t;\boldsymbol{\theta})^{1/2}\, d\mathbf{w}_t, \qquad d\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, dt\mathbf{I}) \tag{2.1}$$

where $\mathbf{x}_t \in \Re^D$ is the D dimensional latent state vector, $\mathbf{f}(t,\mathbf{x}_t;\boldsymbol{\theta}) \in \Re^D$ is the (typically) non-linear drift function, that models the deterministic part of the system, $\boldsymbol{\Sigma}(t,\mathbf{x}_t;\boldsymbol{\theta}) \in \Re^{D\times D}$ is the diffusion or system noise covariance matrix and $d\mathbf{w}_t$ is the differential of a D dimensional Wiener process, $\{\mathbf{w}_t\}_{t\in T}$, which often models the effect of faster dynamical modes not explicitly represented in the drift function but present in the real system. $T = [t_0, t_f]$ is a fixed time window of inference, with t_0 and t_f denoting the initial and final times respectively. The vector $\boldsymbol{\theta} \in \Re^m$ is a set of parameters within the drift and diffusion functions.

Necessary conditions

To be a diffusion process the following limits must exist for all $0 \le s < t$, with $\delta > 0$ (Kloeden and Platen, 1999):

$$\lim_{t\to s}\left[(t-s)^{-1}\int_{|\mathbf{z}-\mathbf{x}|>\delta} p(s,\mathbf{x};t,\mathbf{z})d\mathbf{z}\right] = 0 \tag{2.2}$$

$$\lim_{t\to s}\left[(t-s)^{-1}\int_{|\mathbf{z}-\mathbf{x}|\le\delta} (\mathbf{z}-\mathbf{x})p(s,\mathbf{x};t,\mathbf{z})d\mathbf{z}\right] = f(s,\mathbf{x}) \tag{2.3}$$

$$\lim_{t\to s}\left[(t-s)^{-1}\int_{|\mathbf{z}-\mathbf{x}|\le\delta} (\mathbf{z}-\mathbf{x})(\mathbf{z}-\mathbf{x})^\top p(s,\mathbf{x};t,\mathbf{z})d\mathbf{z}\right] = \boldsymbol{\Sigma}(s,\mathbf{x}) \tag{2.4}$$

where $\mathbf{x}$, $\mathbf{z} \in \Re^D$, $p(s,\mathbf{x};t,\mathbf{z})$ is the *transition pdf* and the dependence of the drift and diffusion functions on the parameters $\boldsymbol{\theta}$ has been omitted for notational brevity. The first limit Eq. (2.2) prevents the process from having large displacements over a small time interval. Conditions (2.3) and (2.4) are the instantaneous rate of change in the mean (drift function) and covariance (diffusion coefficient), given that the process was at state $\mathbf{x}$ at time s (i.e. $\mathbf{x}(s) \equiv \mathbf{x}_s = \mathbf{x}$).

Discrete observations

Often the latent process is only partially observed, at a small number of ordered discrete times $\{t_k\}_{k=1}^{K}$, which satisfy : $t_0 < t_1 < t_2 < \cdots < t_K < t_f$. In addition the observations are subject to error. Hence

$$\mathbf{y}_k = h(\mathbf{x}_{t_k}) + \boldsymbol{\epsilon}_k \,, \qquad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{2.5}$$

where $\mathbf{y}_k \in \Re^d$ denotes the k'th observation taken at time t_k, $h(\cdot) : \Re^D \to \Re^d$ is the general observation operator and the observation noise $\boldsymbol{\epsilon}_k \in \Re^d$, is assumed (for simplicity) to be independent and identically distributed (i.i.d.) Gaussian white, with covariance matrix $\mathbf{R} \in \Re^{d \times d}$. Note that if the nature of the observations varies at different times then $h_k(\cdot)$ is used instead.

2.4 Problem definition

This thesis addresses the problem of inferring the states of a system ($\mathbf{x}_t$), together with the (possibly) unknown model parameters ($\boldsymbol{\theta}$), from systems that are modelled by diffusion processes and observed at a finite set of discrete time points.

This is an interesting and challenging task because diffusion models have been used extensively in the last few decades to model phenomena that exhibit randomness and evolve continuously in time. Meanwhile, observations from most physical systems arrive at discrete times (e.g. hourly, daily, monthly, etc.).

More precisely, one is dealing with a continuous time system, which is observed at discrete times; and that is what makes the problem difficult. In all but a few examples[1], estimation of diffusion models is not straightforward because the SDE that describes the temporal evolution of the system cannot be solved analytically. Moreover, most real world processes are complex, which implies a non-linear drift $\mathbf{f}(\mathbf{x}_t)$ and diffusion function is necessary, if good agreement with the measurable values is to be achieved. This complicates the statistical analysis even more because the (discrete-time) transition densities Eq. (2.2) are no longer tractable, which means that estimation of the model parameters within a traditional Maximum-Likelihood (ML) framework is not possible. Therefore, approximate techniques are sought.

[1] **(a)** Geometric Brownian motion, **(b)** Ornstein-Uhlenbeck process and **(c)** Cox-Ingressol-Ross process, have log-normal, normal and non-centred chi- squared transition densities respectively.

In a Bayesian framework, the goal is: given a system whose evolution is described by a diffusion (such as Equation 2.1) and a set of discrete time observations (Equation 2.5) to estimate the (smoothing) posterior distribution, $p_s(\mathbf{x}_t|\mathbf{y}_{1:K})$, conditioned on the available observations. The system states might be summarised as the mean $\mathbf{m}_t = \langle \mathbf{x}_t \rangle_{p_s}$, together with a measure of its uncertainty $\mathbf{S}_t = \left\langle (\mathbf{x}_t - \mathbf{m}_t)(\mathbf{x}_t - \mathbf{m}_t)^\top \right\rangle_{p_s}$. In addition, when the model parameters $\boldsymbol{\theta}$ are unknown, an estimate of their value is also desirable.

2.5 Existing methodologies

After describing the main inference problem addressed in this thesis, the current section reviews and discusses the main methodologies that have been employed to solve it. Inference for non-linear stochastic dynamical systems, which are observed at a finite set of discrete time instants, is a challenging task because the *missing paths* between observed values must also be estimated, together with any unknown parameters.

A variety of different approaches has been developed to undertake inference in SDEs. This thesis focuses largely on Bayesian approaches which from a methodological point of view can be grouped into the following three main categories: **(a)** sequential, **(b)** Markov chain Monte Carlo (MCMC) and **(c)** variational approaches. Note that this classification is not unique and others are possible.

2.5.1 Sequential approaches

The first category attempts to solve the *Kushner-Stratonovich-Pardoux* (*KSP*) equations (Kushner, 1967a). The *KSP* method (described briefly in Eyink et al. (2004)), can be applied to give the optimal (in terms of variance minimising estimator) Bayesian posterior solution to the inference problem, providing the exact conditional statistics (often expressed in terms of the mean and covariance) given a set of observations and serves as a benchmark for other approximation methods. Initially, the optimal filtering problem was solved by Kushner and Stratonovich (Stratonovich, 1960; Kushner, 1962, 1967a) and later the optimal smoothing setting was given by an adjoint (backward) algorithm due to Pardoux (1982). Unfortunately, the KSP method is computationally intractable when applied to high dimensional non-linear systems (Kushner, 1967b; Miller et al., 1994), hence a number of approximations have been developed to deal with this issue.

For instance, when the problem is linear the filtering part of the KSP equations (i.e. the forward Kolmogorov equations) boil down to the Kalman and Bucy (1961) filter, which is the continuous time version of the well known Kalman filter (Kalman, 1960). When dealing with systems that exhibit non-linear behaviour a variety of approximations, based on the original Kalman filter (KF), have been proposed to overcome these difficulties. The first approach is to linearise the model (usually up to first order) around the current state estimate, which through a Taylor expansion, requires the derivation of the Jacobian of the model evolution equations. However, this Jacobian might not always be easy to compute. Moreover the model should be smooth enough in the time-scales of interest, otherwise linearisation errors will grow causing the filter estimates to diverge. This method is known as the extended Kalman filter (EKF) (Maybeck, 1979) and was succeeded by a family of methods based on statistical linearisation exploiting the observation that it is easier to approximate a probability distribution than a non-linear operator.

A widely used method that has produced a large body of literature is the ensemble Kalman filter (EnKF) (Evensen, 2003), or when dealing with the smoothing problem the ensemble Kalman smoother (EnKS) (Evensen and van Leeuwen, 1999). Recently another strategy has proposed that rather than sampling this ensemble of particles randomly from the initial distribution it is preferable to select a *design* (i.e. deterministically chose them), so as to capture specific information (usually the first two moments), about the distribution of interest. This method is often called the *unscented transform* and the filtering method is thus referred to as the unscented Kalman filter (UnKF), first introduced by Julier et al. (2000). Another popular, approach is the particle filter by Kitagawa (1987), in which the solution of the posterior density (or KSP equations) is approximated by a discrete set of particles with random support (Kivman, 2003; Fearnhead et al., 2008). This method can be seen as a generalisation of the ensemble Kalman filter, because it does not make the Gaussian assumption when the ensemble is updated in the light of the observations. In other words, if the dynamics of the system are linear then both filters should give the same answer, given a sufficiently large number of particles (ensemble) members.

2.5.2 MCMC approaches

The second category applies Monte Carlo methods to sample from the posterior process, focusing on areas (in the state space) of high probability, based on Markov chains (Neal, 1993). When the dynamics of the system is deterministic, then the sampling problem is on the space of initial conditions. In contrast, when the dynamics is stochastic the sampling problem is on the space of (infinite dimensional) sample paths. Therefore MCMC methods for diffusions are also known as "*path-sampling*" techniques. Although early sampling techniques such as the Geman and Geman (1984) Gibbs sampler can be applied to systems, convergence is often too slow. In order to achieve better mixing of the chain and faster convergence other more complex and sophisticated techniques were developed. Stuart et al. (2004), introduced the *Langevin MCMC* method, which essentially generalises the Langevin equation to sampling in infinite dimensions. A similar approach is the *Hybrid Monte Carlo* (HMC) method (see Duane et al. (1987)) which was later generalised for path sampling problems by Alexander et al. (2005). Both algorithms (Langevin MCMC and HMC) need information on the gradient of the target log-posterior distribution and update the entire trajectory (sample path) at each iteration. They combine ideas of molecular dynamics, employing the Hamiltonian of the system (including a kinetic energy term), to produce new configurations which are then accepted or rejected in a probabilistic way using the Metropolis criterion.

Following the work of Pedersen (1995), on *simulated maximum likelihood estimation* (SMLE), Durham and Gallant (2002) examine a variety of numerical techniques to refine the performance of this method by introducing the notion of the *Brownian bridge*, between two consecutive observations, instead of the Euler discretisation scheme that was used in Pedersen (1995). This lead to various “blocking strategies”, for sampling the sub-paths, such as the one proposed by Golightly and Wilkinson (2006), as an extension to the previous “modified bridge” (Durham and Gallant, 2002). The work of Elerian et al. (2001); Eraker (2001) and Roberts and Stramer (2001) is based on a similar direction, that is augmenting the state with additional data between the measured values, in order to form a complete data likelihood and then use a Gibbs sampler or other sampling techniques (e.g. MCMC). A rather different sampling approach is presented by Beskos et al. (2006b), where an “*exact sampling*” algorithm (in the sense that there are no discretisation errors), is developed that does not depend on data imputation between the observable values, but rather on a technique called *retrospective sampling* (see Papaspiliopoulos and Roberts (2008) for further details). Although this method is very appealing and computationally efficient compared to other sampling methods that depend on fine temporal discretisation to achieve sufficient accuracy, the applicability of the method depends heavily on the *exact algorithm*, as introduced by Beskos et al. (2006a).

2.5.3 Variational approaches

The final category (from a Bayesian point of view) of methodologies approximates the posterior process using variational techniques (Jaakkola, 2001). A popular methodology, which is operational at the *European Centre for Medium-Range Weather Forecasts* (ECMWF), is the four dimensional variational data assimilation method, also known as “4D-Var” (Dimet and Talagrand, 1986). This method seeks the most probable trajectory (or the mode), of the approximate posterior smoothing distribution, within a predefined time window. This is found by minimising a cost function which depends on the measured values and the model dynamics. However, this method does not provide uncertainty estimates around the most probable solution. The “4D-Var” method, as adopted by the ECMWF and others, makes the strong assumption that the model is either perfectly known, or that any uncertainties are negligible and hence can be ignored. A generalisation of this strong *perfect model* assumption, is to accept that the model is not perfect and should be treated as an approximate solution to the real equations governing the system. This leads to a *weak formulation* of 4D-Var as described in Derber (1989); Zupanski (1996). The theory behind the *weak formulation* was introduced in early 70's by Sasaki (1970) - several versions are described in Tremolet (2006) and will be discussed later.

Another variational technique that seeks the conditional mean and variance of the posterior smoothing distribution is described in Eyink et al. (2004). In this work Eyink, argues that the ultimate goal of a data assimilation method is to recover not a specific history that generated the observations, but rather the correct posterior distribution, conditioned upon the observations. To achieve that a *mean field* approximation is applied to the KSP equations, which as discussed earlier provides the optimal filtering and smoothing solution to the inference problem, from a Bayesian perspective. More recently the work of Archambeau et al. (2007), suggested a rather different approach, where the true posterior process is approximated by a time-varying linear dynamical system (such as a non-stationary Gaussian process), rather than assuming a fully factorising form to the joint posterior. This linear approximation assumption implies a fine time discretisation, if good accuracy is to be achieved, and tries to optimise globally the approximate posterior process in terms of minimising the Kullback-Leibler divergence (Kullback and Leibler, 1951), between the two probability measures. This method is further reviewed in Chapter 4.

2.5.4 Non-Bayesian approaches

Although, this thesis addresses the inference problem from a Bayesian perspective, to provide a more complete overview of the proposed methodologies, this section reviews briefly the main non-Bayesian estimation techniques (for reviews see Nielsen et al. (2000) and Sorensen (2004)), that have been developed for inference in partially observed diffusion processes. In general, the methods cited here focus largely in estimating the model parameters (i.e. unknown parameters in the drift and diffusion functions) and can be grouped into: **(i)** analytical and numerical approximations of the true likelihood, **(ii)** estimating functions and **(iii)** indirect inference and efficient method of moments (EMM).

The most appealing methods are those that approximate the true likelihood. This approximation can, theoretically, be made arbitrarily accurate. There are three main types: The first one provides numerical solutions to the Fokker-Planck equation (which is a partial differential equation)[1] and was initially recognised by Lo (1988). Later, various implementations were introduced by Hurn and Lindsay (1997) using spectral approximations and Jensen and Poulsen (2002) using the method of finite differences. The second method obtains estimates of the true likelihood via simulations (Pedersen, 1995; Brandt and Santa-Clara, 2002; Hurn et al., 2003). A common

[1] Also known as the Kolmogorov *forward* equation.

characteristic of these approaches is the use of a numerical scheme (such as the Euler) to move from one state of the system $\mathbf{x}_k$, at time t_k, to the next state $\mathbf{x}_{k+1}$, at time t_{k+1}, in n time steps. Even with efficient modern computers both numerical approaches are quite computationally demanding. The third approach provides analytical, yet very accurate, discrete approximations to the likelihood function (Florens-Zmirou, 1989; Shoji and Ozaki, 1997; Ait-Sahalia, 1999, 2002). The core idea behind these methods is to replace the true transition density with another that has closed-form solutions and includes the (hyper-) parameters $\boldsymbol{\theta}$ of the SDE (Equation 2.1). The simplest case is to use as a proxy for the true transition pdf the Gaussian distribution such as $\mathcal{N}(\mathbf{x}_k + \mathbf{f}(t, \mathbf{x}_k; \boldsymbol{\theta})\ \delta t, \boldsymbol{\Sigma}\delta t)$. However, the resulting mathematical expressions are quite complicated even for low order approximations. Moreover, the bias that is introduced due to the discrete time approximation makes the estimates of the parameters inconsistent for any fixed sampling interval.

Estimation via estimating functions is generally faster (Jacobsen, 2001). Roughly speaking, an estimation function is defined as $F(\mathbf{y}, \boldsymbol{\theta}) : \Re^p \to \Re$, where its arguments are the observations $\mathbf{y}$ and the model parameters $\boldsymbol{\theta}$. The property of this function is that it goes to zero as the parameters $\boldsymbol{\theta}$ tend to their optimal values (i.e. $F \to 0$ as $\boldsymbol{\theta} \to \boldsymbol{\theta}_{opt}$). An example is the *score function* yielding the maximum likelihood estimator. However, for the SDEs the score function is not available therefore alternative solutions are sought. The so called *simple estimating functions* are available in explicit form but provide only estimators for parameters from the marginal distribution (Kessler and Sorensen, 1999; Sorensen, 2000). Still, they may be useful for preliminary analysis, for example in combination with *martingale estimating functions*. The latter are analytically available for a few models but in general they must be simulated (Bibby and Sorensen, 1995). This basically amounts to simulating conditional expectations, which is faster than calculating conditional densities as required by the numerical likelihood approximations mentioned above.

Indirect inference (Gourieroux et al., 1993) and EMM (Gallant and Tauchen, 1996), which is closely related to the General Methods of Moments (Hansen, 1982; Hansen and Scheinkman, 1995; Duffie and Singleton, 1993), introduce discrete time auxiliary (usually wrong) models to approximate the true models. Then, the model parameters of the auxiliary model (e.g. $\boldsymbol{\xi}$) are linked to the true parameters $\boldsymbol{\theta}$ with the so-called binding function (i.e. $\boldsymbol{\xi} = \nu(\boldsymbol{\theta})$). Subsequently, maximum likelihood estimates are obtained for the auxiliary (proxy) model ξ_{ML} and the estimates for the true parameters are obtained using the inverse of the binding function (i.e. $\hat{\boldsymbol{\theta}} = \nu^{-1}(\xi_{ML})$). Nevertheless, the quality of the estimators depends heavily on the auxiliary model which, in essence, is chosen arbitrarily.

Most of the aforementioned methods are, in principle, applicable to multivariate diffusions as well. With a few exceptions this has yet to be demonstrated in practice. Moreover, the computational cost will be even more substantial than for univariate processes. A more comprehensive review for these methods can be found in Jeisman (2005).

2.6 Discussion

This chapter initially introduced the main building blocks that are used extensively later in the thesis. An informal definition of stochastic processes, along with some useful properties and some simple characteristic examples (i.e. the Gaussian and Wiener process), are also provided. A thorough mathematical description of stochastic processes that would require proper stochastic calculus is avoided. Instead a more practical approach is followed and relevant references are cited.

The importance of partially observed diffusions was highlighted. The necessary limit conditions that distinguish them from the other families of the stochastic processes were given and assumed to be satisfied for all the examples in the thesis. Furthermore, the notion of discrete observations was further clarified.

Defining the problem addressed in the thesis is of great importance. The difficulty of obtaining estimates of the system's states together with unknown model parameters was stressed and the major methodologies to tackle this problem were reviewed. Although a complete list of references is not claimed the effort was to gather the most well known and widely accepted methods. In spite of the fact that the inference problem here is placed within a Bayesian framework, alternative (non-Bayesian) approaches that deal with the estimation of parameters in SDEs were also reviewed.

3 Systems studied

CONTENTS

"Essentially, all models are wrong, but some are useful."
— George E. P. Box, English statistician.

3.1 Foreword

When developing new methodologies to solve the inference problem, as described in Chapter 2, it is important to validate them on dynamical systems (or models) with known properties and broad acceptance from the scientific community as benchmark models, before applying them to real world problems. However, to avoid confusion, is also necessary to describe or define what the terms "dynamical system", "dynamical model" mean.

Virtually every physical process that humans observe can be described by a *mathematical model*. That is a set of mathematical expressions (i.e. functions) that form relationships between some properties of the process. Usually, these properties are denoted by a finite set of variables (also known as the *state vector*) and assumed to represent fully, or adequately enough, the state of the process at any given time. This mathematical formulation that describes the temporal evolution of the process is known as a *dynamical system*. Throughout this work the terms "system" and "model" are used interchangeably.

The purpose of this chapter is to summarize and briefly describe the dynamical systems that will be used later to test the algorithms developed. These vary in dimensionality and non-linearity, ranging from univariate linear to forty dimensional non-linear. Characteristic examples are given and the model equations are defined properly for all systems considered.

3.1.1 Chapter outline

Section 3.2, introduces the one dimensional Ornstein-Uhlenbeck process (OU). The linearity in the assumed dynamics of this system allows many analytic calculations and inference to be performed exactly. Next the univariate and strongly non-linear Double Well (DW) system is reviewed in Section 3.3. To identify how the methods developed later scale in higher dimensions Section 3.4, presents a stochastic version of the three dimensional chaotic Lorenz '63 system (L3D). The last system considered is the forty dimensional stochastic Lorenz '96 (L40D), followed by a discussion section that concludes the chapter.

3.2 The Ornstein-Uhlenbeck process

The one dimensional linear Ornstein-Uhlenbeck process (Uhlenbeck and Ornstein, 1930), originates from the physics literature and was proposed as a mathematical model for the velocity of a particle undergoing Brownian motion (see Figure 2.1(b)). Here it is understood as a continuous

Markov process with dynamics that can be represented by the following SDE:

$$dx_t = -\theta x_t \, dt + \sigma \, dw_t \; , \tag{3.1}$$

where $\theta > 0$ is the drift parameter, $\sigma \in \Re$ is the diffusion coefficient (noise standard deviation) and $w_t \in \Re$ is the univariate Wiener process.

In the experiments that follow (see Chapter 6), this system is considered as a reference example. Actually, the solution for the kernel covariance function is known exactly, which is induced by the corresponding Gaussian (Markov) prior process. Karatzas and Shreve (1991), have shown that Eq. (3.1) has the unique strong solution:

$$x_t = \exp\{-\theta t\} \left(x_0 + \sigma \int_0^t \exp\{\theta s\} dw_s \right) \; , \; x_{t=0} = x_0 \; . \tag{3.2}$$

The OU process x_t is a normally distributed random variable with mean and variance given by:

$$\begin{aligned} \langle x_t \rangle &= \left\langle \exp\{-\theta t\} \left(x_0 + \sigma \int_0^t \exp\{\theta s\} dw_s \right) \right\rangle \\ &= \langle \exp\{-\theta t\} x_0 \rangle + \left\langle \exp\{-\theta t\} \sigma \int_0^t \exp\{\theta s\} dw_s \right\rangle \\ &= \exp\{-\theta t\} \langle x_0 \rangle + \exp\{-\theta t\} \sigma \int_0^t \exp\{\theta s\} \langle dw_s \rangle \\ &= \exp\{-\theta t\} \langle x_0 \rangle \; , \end{aligned} \tag{3.3}$$

and

$$\begin{aligned} var[x_t] &= \left\langle (x_t - \langle x_t \rangle)^2 \right\rangle \\ &= \left\langle \left(\exp\{-\theta t\} \left(x_0 + \sigma \int_0^t \exp\{\theta s\} dw_s \right) - \exp\{-\theta t\} \langle x_0 \rangle \right)^2 \right\rangle \\ &= \exp\{-2\theta t\} \left\langle \left((x_0 - \langle x_0 \rangle) + \sigma \int_0^t \exp\{\theta s\} dw_s \right)^2 \right\rangle \\ &= \exp\{-2\theta t\} \left(\left\langle (x_0 - \langle x_0 \rangle)^2 \right\rangle + \sigma^2 \int_0^t \exp\{2\theta s\} \left\langle dw_s^2 \right\rangle \right) \\ &= \exp\{-2\theta t\} \left(\mathrm{var}[x_0] + \sigma^2 \int_0^t \exp\{2\theta s\} ds \right) \\ &= \exp\{-2\theta t\} \left(\mathrm{var}[x_0] + \frac{\sigma^2}{2\theta} \int_0^t \exp\{2\theta s\} (2\theta s)' ds \right) \\ &= \exp\{-2\theta t\} \left(\mathrm{var}[x_0] + \frac{\sigma^2}{2\theta} \Big[\exp\{2\theta s\} \Big]_0^t \right) \\ &= \exp\{-2\theta t\} \left(\mathrm{var}[x_0] + \frac{\sigma^2}{2\theta} \Big[\exp\{2\theta s\} - 1 \Big] \right) \\ &= \exp\{-2\theta t\} \left(\mathrm{var}[x_0] - \frac{\sigma^2}{2\theta} \right) + \frac{\sigma^2}{2\theta} \; , \end{aligned} \tag{3.4}$$

where $\langle dw_s \rangle = 0$ and $\langle dw_s^2 \rangle = ds$ from the properties of the Wiener process (see Section 2.2). In a similar manner the covariance $cov(x_t, x_s)$ is computed as follows:

$$
\begin{aligned}
cov(x_t, x_s) &= \langle (x_t - \langle x_t \rangle)(x_s - \langle x_s \rangle) \rangle \\
&= \langle x_t x_s \rangle - \langle x_t \rangle \langle x_s \rangle \\
&= \left\langle \exp\{-\theta t\} \left(x_0 + \sigma \int_0^t \exp\{\theta\kappa\} dw_\kappa \right) \exp\{-\theta s\} \left(x_0 + \sigma \int_0^s \exp\{\theta\lambda\} dw_\lambda \right) \right\rangle \\
&\quad - \exp\{-\theta(t+s)\} \langle x_0 \rangle^2 \\
&= \exp\{-\theta(t+s)\} \left\langle \left(x_0 + \sigma \int_0^t \exp\{\theta\kappa\} dw_\kappa \right) \left(x_0 + \sigma \int_0^s \exp\{\theta\lambda\} dw_\lambda \right) \right\rangle \\
&\quad - \exp\{-\theta(t+s)\} \langle x_0 \rangle^2 \\
&= \exp\{-\theta(t+s)\} \left(\langle x_0^2 \rangle + \sigma^2 \int_0^t \int_0^s \exp\{\theta(\kappa+\lambda)\} \langle dw_\lambda dw_\kappa \rangle - \langle x_0 \rangle^2 \right) \\
&= \exp\{-\theta(t+s)\} \left(\text{var}[x_0] + \frac{\sigma^2}{2\theta} \Big[\exp\{2\theta s\} - 1 \Big] \right) \\
&= \text{var}[x_0] \exp\{-\theta(t+s)\} + \frac{\sigma^2}{2\theta} \Big(\exp\{-\theta(t-s)\} - \exp\{-\theta(t+s)\} \Big) \,, \qquad (3.5)
\end{aligned}
$$

with $0 \le s \le t$. Note that if $x_0 \sim \mathcal{N}(0, \frac{\sigma^2}{2\theta})$ then $\{x_t\}_{t \in T}$, becomes (strictly) stationary Gaussian process with covariance function (equilibrium kernel):

$$cov(x_t, x_s) = \frac{\sigma^2}{2\theta} \exp\{-\theta(t-s)\} \,. \qquad (3.6)$$

Otherwise, if x_0 is known exactly (i.e. var$[x_0]$ = 0), then the non-equilibrium kernel yields:

$$cov(x_t, x_s) = \frac{\sigma^2}{2\theta} \Big(\exp\{-\theta(t-s)\} - \exp\{-\theta(t+s)\} \Big) \,. \qquad (3.7)$$

From the above expressions it is clear that using the right kernel in a Gaussian process regression smoother, the exact (predictive) posterior process can be computed (Rasmussen and Williams, 2006).

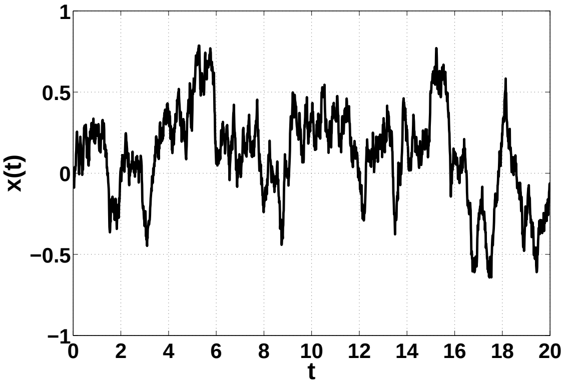


Figure 3.1: Example of an OU trajectory defined on $T = [0, 20]$, with $x_0 = 0$.

An example of an OU trajectory is shown in Figure 3.1, where the simulation is defined on $T = [0, 20]$, with noise variance $\sigma^2 = 0.2$ and drift parameter $\theta = 1$. Applications of the OU process can be found in many disciplines such as physics (e.g. modelling the velocity of a particle) and finance (e.g. modelling interest rates, currency exchange rates, commodity prices, etc.).

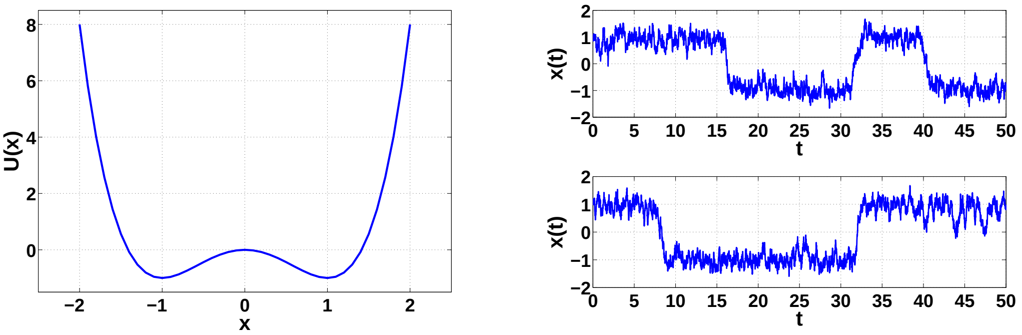
3.3 The double well system

The double well (DW), is a non-linear system with dynamics described by the following stochastically forced scalar differential equation:

$$dx_t = 4x_t(\theta - x_t^2)\, dt + \sigma\, dw_t \;, \tag{3.8}$$

where $\theta > 0$, is the drift parameter and σ, w_t are defined as in Eq. (3.1). The force (i.e. the drift function) of this system arises from a double-well potential function $U(x_t) = -2x_t^2 + x_t^4$, with three equilibrium values at $x_t = 0$ and $x_t = \pm\, 1$. Notice that the drift function in Eq. (3.8), is simply the derivative: $-\frac{dU(x_t)}{dx_t} = 4x_t(1 - x_t^2)$, for $\theta = 1$.

As shown in Figure 3.2(a) the position of a particle at 0 is unstable, while at ± 1 it is stable in the absence of noise. However, within the current setting weak random forces occur which make the state of the system x_t fluctuate about one of the wells for rather long periods of time and occasionally drive the particle from one basin to the other (see Fig. 3.2(b)). This effect is known as "transition" between the two stable states.



(a) Double well potential

(b) Double well simulations

Figure 3.2: (a) The double well potential. The stable points in this example are at $x = \pm 1$, while at position $x = 0$ exists the unstable point. (b) Two examples of DW sample paths including multiple transitions between the two wells, defined on $T = [0, 50]$. The parameter setting for both examples is $\theta = 1$, $\sigma^2 = 0.8$ and $x_0 \sim 0.5\mathcal{N}(0, 2\sigma^2)$.

Systems of this type have been proposed, in the early '80s, as simple models of the earth's climate exhibiting bimodality in which the two deterministic stable states represent conditions such as "normal-age" and "ice-age" (Sutera, 1980; Nicolis and Nicolis, 1981). Although a simple system, the double well has served as a standard benchmark for data assimilation methods in a number of references such as Miller et al. (1994, 1999); Eyink and Restrepo (2000); Eyink et al. (2004); Archambeau et al. (2007, 2008).

3.4 The Lorenz '63 (3D model)

The next system is the stochastic three dimensional chaotic Lorenz '63 (L3D), driven by the following SDE:

$$d\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} \sigma(y_t - x_t) \\ \rho x_t - y_t - x_t z_t \\ x_t y_t - \beta z_t \end{bmatrix} dt + \begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_z \end{bmatrix} d\begin{bmatrix} w_t^x \\ w_t^y \\ w_t^z \end{bmatrix}, \tag{3.9}$$

or in a more compact form by:

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t)\, dt + \mathbf{\Sigma}^{1/2}\, d\mathbf{w}_t\ , \tag{3.10}$$

where $\mathbf{f}(\mathbf{x}_t) \in \Re^3$ is the drift function, with state vector $\mathbf{x}_t = [x_t\ y_t\ z_t]^\top \in \Re^3$ representing all three dimensions, $\boldsymbol{\theta} = [\sigma\ \rho\ \beta]^\top \in \Re^3$, is the drift parameter vector, $\mathbf{\Sigma} \in \Re^{3\times3}$ is a (diagonal) covariance matrix and $\mathbf{w}_t \in \Re^3$ is an uncorrelated multivariate Wiener process.

The deterministic version of this model (i.e. without the noisy part of Eq. (3.9)) was first introduced by Lorenz (1963) as a low dimensional analogue for large scale thermal convection in the atmosphere. It is an approximate model of the convective motion of a fluid that is cooled from above and heated from below. The state vector variables can be physically interpreted as follows: x_t represents the intensity of convective motion, y_t the temperature difference between ascending and descending currents and z_t the distortion of the vertical temperature profile from linearity.

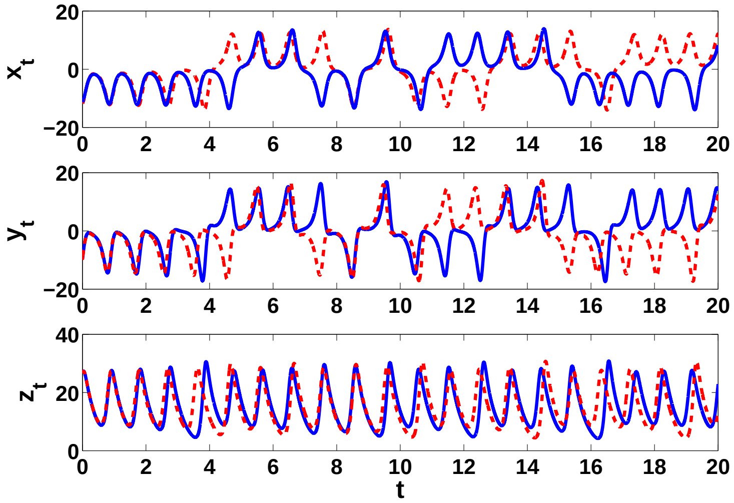


Figure 3.3: Illustration of the L3D chaotic behaviour. Both examples are presented as time series on each separate dimension. The time window is $T = [0, 20]$ for both solutions and the drift vector $\boldsymbol{\theta} = [10\ 28\ 2.6667]^\top$.

This multi-dimensional non-linear system is from the first dynamical systems that was shown to produce chaotic behaviour when its drift parameters σ, ρ and β lie within a specific range of values. The choice of the drift values, in this work, are those to produce chaotic behaviour ($\sigma =$

10.0, $\rho = 28.0$ and $\beta = 8/3$)[1] and they are the most common in use. *Chaotic behaviour* means the solution of the system over a long time window evolves unpredictably (although deterministically), when small changes occur in the initial conditions (i.e. the initial state vector $\mathbf{x}_0$), however it is a well known fact that unstable periodic orbits may occur even in chaotic dynamics. Figure 3.3 demonstrates this chaotic effect on two examples, of the deterministic L3D. Both examples are presented as time series in each separate dimension and their initial conditions were identical except from the first dimension x_t, of the state vector. More precisely, the initial state vector for the blue continuous line is $\mathbf{x}_0^{blue} = [-11.8114 \; -9.9392 \; 26.5024]^\top$, whereas for the red dashed line is $\mathbf{x}_0^{red} = [-11.3500 \; -9.9392 \; 26.5024]^\top$. It is obvious that the two examples start to deviate after the second time unit ($t = 2$), and their paths remain different until the end of their time window.

The incorporation of additive noise to the system equations (see Eq. 3.10), makes the behaviour of the system more unpredictable and adds one more degree of difficulty when applying data assimilation methods. This is illustrated in Figure 3.4, where the same solution $\mathbf{x}_t$ is present in the smooth deterministic version (left column) and the corresponding noisy one (right column). Both examples share the same initial state vector $\mathbf{x}_0$, they are defined on $T = [0, 50]$ and the noisy simulation has diffusion covariance matrix $\mathbf{\Sigma} = \text{diag}\{7, 7, 7\}$. In addition, for all the simulations that follow the deterministic equations were integrated forward in time for $T_{burn} = 5000$ units, in order to get the initial state vector $\mathbf{x}_0$ on the attractor and then the stochastic sample path was generated.

The Lorenz ’63 (or L3D) system, has been studied extensively not only as a standard benchmark but also on its own terms and has produced a large number of references (see for example Evensen (1997); Evensen and van Leeuwen (1999), Miller et al. (1994, 1999) and Hansen and Penland (2006, 2007)).

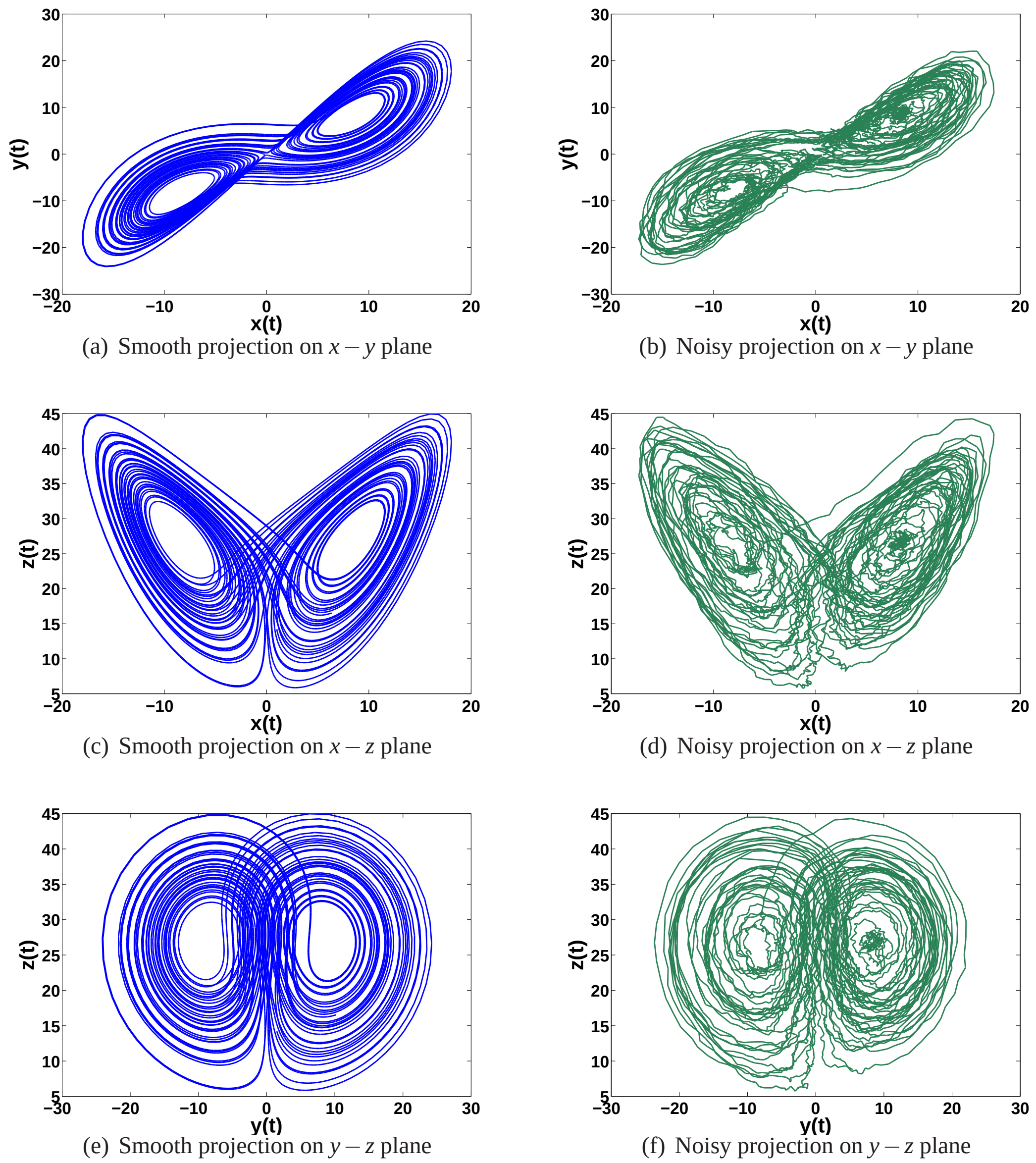
3.5 The Lorenz ’96 (40D model)

Lorenz (1996), introduced a toy model to represent some atmospheric quantity, which consisted of $N > 0$ variables x_t^i, whose evolution is governed by N differential equations, as follows:

$$\frac{dx_t^i}{dt} = (x_t^{i+1} - x_t^{i-2})x_t^{i-1} - x_t^i + \theta \; .$$

Here N is set to forty (i.e. $i \in \{1, 2, \ldots, 40\}$), with *cyclic indices* such as $x_t^{i-N} = x_t^{i+N} = x_t^i$ and $\theta = 8.0$ is the forcing (drift) parameter. These 40 variables form a cyclic chain and can be seen as meteorological variables of 40 sites which are spaced equally around a latitude circle (see Figure 3.5).

[1] In practice, for the experiments that follow, this parameter was set to $\beta = 2.6667$.



(a) Smooth projection on $x-y$ plane

(b) Noisy projection on $x-y$ plane

(c) Smooth projection on $x-z$ plane

(d) Noisy projection on $x-z$ plane

(e) Smooth projection on $y-z$ plane

(f) Noisy projection on $y-z$ plane

Figure 3.4: L3D convection equations: Projections of the deterministic and stochastic examples in phase space. Both simulations have the same initial conditions $\mathbf{x}_0 = [0.9961,\ 1.4949,\ 14.1989]^\top$.

The equations contain quadratic, linear, and constant terms simulating advection, internal damping and external forcing of some atmospheric variable x_t^i, therefore it can be seen as a minimalistic weather model (Lorenz and Emanuel, 1998).

However, in the current framework additive noise is added in every equation, forming the following stochastic differential equation:

$$d\mathbf{x}_t = \begin{bmatrix} (x_t^2 - x_t^{39})x_t^{40} - x_t^1 + \theta \\ (x_t^3 - x_t^{40})x_t^1 - x_t^2 + \theta \\ \vdots \\ (x_t^1 - x_t^{38})x_t^{39} - x_t^{40} + \theta \end{bmatrix} dt + \mathbf{\Sigma}^{1/2}\, d\mathbf{w}_t\ , \quad \theta > 0 \in \Re\ . \tag{3.11}$$

The state vector $\mathbf{x}_t$ consists of forty variables $\left(\text{i.e. } \mathbf{x}_t = \left[x_t^1\ x_t^2\ \dots\ x_t^{40}\right]^\top\right)$, $\mathbf{\Sigma} \in \Re^{40\times 40}$ is the

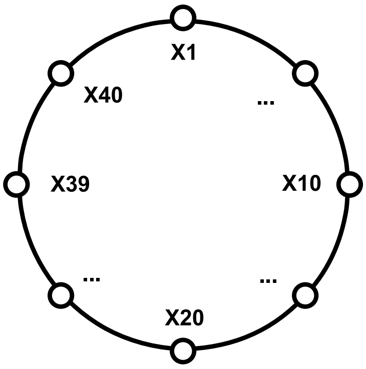
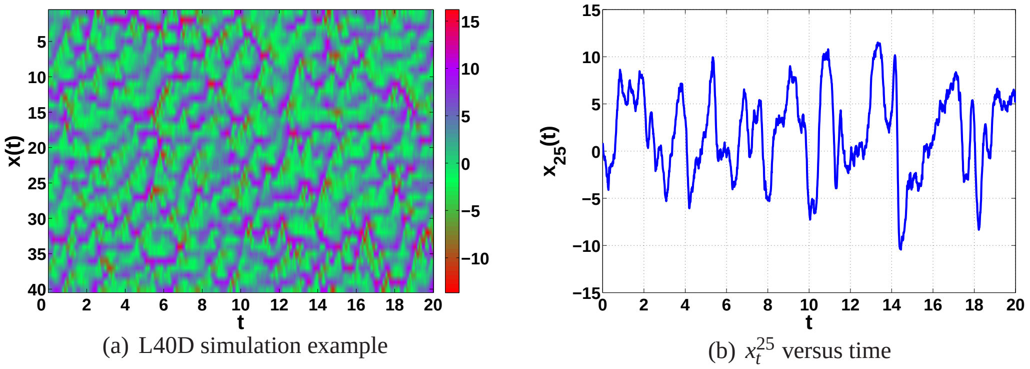


Figure 3.5: Illustration of the sites' placement, at equal distances, on a circular grid for $N = 40$.

diagonal system noise covariance, the drift parameter θ is constant (same for all variables and independent of time) and $\mathbf{w}_t \in \Re^{40}$ is a multidimensional standard Wiener process.



(a) L40D simulation example

(b) x_t^{25} versus time

Figure 3.6: (a) an example of L40D simulation, displaying all forty dimensions. (b) typical time-series example of the 25'th dimension. All simulations are performed on $T = [0, 20]$.

Figure 3.6(a), presents an example of the stochastic L40D simulation on a time interval of twenty units ($T = [0, 20]$). To make the effect of the added noise more apparent, Figure 3.6(b) shows only the 25'th dimension (i.e. the variable x_t^{25} as a function of time). The strength of the random fluctuations, in this particular example, has covariance matrix $\mathbf{\Sigma} = \text{diag}\{7, 7, \dots, 7\}$. A more thorough study of the properties of this proposed system (deterministic version) and some variations of it can be found in Lorenz (2005), as well as Orrell et al. (2001); Orrell (2001, 2003).

3.6 Discussion

This chapter has presented the dynamical systems that are used later in the thesis to test the approximation algorithms developed. Initially, a clarification took place concerning the terms "system" and "model" to avoid confusion. The aim here was not to provide a full description of the systems, but rather to briefly highlight some of their properties and give some characteristic examples.

The choice of the systems was mainly because of their increased dimensionality and non-

System	Dimensions	Linear	Chaotic	Solver	δt
OU	1	yes	no	Euler-Maruyama	0.01
DW	1	no	no	Euler-Maruyama	0.01
L3D	3	no	yes	Euler-Maruyama	0.01
L40D	40	no	yes	Euler-Maruyama	0.01

Table 3.1: Summary of dynamical systems. Column "**Solver**" refers to the numerical integration method that was used to produce the "true" trajectories that generated the observations. In addition, the variable δt represents the time discretisation step of the numerical integration method. Note that when the system is marked as "**Chaotic**", it implies that their model parameters (i.e. drift vector $\boldsymbol{\theta}$), lie within the regimes that produce this chaotic behaviour.

linearity, starting with the one dimensional linear OU process and finishing with the forty dimensional non-linear Lorenz '96 (as seen in Table 3.1). Moreover, the value of these systems is reflected by the number of references that can be found in the literature. Hence a broad acceptance as benchmark models is evident.

Since all the simulations took place on digital computers the "truth", of each system, was generated with numerical integration schemes that discretised the model equations and solved them forwards in time. The method of choice here is the simple first order Euler-Maruyama scheme (Kloeden and Platen, 1999), keeping the time discretisation step δt small, so that good accuracy is achieved.

4 The variational Gaussian process approximation algorithm

CONTENTS

4.1 Foreword

When modelling real world dynamical systems, one must take into account that in general the prior process is not Gaussian. Subsequently, if the prior process is non-Gaussian then the posterior process is also non-Gaussian. If the process is assumed Markovian (see definition in Chapter 2), then any marginal probability can be expressed as a product of the conditional probabilities (i.e. the transition kernels). However, even for the prior process (assuming is non-linear) that would require the solution of the Fokker-Plack equation, which is a partial differential equation. For the majority of the real systems this is not possible, therefore approximation methods are sought.

A popular approximation method in machine learning is the *Gaussian process regression* (MacKay, 1998; Rasmussen and Williams, 2006; Osborne, 2007). The idea of the Gaussian process regression modelling is to place a prior distribution $p(\mathbf{x}_t)$ directly on the space of functions and then perform inference in a Bayesian way. Alternatively, this can be seen as a generalization of the Gaussian distribution over a finite vector space. Hence the approximation reduces to the approximation of a, possibly large, multivariate (but finite dimensional) vector. So the important feature that the process is infinite dimensional almost never plays any practical role.

This chapter reviews the recently proposed variational Gaussian process approximation (hereafter VGPA) method, as was first introduced in Archambeau et al. (2007). This algorithm, follows the variational method (Jaakkola, 2001) to define a linear (Gaussian) process approximation q to the true posterior process p. This is done by minimising the Kullback-Leibler divergence between the two posterior measures, $\mathrm{KL}[q\|p]$. Unlike other variational approaches that enforce a factorising posterior density, in an infinite dimensional setting such formulation does not make much sense. However, such a continuous time setting is not new (Eyink et al., 2004; Apte et al., 2007). The VGPA algorithm, was initially proposed for solving the state estimation (smoothing) problem and later was extended by the authors to include also estimation of (hyper-) parameters (Archambeau et al., 2008).

4.1.1 Chapter outline

The remainder of this chapter is detailed as follows. Section 4.2 introduces the basic setting of the the SDE with the additive noise and the model for the discrete time observations of the algorithm. The core of the VGPA algorithm is reviewed in Section 4.3, where the Bayesian framework is defined first, in terms of the posterior conditional density, and then the so called *variational free energy* is defined and analysed. Section 4.4 outlines the proposed state estimation (smoothing) algorithm and subsequently two approaches of estimating the (hyper-) parameters are described in Section 4.5. Both state and parameter estimation procedures are summarised by pseudocodes in

Tables 4.1 and 4.2 respectively. The chapter concludes with a discussion.

4.2 Basic setting

Equation (2.1) defines a system with multiplicative (i.e. state dependent) system noise. The VGPA framework considers diffusion processes with additive system noise (Archambeau et al., 2007; Beskos et al., 2006b). At first this might seem restrictive, however as stated in Kloeden and Platen (1999), re-parametrisation makes it possible to map a class of multiplicative noise models into this additive class. Hence, the following SDE is considered:

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t)\, dt + \mathbf{\Sigma}^{1/2}\, d\mathbf{w}_t\,, \qquad d\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, dt\mathbf{I}) \tag{4.1}$$

where $\mathbf{x}_t \in \Re^D$ is the (latent) state vector, $\mathbf{f}(\mathbf{x}_t) \in \Re^D$ is (usually) a non-linear drift function, $\mathbf{\Sigma} \in \Re^{D\times D}$ is the noise covariance matrix which for simplicity is assumed diagonal (i.e. $\mathbf{\Sigma} = diag\{\sigma_i^2\}$ for $i = 1, 2, \ldots, D$) and $\{\mathbf{w}_t\}_{t\in T}$ is the standard D dimensional *Wiener process*. Moreover the dependency of the drift function $\mathbf{f}(\mathbf{x}_t)$ to the parameter vector $\boldsymbol{\theta}$ has been suppressed for notational convenience.

Observation model

The stochastic process $\{\mathbf{x}_t\}_{t\in T}$ is assumed to be observed at a finite set of discrete time instants $\{t_k\}_{k=1}^K$, leading to a set of discrete time observations $\{\mathbf{y}_k \in \Re^d\}_{k=1}^K$. In addition the observations are corrupted by i.i.d. Gaussian white noise. Hence:

$$\mathbf{y}_k = h(\mathbf{x}_{t_k}) + \boldsymbol{\epsilon}_k\,, \qquad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})\,. \tag{4.2}$$

Moreover, it is further assumed that the dimensionality of the observation vector is equal to the state's vector (i.e. $d = D$) and that the discrete time measurements are "direct observations" of the state variables (i.e. $\mathbf{y}_k = \mathbf{x}_{t_k} + \boldsymbol{\epsilon}_{t_k}$). This assumption simplifies the presentation of the algorithm and is the most common case in practice. Adding arbitrary observation operators to the equations only affects the system in the observation energy term in Eq. (4.4) and can be readily included if required.

4.3 Approximate inference for diffusions

In this algorithm inference is performed on the *conditional posterior distribution* of the state variables given the observations, thus following the Bayesian paradigm the posterior measure is given as follows:

$$p_{post}(\{\mathbf{x}_t\}_{t\in T}|\mathbf{y}_{1:K}) = \frac{1}{Z}\prod_{k=1}^{K} p(\mathbf{y}_k|\mathbf{x}_{t_k})\, p_{prior}(\{\mathbf{x}_t\}_{t\in T})\,, \tag{4.3}$$

where K denotes the number of noisy observations, Z is the normalising marginal likelihood (i.e. $Z = p(\mathbf{y}_{1:K})$), p_{post} represents the posterior measure *over paths* $\{\mathbf{x}_t\}_{t\in T}$, p_{prior} represents the prior measure over paths defined by Eq. (4.1) and $p(\mathbf{y}_k|\mathbf{x}_{t_k})$ is the likelihood for the observation at time t_k from Eq. (4.2).

4.3.1 Variational Free energy

The VGPA algorithm approximates the true posterior process by another that belongs to a family of tractable ones, in this case the Gaussian processes. This is achieved by minimising the so called "*variational free energy*", defined as follows:

$$\mathcal{F}(q(\mathbf{x}|\boldsymbol{\Sigma}),\boldsymbol{\theta},\boldsymbol{\Sigma}) = -\left\langle \ln \frac{p(\mathbf{y}_{1:K},\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\Sigma})}{q(\mathbf{x}|\boldsymbol{\Sigma})} \right\rangle_{q(\mathbf{x}|\boldsymbol{\Sigma})} , \tag{4.4}$$

where $\mathbf{x} = \{\mathbf{x}_t\}_{t\in T}$, p is the *true* posterior process, q is the *approximate* posterior process and $\langle . \rangle_{q(\mathbf{x}|\boldsymbol{\Sigma})}$ denotes the expectation with respect to $q(\mathbf{x}|\boldsymbol{\Sigma})$. As shown in Archambeau et al. (2008), see also Appendix A, this expression provides an upper bound to the negative log marginal likelihood $-\ln p(\mathbf{y}_{1:K}|\boldsymbol{\theta},\boldsymbol{\Sigma})$:

$$-\ln p(\mathbf{y}_{1:K}|\boldsymbol{\theta},\boldsymbol{\Sigma}) = \mathcal{F}(q(\mathbf{x}|\boldsymbol{\Sigma}),\boldsymbol{\theta},\boldsymbol{\Sigma}) - \mathrm{KL}[q(\mathbf{x}|\boldsymbol{\Sigma})\|p(\mathbf{x}|\mathbf{y}_{1:K},\boldsymbol{\theta},\boldsymbol{\Sigma})] \tag{4.5}$$

$$\leq \mathcal{F}(q(\mathbf{x}|\boldsymbol{\Sigma}),\boldsymbol{\theta},\boldsymbol{\Sigma}) , \quad \text{because KL} \geq 0 . \tag{4.6}$$

However, for this bound to be finite a critical assumption takes place. The system noise covariance (i.e. $\boldsymbol{\Sigma}$), for both processes p and q must be the same. Otherwise the $\mathrm{KL}[q\|p] \rightarrow \infty$, (Archambeau et al., 2008).

4.3.2 Optimal approximate posterior process

The approximation of the true posterior process by a Gaussian process implies that q will be defined by a *linear* SDE. It follows that:

$$d\mathbf{x}_t = \mathbf{g}_L(\mathbf{x}_t)\, dt + \boldsymbol{\Sigma}^{1/2}\, d\mathbf{w}_t , \quad \text{where} \quad \mathbf{g}_L(\mathbf{x}_t) = -\mathbf{A}_t\mathbf{x}_t + \mathbf{b}_t , \tag{4.7}$$

with $\mathbf{A}_t \in \Re^{D\times D}$ and $\mathbf{b}_t \in \Re^D$ define the time varying linear drift in the approximating process, and $\{\mathbf{w}_t\}_{t\in T}$ is a D-dimensional Wiener process with respect to the approximate measure q. Both of these variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$ are time dependent functions that need to be optimised as part of the estimation procedure. The time dependence of these parameters is necessary due to the non-stationarity that is introduced in the process by the observations.

Continuing the derivation of Equation (4.4), as given in Appendix A, leads to the following expression:

$$\mathcal{F}(q(\mathbf{x}),\boldsymbol{\theta},\boldsymbol{\Sigma}) = \mathrm{KL}[q_0\|p_0] + \int_{t_0}^{t_f} E_{sde}(t)dt + \int_{t_0}^{t_f} E_{obs}(t)\sum_k \delta(t-t_k)dt , \tag{4.8}$$

where t_0 and t_f define the initial and final times of the total time window (i.e. $T = [t_0, t_f]$), $\delta(\cdot)$ is Dirac's delta function, $\mathrm{KL}[q_0 \| p_0]$ is a shorthand notation for the KL at the initial state (i.e. $\mathrm{KL}[q(\mathbf{x}_0) \| p(\mathbf{x}_0)]$) and the energy functions are given by:

Energy from the SDE:

$$E_{sde}(t) = \frac{1}{2} \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1} (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t)) \right\rangle_{q_t} . \tag{4.9}$$

Energy from the observations:

$$E_{obs}(t) = \frac{1}{2} \left\langle (\mathbf{y}_t - \mathbf{x}_t)^\top \mathbf{R}^{-1} (\mathbf{y}_t - \mathbf{x}_t) \right\rangle_{q_t} + \frac{d}{2} \ln(2\pi) + \frac{1}{2} \ln |\mathbf{R}| , \tag{4.10}$$

where $\mathbf{y} = \{\mathbf{y}_t \ , t_0 \leq t \leq t_f\} \in \Re^d$ is written as a continuous-time observable process; the discrete time nature of the actual observations adds the delta function in Equation (4.8).

4.3.3 Gaussian process posterior moments

The Gaussian marginal at time 't' is defined as follows:

$$q(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_t, \mathbf{S}_t) \ , \quad t \in T \ , \tag{4.11}$$

where $\mathbf{m}_t \in \Re^D$ and $\mathbf{S}_t \in \Re^{D \times D}$, are respectively the marginal mean and covariance at time 't'. The time evolution of this general time varying linear system in Eq. (4.7), is determined by two ordinary differential equations (ODEs), one for the marginal means $\mathbf{m}_t$ and one for the marginal covariances $\mathbf{S}_t$ (see Eq. 4.11). These are given by the following equations (see Kloeden and Platen, 1999, Ch. 4):

$$\dot{\mathbf{m}}_t = -\mathbf{A}_t \mathbf{m}_t + \mathbf{b}_t \ , \tag{4.12}$$

$$\dot{\mathbf{S}}_t = -\mathbf{A}_t \mathbf{S}_t - \mathbf{S}_t \mathbf{A_t}^\top + \mathbf{\Sigma} \ , \tag{4.13}$$

and thus become functionals of $\mathbf{A}_t$ and $\mathbf{b}_t$, where $\dot{\mathbf{m}}_t \in \Re^D$ and $\dot{\mathbf{S}}_t \in \Re^{D \times D}$ denote the time derivatives $\frac{d\mathbf{m}_t}{dt}$ and $\frac{d\mathbf{S}_t}{dt}$ respectively.

4.4 State estimation (smoothing algorithm)

The parameters that need to be estimated, in order to find the optimal Gaussian process approximation, q_t, are the variational linear $\mathbf{A}_t$ and bias $\mathbf{b}_t$ parameters (recall that these are also functions of time), and the marginal at time 't' means $\mathbf{m}_t$ and covariances $\mathbf{S}_t$.

However, Equations (4.12) and (4.13) are constraints to be satisfied ensuring consistency in the algorithm (Archambeau et al., 2007, 2008). One way to enforce these constraints, within a predefined time window $[t_0, t_f]$, is to formulate the following $\mathcal{L}$agrangian, and then look for its stationary points:

$$\mathcal{L} = \mathcal{F}(q(\mathbf{x}_t|\boldsymbol{\Sigma}), \boldsymbol{\theta}, \boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \left(\boldsymbol{\lambda}_t^\top \underbrace{(\dot{\mathbf{m}}_t + \mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t)}_{\text{ODE for the means}} + \mathrm{tr}\{\boldsymbol{\Psi}_t \underbrace{(\dot{\mathbf{S}}_t + \mathbf{A}_t\mathbf{S}_t + \mathbf{S}_t\mathbf{A}_t{}^\top - \boldsymbol{\Sigma})}_{\text{ODE for the covariances}}\} \right) dt \ , \tag{4.14}$$

where $\boldsymbol{\lambda}_t \in \Re^D$, $\boldsymbol{\Psi}_t \in \Re^{D\times D}$ are time dependent Lagrange multipliers, with $\boldsymbol{\Psi}_t$ being symmetric matrix. Given a set of fixed parameters for the diffusion coefficient $\boldsymbol{\Sigma}$ and the drift $\boldsymbol{\theta}$, minimising this quantity Eq. (4.14) and hence the free energy Eq. (4.4), will lead to the optimal (in the KL-sense) approximate posterior process (Minka, 2005).

Next, taking the functional derivatives of $\mathcal{L}$, with respect to the parameters of interest results in the following equations:

$$\nabla_{\mathbf{A}_t}\mathcal{L} = \nabla_{\mathbf{A}_t}E_{sde}(t) - 2\boldsymbol{\Psi}_t\mathbf{S}_t - \boldsymbol{\lambda}_t\mathbf{m}_t^\top \tag{4.15}$$

$$\nabla_{\mathbf{b}_t}\mathcal{L} = \nabla_{\mathbf{b}_t}E_{sde}(t) + \boldsymbol{\lambda}_t \tag{4.16}$$

$$\nabla_{\mathbf{m}_t}\mathcal{L} = \nabla_{\mathbf{m}_t}E_{sde}(t) + \dot{\boldsymbol{\lambda}}_t - \mathbf{A}_t^\top\boldsymbol{\lambda}_t \tag{4.17}$$

$$\nabla_{\mathbf{S}_t}\mathcal{L} = \nabla_{\mathbf{S}_t}E_{sde}(t) + \dot{\boldsymbol{\Psi}}_t - 2\boldsymbol{\Psi}_t\mathbf{A}_t \ , \tag{4.18}$$

where all these gradients along with the functional gradients of $E_{sde}(t)$, with respect to $\mathbf{A}_t$, $\mathbf{b}_t$, $\mathbf{m}_t$ and $\mathbf{S}_t$, are derived in Appendix A. A closer look at Equations (4.17) and (4.18), shows that setting them equal to zero and rearranging results in a set of ordinary differential equations that describe the time evolution of the Lagrange multipliers $\boldsymbol{\lambda}_t$ and $\boldsymbol{\Psi}_t$:

$$\dot{\boldsymbol{\lambda}}_t = -\nabla_{\mathbf{m}_t}E_{sde}(t) + \mathbf{A}_t^\top\boldsymbol{\lambda}_t \tag{4.19}$$

$$\dot{\boldsymbol{\Psi}}_t = -\nabla_{\mathbf{S}_t}E_{sde}(t) + 2\boldsymbol{\Psi}_t\mathbf{A}_t \ . \tag{4.20}$$

Nonetheless, these ODEs must include the effect of the observations.This is done with two *jump conditions*, which are given by:

$$\boldsymbol{\lambda}(t_k^+) = \boldsymbol{\lambda}(t_k^-) - \nabla_{\mathbf{m}_t}E_{obs}(t_k) \tag{4.21}$$

$$\boldsymbol{\Psi}(t_k^+) = \boldsymbol{\Psi}(t_k^-) - \nabla_{\mathbf{S}_t}E_{obs}(t_k) \ , \tag{4.22}$$

where the superscripts t_k^- and t_k^+ indicate times just before and after the observation time and the functional derivatives of $\nabla_{\mathbf{m}_t}E_{obs}(t_k)$ and $\nabla_{\mathbf{S}_t}E_{obs}(t_k)$ are derived in Appendix A. Due to their discrete time nature the observations create an instantaneous “shock” in the system, at measurement times, whose amplitude is given by the functional derivatives of the observation energy term (E_{obs}), with respect to the marginal mean and variances. These equations are necessary to ensure that the posterior distribution Eq. (4.3) is continuous in time. One must note that choosing another formulation for the estimation problem (i.e. without the use of Lagrange multipliers) different forms of *jump conditions* might be available (Eyink et al., 2004).

A possible algorithm that solves the problem of estimating the optimal Gaussian approximate process, was introduced in Archambeau et al. (2008) and included a *forward* in time ($t_0 \rightarrow t_f$) solution of the ODEs for the means and the covariances (Equations 4.12 and 4.13), followed by a *backward* in time ($t_0 \leftarrow t_f$) solution of the ODEs for the Lagrange multipliers (Equations 4.19 and 4.20), and at the end take one *gradient step* (Equations 4.15 and 4.16). This gradient based algorithm, is briefly summarized in the pseudocode as shown in Table 4.1.

Optimal Gaussian process estimation algorithm

```
fix: $t_0, t_f, \boldsymbol{\theta}, \boldsymbol{\Sigma}, \mathbf{R}, n = 1, N_{max} = 1000$                      \* set initial values *\
initialise $\left(\{\mathbf{A}_t\}_{t=t_0}^{t_f}, \{\mathbf{b}_t\}_{t=t_0}^{t_f}, \mathbf{m}_0, \mathbf{S}_0\right)$   \* initialise the algorithm *\
while $(n \leq N_{max})$                                                   \* inner-loop (START) *\
    fwd-ODEs $\rightarrow \{\mathbf{m}_t, \mathbf{S}_t\}_{t=t_0}^{t_f}$                                  \* compute marginal moments *\
    likelihood $\rightarrow \{E_{obs}, \nabla_{\mathbf{m}_t} E_{obs}, \nabla_{\mathbf{S}_t} E_{obs}\}_{k=1}^{K}$        \* observation likelihood *\
    prior $\rightarrow \{E_{sde}, \nabla_{\mathbf{m}_t} E_{sde}, \nabla_{\mathbf{S}_t} E_{sde}\}_{t=t_0}^{t_f}$            \* prior process energy *\
    bwd-ODEs $\rightarrow \{\boldsymbol{\lambda}_t, \boldsymbol{\Psi}_t\}_{t=t_f}^{t_0}$                                 \* ensure consistency *\
    compute $\{KL0\}$                                                      \* KL at time t=0 *\
    compute $\{\nabla_{\mathbf{A}_t} \mathcal{L}, \nabla_{\mathbf{b}_t} \mathcal{L}\}_{t=t_0}^{t_f}$                               \* new gradients *\
    update $\{\mathbf{A}_t^*, \mathbf{b}_t^*\}_{t=t_0}^{t_f}$                                       \* update variational params *\
    $\mathbf{A}_t \leftarrow \mathbf{A}_t^*, \mathbf{b}_t \leftarrow \mathbf{b}_t^*$                                        \* set the new At and bt *\
    check $\mathcal{L}$ for convergence                                      \* compute Lagrangian *\
    n $\leftarrow$ n+1                                                     \* increase loop counter *\
end while                                                                  \* inner-loop (END) *\
return $(\mathcal{L}, \{\mathbf{A}_t, \mathbf{b}_t, \mathbf{m}_t, \mathbf{S}_t, \boldsymbol{\lambda}_t, \boldsymbol{\Psi}_t\}_{t=t_0}^{t_f})$           \* output (optimal) values *\
```

Table 4.1: Pseudocode of the optimal Gaussian process approximation algorithm in practice. After initialising all the necessary parameters the algorithm iterates, given a fixed set of drift and noise parameters ($\boldsymbol{\theta}$, $\boldsymbol{\Sigma}$ and $\mathbf{R}$), to minimise the Lagrangian cost function. The backward ODEs start with $\boldsymbol{\lambda}(t_f) = 0$ and $\boldsymbol{\Psi}(t_f) = 0$, because at the final time there are no consistency constraints.

4.5 Hyper-parameter estimation

The classical approach to parameter estimation, from incomplete data, is the Expectation - Maximization (EM) algorithm, that was first introduced by Dempster et al. (1977) and later extended to partially observed diffusions by Dembo and Zeitouni (1986). However, even though the EM algorithm is well studied with a broad range of applications it can not be applied successfully in the current variational framework, because the approximate posterior distribution q_t, induced by Eq. (4.7), is restricted to have the same diffusion coefficient $\mathbf{\Sigma}$. Therefore, although an EM approach can be used to estimate the drift parameters $\boldsymbol{\theta}$, the system noise $\mathbf{\Sigma}$ would be held constant during the Maximization step. As a result, different approaches for estimating the parameters have to be adopted.

4.5.1 Discrete approximations to the posterior distributions

As shown in Equation (4.6), the *variational free energy* provides an upper bound to the negative log-marginal likelihood. Thus the negative *free energy* can substitute the log marginal likelihood and by choosing suitable prior distributions $p_0(\boldsymbol{\theta})$ and $p_0(\mathbf{\Sigma})$, $\boldsymbol{\theta}$ and $\mathbf{\Sigma}$ can be treated as random variables and discrete approximations can be constructed to the posterior distribution over the (hyper-) parameters.

For example consider the drift parameters $\boldsymbol{\theta}$. Initially a set of points $\mathtt{D}_{\boldsymbol{\theta}} = \{\boldsymbol{\theta}_i\}_{i=1}^{n_\theta}$ is selected to approximate the posterior distribution and then the variational approximation method runs to convergence with these selected values. This yields to a corresponding set of free energy values $\mathtt{D}_{\mathcal{F}} = \{\mathcal{F}(q(\mathbf{x}|\mathbf{\Sigma}), \boldsymbol{\theta}_i, \mathbf{\Sigma})\}_{i=1}^{n_\theta}$ that can be used to evaluate $\exp\{-\mathcal{F}(q(\mathbf{x}|\mathbf{\Sigma}), \boldsymbol{\theta}_i, \mathbf{\Sigma})\}$, instead of the true marginal likelihood $p(\mathbf{y}_{1:K}|\boldsymbol{\theta}, \mathbf{\Sigma})$:

$$p(\boldsymbol{\theta}|\mathbf{y}_{1:K}) \propto \left\{ \exp\{-\mathcal{F}(q(\mathbf{x}|\mathbf{\Sigma}), \boldsymbol{\theta}_i, \mathbf{\Sigma})\} p_0(\boldsymbol{\theta}_i) \right\}_{i=1}^{n_\theta}, \tag{4.23}$$

where $n_\theta \in N$ is the number of discrete points. Similar discrete approximations, to the posterior distribution, can be computed for the system noise $\mathbf{\Sigma}$. In the above procedure the parameters that are not approximated are kept fixed (to their true values). In the simulations that follow (Chapter 7), Gamma priors are defined for the drift parameters and inverse Gamma for the system noise covariance, i.e. $p_0(\boldsymbol{\theta}) = \mathcal{G}(\alpha, \beta)$ and $p_0(\mathbf{\Sigma}) = \mathcal{G}^{-1}(a, b)$. The values of the parameters α, β, a and b, were chosen such as the mean value of the distribution coincides to the true values of $\boldsymbol{\theta}$ and $\mathbf{\Sigma}$, but with large variance to reflect the initial "ignorance" about the true values of the parameters.

4.5.2 Maximum likelihood type-II point estimates

Another approach for estimating the (hyper-) parameters, as suggested in Archambeau et al. (2008), is also based on the bound that the *variational free energy* provides to the marginal likelihood Eq. (4.6), but instead of constructing approximate posterior distributions to the (hyper-) parameters, as in the previous section, it employs a conjugate gradient algorithm to provide point estimates. More specifically, the algorithm works in an outer / inner loop optimisation framework, where in the inner loop the variational approximation framework is used to compute the optimal approximate posterior process $q(\mathbf{x}_t)$, given a fixed set of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ (Table 4.1). Then, in the outer loop, a gradient step is taken to improve the current estimates of the (hyper-) parameters. This procedure, as shown in Table 4.2, alternates until the gradients of the optimal process Eq. (4.14), with respect to the $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ are zero ($\nabla_{\boldsymbol{\theta}}\mathcal{L} = 0$ and $\nabla_{\boldsymbol{\Sigma}}\mathcal{L} = 0$), or the estimates cannot improve any further (i.e. the optimal Gaussian process estimated in the inner loop does not change significantly, e.g. $\Delta\mathcal{L} \leq 1.0e-6$ in Table 4.2).

ML type-II parameter estimation algorithm

```
fix: $\{\boldsymbol{\theta}_0, \boldsymbol{\Sigma}_0, n = 1, N_{max} = 1,000\}$          \* initialise the algorithm *\
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$, $\boldsymbol{\Sigma} \leftarrow \boldsymbol{\Sigma}_0$          \* set the initial parameter values *\
$\mathcal{L} \leftarrow$ inner-loop($\boldsymbol{\theta}$, $\boldsymbol{\Sigma}$)          \* optimal process from Table 4.1 *\
while ($n \leq N_{max}$)          \* outer-loop (START) *\
    compute$\{\nabla_{\boldsymbol{\theta}}\mathcal{L}, \nabla_{\boldsymbol{\Sigma}}\mathcal{L}\}$          \* gradients w.r.t. the parameters *\
    if ($\nabla_{\boldsymbol{\theta}}\mathcal{L}^\top\nabla_{\boldsymbol{\theta}}\mathcal{L} == 0$ or $\nabla_{\boldsymbol{\Sigma}}\mathcal{L}^\top\nabla_{\boldsymbol{\Sigma}}\mathcal{L} == 0$)          \* check if the gradients are zero *\
        return$\{\boldsymbol{\theta}, \boldsymbol{\Sigma}\}$          \* return the old parameter values *\
    end
    update$\{\boldsymbol{\theta}^*, \boldsymbol{\Sigma}^*\}$          \* new parameter values *\
    $\mathcal{L}^* \leftarrow$ inner-loop($\boldsymbol{\theta}^*$, $\boldsymbol{\Sigma}^*$)          \* new cost function value *\
    if $\{\Delta\mathcal{L}^*$ & $\Delta\boldsymbol{\theta}^*$ & $\Delta\boldsymbol{\Sigma}^*\} \leq 1.0e-6$          \* check for termination *\
        return$\{\boldsymbol{\theta}^*, \boldsymbol{\Sigma}^*\}$          \* return the new parameter values *\
    end
    $\mathcal{L} \leftarrow \mathcal{L}^*$, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^*$, $\boldsymbol{\Sigma} \leftarrow \boldsymbol{\Sigma}^*$          \* set the old values to the new *\
    n $\leftarrow$ n+1          \* increase the loop counter by one *\
end while          \* outer-loop (END) *\
return$\{\boldsymbol{\theta}, \boldsymbol{\Sigma}\}$          \* if it has not convergence yet *\
```

Table 4.2: Pseudocode of the "ML type-II" point estimation algorithm in practice. Every time the parameters are updated the *inner-loop($\boldsymbol{\theta}$,$\boldsymbol{\Sigma}$)* function, see Table 4.1, recomputes the optimal Gaussian process approximation for a given set of fixed parameter values.

This method is referred here as *Maximum-Likelihood type-II* or *ML type-II*, for brevity. In practice, to make the comparison with other Bayesian estimation methods (Chapter 7) more fair, prior distributions over the (hyper-) parameters have been assigned, as shown in the previous section (i.e. $p_0(\boldsymbol{\theta}) = \mathcal{G}(\alpha, \beta)$ and $p_0(\boldsymbol{\Sigma}) = \mathcal{G}^{-1}(a, b)$). Therefore the algorithm provides approximate MAP point estimates.

4.5.3 Parameters to estimate

The parameters to estimate are the prior mean and variance over the initial state $\mathbf{x}_0$, the parameters in the drift function $\boldsymbol{\theta}$, the diagonal elements of the system noise covariance matrix $\boldsymbol{\Sigma}$ and the parameters related to the observable process $\mathbf{R}$. When using the point estimate approach, the gradients of the (cost) Lagrangian function, with respect to the parameters of interest need to be computed. These are given as follows.

Initial state: The initial approximate posterior process $q(\mathbf{x}_0)$, is equal to $\mathcal{N}(\mathbf{m}_0, \mathbf{S}_0)$, where the initial true posterior process $p(\mathbf{x}_0)$ is chosen to be an isotropic Gaussian (i.e. $\mathcal{N}(\boldsymbol{\mu}_0, \tau_0\mathbf{I})$). Taking the gradients of Eq. (4.14), with respect to $\mathbf{m}_0$ and $\mathbf{S}_0$ leads to the following expressions:

$$\nabla_{\mathbf{m}_0}\mathcal{L} = \boldsymbol{\lambda}_0 + \tau_0^{-1}(\mathbf{m}_0 - \boldsymbol{\mu}_0) \tag{4.24}$$

$$\nabla_{\mathbf{S}_0}\mathcal{L} = \boldsymbol{\Psi}_0 + \frac{1}{2}\left(\tau_0^{-1}\mathbf{I} - \mathbf{S}_0^{-1}\right) . \tag{4.25}$$

Drift parameters: Similarly the gradient of Eq. (4.14), with respect to $\boldsymbol{\theta}$ is given:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L} &= \int_{t_0}^{t_f} \nabla_{\boldsymbol{\theta}} E_{sde}(t)dt & (4.26)\\ &= \int_{t_0}^{t_f} \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \boldsymbol{\Sigma}^{-1} \nabla_{\boldsymbol{\theta}}\mathbf{f}(\mathbf{x}_t) \right\rangle_{q_t} dt , & (4.27)\end{aligned}$$

where $\nabla_{\boldsymbol{\theta}} E_{sde}(t)$ has been computed as shown in Appendix A.

System noise: The gradient of Eq. (4.14), with respect to the system noise covariance $\boldsymbol{\Sigma}$ is given by:

$$\begin{aligned}\nabla_{\boldsymbol{\Sigma}}\mathcal{L} &= \int_{t_0}^{t_f} \nabla_{\boldsymbol{\Sigma}} E_{sde}(t)dt + \int_{t_0}^{t_f} \boldsymbol{\Psi}_t dt & (4.28)\\ &= -\int_{t_0}^{t_f} \frac{1}{2}\boldsymbol{\Sigma}^{-1} \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \right\rangle_{q_t} \boldsymbol{\Sigma}^{-1} dt + \int_{t_0}^{t_f} \boldsymbol{\Psi}_t dt , & (4.29)\end{aligned}$$

where the matrix $\boldsymbol{\Sigma}$ is assumed symmetric.

Observation noise: Finally, the gradient of Eq. (4.14) with respect to the observation noise covariance $\mathbf{R}$ is given by:

$$\begin{aligned}\nabla_{\mathbf{R}}\mathcal{L} &= \int_{t_0}^{t_f} \nabla_{\mathbf{R}} E_{obs}(t) \sum_n \delta(t - t_n)dt & (4.30)\\ &= \frac{1}{2}\mathbf{R}^{-1} \int_{t_0}^{t_f} \left(\mathbf{I} - \left\langle (\mathbf{y}_t - h(\mathbf{x}_t))(\mathbf{y}_t - h(\mathbf{x}_t))^\top \right\rangle_{q_t} \mathbf{R}^{-1}\right) \sum_n \delta(t - t_n)dt , & (4.31)\end{aligned}$$

where the general observation operator $h(\cdot)$ is left to provide a more general expression. In the case that this operator is linear (or even identity), then the above expression can be further simplified.

4.6 Discussion

This chapter reviewed a recently proposed variational Gaussian process approximation algorithm, for inference in partially observed diffusions. The main novelty of this work is that the posterior conditional distribution is over infinite dimensional sample paths, rather than a finite dimensional multivariate posterior as in standard Gaussian process inference. The algorithm as presented also covers multivariate systems and is derived in a continuous time framework. However, this approach is not new and the benefits of modelling the problem in continuous time first and then discretising have been established by Apte et al. (2007). Moreover, one difference with previous work on the same direction is that the new VGPA algorithm provides a natural way to estimate the model parameters.

So far, issues concerning discretisation schemes and initialisation of the variational parameters have not been discussed. As will shown in following chapters when solving the problem on a digital computer the continuous time framework must be discretised. The choice for the prior SDE Equation (4.1), is the simple Euler-Maruyama, although other schemes are also possible and their effect on the performance of the algorithm is still an open question. The initialisation of the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$ can be done in many ways. All the analytic derivations of the VGPA framework for the systems studied in this thesis are shown in Appendix D. Also, expressions that can initialise these parameters optimally are given. In practice, the univariate systems (OU and DW) could be initialised almost arbitrarily, showing good robustness. On the contrary, more care should be taken in the multivariate systems (L3D and L40D). Nevertheless, it is not yet clear whether this sensitivity of the algorithm in these systems is due to their higher dimensionality or their chaotic behaviour.

5 Radial basis function extension

CONTENTS

5.1 Foreword

This chapter derives and presents a new radial basis function framework that extends the variational Bayesian algorithm for approximate inference in diffusion processes, as discussed in Chapter 4. It is shown that the new radial basis function approximation based algorithm not only converges to the original VGPA algorithm, but also has beneficial characteristics when estimating (hyper-) parameters. The new approach is validated on three non-linear dynamical systems, namely the univariate stochastic double well (DW), and the multivariate Lorenz ’63 (L3D) and Lorenz ’96 (L40D). Results show that this new approach is able to recover good estimates of the system and noise parameters in the multivariate case, even for chaotic systems.

5.1.1 Chapter outline

Initially, the main characteristics and benefits of using RBFs are highlighted. This is followed by the main contribution of this chapter, which is the (global) approximation, of the aforementioned variational Bayesian inference algorithm (see Chapter 4 and also the references of Archambeau et al. (2007, 2008)), in terms of RBF expansion. For this purpose the new RBF approximation framework, for the general multidimensional case, will be derived and explained in detail. To validate this new approach a series of experiments have been performed. Results for state estimation are given in Section 5.4, and for (hyper-) parameter estimation in Section 5.5. The chapter concludes with a thorough discussion concerning implementation and other issues.

5.2 Radial basis function networks

Radial basis function networks are a class of artificial neural networks, that were introduced as an alternative to multi-layer perceptrons (MLP) (Bishop, 1995). They originate from techniques of performing interpolation on multivariate data, but their use can also be found in function approximation (Broomhead and Lowe, 1988), classification problems, time series prediction and so on. Two of the main features that make the use of RBF networks attractive are the simplicity of its architecture (usually only one layer of hidden units) and the fact that the activation of the hidden units is determined by the distance of the input vector from a prototype vector (also known as the “origin”). These two characteristics make the training methods used for RBF networks substantially faster than those required when training MLP networks (Bishop, 1995).

Typically a RBF network consists of three layers, as shown in Figure 5.1. The first is referred to as the input layer, the second is the hidden units (i.e. the basis functions) and the last is the output layer.

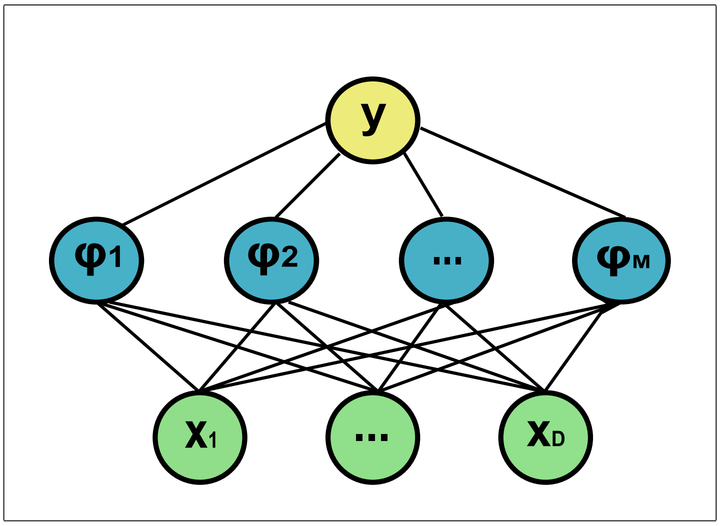


Figure 5.1: Typical architecture of a RBF network. The vector **x** is used as input to all RBFs, each with different parameters. The output of the network **y**, is a linear combination of the weighted radial basis functions outputs $\phi_{1:M}$.

In the context of function approximation, which is of primary interest, the approximation of a multidimensional function, $f(\mathbf{x}) : \Re^D \rightarrow \Re$, is performed by a set of D-dimensional basis functions which are defined as:

$$\phi_i(\mathbf{x}) = \phi_i(\|\mathbf{x} - \mathbf{c}_i\|) \,, \tag{5.1}$$

where $\|.\|$ denotes the Euclidean distance, $\phi_i(\mathbf{x}) : \Re^D \rightarrow \Re$, is the basis function, $c_i \in \Re^D$ is the i'th centroid and $i \in \mathcal{N}^*$.

Given this setting, the approximation of $f(\mathbf{x})$ is given by:

$$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) = \sum_{i=0}^{L} w_i \times \phi_i(\mathbf{x}) \,, \tag{5.2}$$

where $w_i \in \Re$ is the i'th weight and $L \in \mathcal{N}^*$ is the total number of basis functions. A common choice of basis functions in the literature, is the Gaussian or square exponential kernel, as defined in Equation (5.6), (Verleysen and Hlavackova, 1994; Benoudjit et al., 2002). However, depending on the specific problem other choices of basis functions have also been proposed, such as sigmoidal (Tsai et al., 1996).

Theoretical guidance on how many basis functions one needs to use, or which family of basis functions is the most appropriate, in the context of approximate inference for diffusion processes, have yet to be established, and some empirical results are presented later.

5.3 Global approximation of the variational parameters

The idea of approximating continuous (or discrete) functions by RBFs is far from new (Kurkova and Hlavackova, 1994). Here, the complexity of the original VGPA algorithm is controlled by using RBFs to approximate the time varying variational parameters ($\mathbf{A}_t$ and $\mathbf{b}_t$, see Eq. (4.7)). In the original variational framework, these functions are discretized with a small time discretisation step (e.g. $\delta t = 0.01$), resulting in a set of discrete time variables that need to be optimised during the process of minimising the free energy.

The size of that set (number of variables) scales proportional with the length of the time window of inference, the dimensionality of the data and the time discretisation step. In total one needs to infer:

$$N_{total} = (D+1) \times D \times |t_f - t_0| \times \delta t^{-1} \ , \tag{5.3}$$

variables, where D is the system dimension, t_0 and t_f are the initial and final times and the time step δt must be small for numerical stability.

By replacing the discretized time varying functions $\mathbf{A}_t$ and $\mathbf{b}_t$, with RBF expansions the following expressions are obtained:

$$\tilde{\mathbf{A}}_t = \sum_{i=0}^{L_A} \boldsymbol{A}_i \times \phi_i(t) \ \text{ and } \ \tilde{\mathbf{b}}_t = \sum_{i=0}^{L_b} \boldsymbol{b}_i \times \pi_i(t) \ , \tag{5.4}$$

where $\boldsymbol{A}_i \in \Re^{D\times D}$ and $\boldsymbol{b}_i \in \Re^D$ are the “weights”, $\phi_i(t)$, $\pi_i(t) : [0,\infty] \to \Re$ are fixed basis functions (regarded here as functions of time) and L_A, $L_b \in \mathcal{N}^*$, are the total number of RBFs considered. Therefore the new (approximate) expression for the $\mathcal{L}$agrangian becomes:

$$\tilde{\mathcal{L}} = \tilde{\mathcal{F}}(q(\mathbf{x}_t), \boldsymbol{\theta}, \boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top (\dot{\mathbf{m}}_t + \tilde{\mathbf{A}}_t \mathbf{m}_t - \tilde{\mathbf{b}}_t)\, dt - \int_{t_0}^{t_f} \mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t + \tilde{\mathbf{A}}_t \mathbf{S}_t + \mathbf{S}_t \tilde{\mathbf{A}}_t^\top - \boldsymbol{\Sigma})\}\, dt \ . \tag{5.5}$$

The number of basis functions for each term, along with their class, need not to be the same. However, in the absence of any general theory, or particular knowledge about the functions, an empirical approach is followed that suggests the same number of Gaussian basis functions (Verleysen and Hlavackova, 1994). Hence $L_{Ab} = L_A = L_b$ and $\phi_i(t) = \pi_i(t)$ where:

$$\phi_i(t) = \exp\left\{ -0.5 \left(\frac{\|t - c_i\|}{\lambda_i} \right)^2 \right\} \ , \tag{5.6}$$

with c_i and $\lambda_i \in \Re$ are the i'th centre and width respectively (which controls the smoothness of the function) and $\|.\|$ is the Euclidean norm. Having precomputed the basis function maps $\phi_i(t)$ $\forall$ i $\in \{0,1,2,\cdots,L_{Ab}\}$ and $\forall$ t $\in [t_0, t_f]$, as shown in Table 5.1, the optimisation problem reduces to calculating the weights of the basis functions, with:

$$L_{RBF} = (D+1) \times D \times (L_{Ab} + 1) \ , \tag{5.7}$$

parameters. Typically the expected number of the RBF weights is much smaller than the initial number of discrete time variables (i.e. $L_{RBF} \ll N_{total}$), thus making the optimisation problem smaller and more stable.

$\boldsymbol{T}=[t_0,t_f]$	t_0	t_1	$\cdots$	t_f
$\varphi_0 \mapsto$	1	1	$\cdots$	1
$\varphi_1 \mapsto$	$\phi_1(t_0)$	$\phi_1(t_1)$	$\cdots$	$\phi_1(t_f)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\varphi_{L_{Ab}} \mapsto$	$\phi_{L_{Ab}}(t_0)$	$\phi_{L_{Ab}}(t_1)$	$\cdots$	$\phi_{L_{Ab}}(t_f)$

Table 5.1: Example of $\mathbf{\Phi}(t)$ matrix, defined on $T=[t_0,t_f]$. Here T is discretised (i.e. $T=[t_0,\ t_0+\delta t,\ t_0+2\delta t,\dots,\ t_0+N\delta t=t_f]$), where N is the total number of discrete time units. Although the basis functions are defined in continuous time and can be evaluated at any time instant, in practice the entries of $\mathbf{\Phi}(t)$ matrix contain only the evaluations of the basis functions at the discrete time instants of set T.

As in the original VGPA algorithm the parameters are determined using a scaled conjugate gradient (SCG) optimisation algorithm, as detailed in Nabney (2002), to minimise the $\mathcal{L}$agrangian cost function, Eq. (5.5). To do that, the partial derivatives of the approximate $\mathcal{L}$agrangian are computed with respect to the weights of the approximating functions, as shown in Appendix B (i.e. $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$, $\forall$ i $\in[0,1,2,\dots,L_{Ab}]$):

$$\frac{\partial\tilde{\mathcal{L}}}{\partial\boldsymbol{A}_i} \text{ and } \frac{\partial\tilde{\mathcal{L}}}{\partial\boldsymbol{b}_i}\,. \tag{5.8}$$

Once these weights have been determined, the pre-computed basis functions are used and with simple matrix multiplications the approximated time-varying linear dynamical system as a continuous function of time is retrieved. Schematically, in matrix notation, this is:

$$\tilde{\mathbf{A}}_t \overset{reshape}{\longleftarrow} \begin{pmatrix} A_1(t) \\ A_2(t) \\ \vdots \\ A_{D^2}(t) \end{pmatrix} = \begin{pmatrix} A_{1,0} & A_{1,1} & \cdots & A_{1,L_{Ab}} \\ A_{2,0} & A_{2,1} & \cdots & A_{2,L_{Ab}} \\ \vdots & \vdots & \ddots & \vdots \\ A_{D^2,0} & A_{D^2,1} & \cdots & A_{D^2,L_{Ab}} \end{pmatrix} \times \begin{pmatrix} \phi_0(t) \\ \phi_1(t) \\ \vdots \\ \phi_{L_{Ab}}(t) \end{pmatrix}.$$

Here $A_{j,i}$ represents the j'th component of the $\boldsymbol{A}_i$'th weight. Effectively, the $\boldsymbol{A}_i$ weights have been reshaped in column vectors and packed all together in one matrix with dimensions $D^2\times(L_{Ab}+1)$.

For the $\tilde{\mathbf{b}}_t$ a similar procedure is followed, only here things are simpler because the $\boldsymbol{b}_i$ weights are already vectors, so there is no need to reshape them. Hence that yields:

$$\tilde{\mathbf{b}}_t \leftarrow \begin{pmatrix} b_1(t) \\ b_2(t) \\ \vdots \\ b_D(t) \end{pmatrix} = \begin{pmatrix} b_{1,0} & b_{1,1} & \cdots & b_{1,L_{Ab}} \\ b_{2,0} & b_{2,1} & \cdots & b_{2,L_{Ab}} \\ \vdots & \vdots & \ddots & \vdots \\ b_{D,0} & b_{D,1} & \cdots & b_{D,L_{Ab}} \end{pmatrix} \times \begin{pmatrix} \phi_0(t) \\ \phi_1(t) \\ \vdots \\ \phi_{L_{Ab}}(t) \end{pmatrix},$$

where $b_{j,i}$ represents the j'th component of the $\boldsymbol{b}_i$'th weight.

$System$	θ	$\mathbf{\Sigma}$	N_{obs}	$\mathbf{R}$
DW	1	0.8	2	0.04
$L3D$	$[10, 28, 2.6667]$	4	10	2
$L40D$	8	5	10	1

Table 5.2: Summary of the experimental setup.

In addition to the above re-parametrisation, of the initial algorithm, a modified Gram-Schmidt orthogonalisation is employed (Golub and van Loan, 1996), to improve numerical stability and the speed of convergence. This is done on the pre-computed ϕ_i vectors, as shown in Table 5.1. In practice this orthogonalisation dramatically reduces the number of iterations required for the algorithm to reach convergence. Here ϕ_i refers to the i'th basis function (row-vector) that contains the pre-computed values for all t $\in [t_0, t_f]$. Hence $\phi_i \in \Re^{1\times N}$, where $N = |t_0 - t_f|/\delta t$.

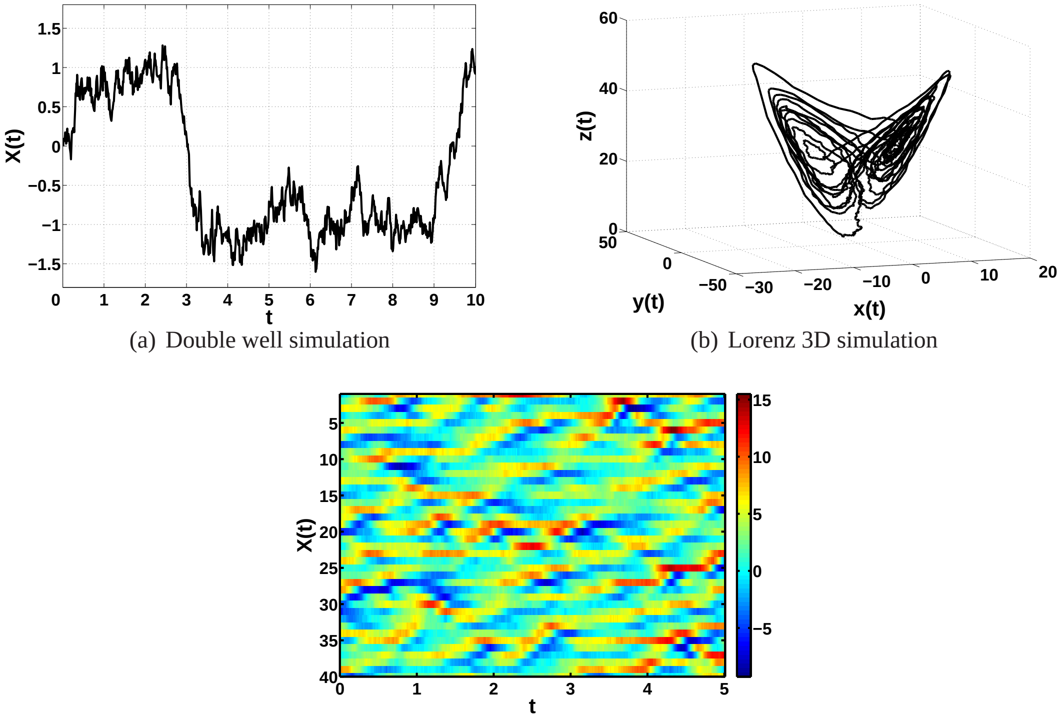
5.4 Results of state estimation

The experimental results, on both state and parameter estimation, will be presented in this section and the following one. To test the stability and the convergence properties of the new RBF approximation algorithm, the new approach is validated with three highly non-linear dynamical systems. These are the univariate double well system (DW), the three dimensional Lorenz (L3D) system and the forty dimensional Lorenz (L40D), (see Chapter 3).

Experimental setup

For the simulations here, a time window of ten units ($t_0 = 0$, $t_f = 10$) for the DW system is considered (Fig. 5.2(a)), twenty ($t_0 = 0$, $t_f = 20$) for the L3D (Fig. 5.2(b)), and five ($t_0 = 0$, $t_f = 5$) for the L40D (Fig. 5.2(c)). The original theoretical framework (see Chapter 4) addresses continuous time sample paths, however when solving the problem on a digital computer, one has at some point to discretise the equations. This is done with a relatively small time discretisation step (e.g. $\delta t = 0.01$), which is identical for both the SDEs, see Eq. (4.1) and the ODEs, Eq. (4.12) and Eq. (4.13). The discretisation scheme that was chosen for the SDEs is the Euler-Maruyama, and for the ODEs is the Euler method.

The *true* parameters, that generated the sample paths are summarised in Table 5.2. Note that in the multivariate systems the noise covariance matrix $\mathbf{\Sigma}$ and the noise on the observations $\mathbf{R}$ are diagonal matrices and N_{obs} represents the number of available i.i.d. observations *per time unit* (i.e. observation density). These need to be relatively high in the chaotic systems if the parameters are



(a) Double well simulation

(b) Lorenz 3D simulation

(c) Lorenz 40D simulation

Figure 5.2: (a) Sample path of a double well potential system, used in the experiments, with two (rather uncommon) transitions between the wells. (b) A typical trajectory of the L3D system. (c) All forty dimensions of the L40D, for the time period [0-5], with $\theta = 8$.

to be identified with any accuracy.

Finally, the basis functions that were used in all systems were Gaussian Eq. (5.6), with centres c_i chosen equally spaced within the time windows and widths λ_i sufficiently large to permit overlap of neighbouring basis functions (Haykin, 1999):

$$\lambda_i = \frac{\max(centre) - \min(centre)}{L_{Ab}} \, , \tag{5.9}$$

where $L_{Ab} > 0$, is the total number of centres.

Although there exists methods to optimise the locations of the centres, as well as the widths of the basis functions (Benoudjit et al., 2002), a uniform distribution of the centroids is suggested, with fixed widths which is a sufficiently *close to optimal* solution. In this work RBFs are not applied in a traditional way, such as fitting a response function to a set of data (observations), rather they are employed to create a basis function set in continuous time, resulting in a constraint on the available solutions of the approximating functions $\mathbf{A}_t$ and $\mathbf{b}_t$.

Results

Figure 5.3 compares the results obtained from the RBF approximation algorithm, on the DW system, with basis function density $M = 40$[1], to the outcomes of a Hybrid Monte Carlo (HMC) sample from the posterior process, using the true values for the drift and diffusion parameters, which provides a reference solution to the smoothing problem. Note that although the variance of the RBF approximation is slightly underestimated, the mean path matches the HMC results rather well and the times of the transitions between the two wells are tracked correctly. The only difference in the mean paths is located at the beginning of the time window. This is due to the fact that the RBF algorithm starts at a fixed point (i.e. $m(0) =$ fixed), rather than optimised.

The variational approximation as employed here is likely to underestimate the variance of the approximating process due to the expectation in the KL divergence being taken with respect to the approximating distribution in Eq. (4.4). Empirically this is found to have a relatively minor impact as long as the system is well observed, which keeps the true posterior process close to Gaussian. Where the true posterior process is strongly non-Gaussian, and in particular where it is multi-modal a more significant underestimation exists, as might be expected. The results shown here are typical examples, where the systems have uni-modal posteriors.

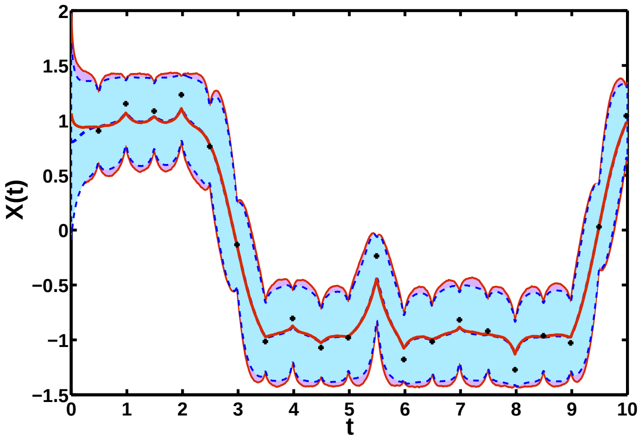


Figure 5.3: Comparison of the approximated marginal mean and variance (of a single DW realisation), between the “correct” HMC estimates (solid red lines) and the RBF variational algorithm (dashed blue lines). The crosses indicate the noisy observations.

To provide a robust demonstration of the consistency of the results of the RBF approximation, with respect to the original discretized VGPA, one hundred different realisations of the observation noise, from a single dataset, were used. Here the number of basis functions in the RBF was increased to explore convergence of the RBF to the original VGPA. Summary statistics from these experiments, on the DW system, concerning the convergence of the free energy obtained

[1] Here M denotes the density of the basis functions per time unit. Hence $L_{Ab} = |t_f - t_0| \times M$ and in this example $L_{Ab} = 400$.

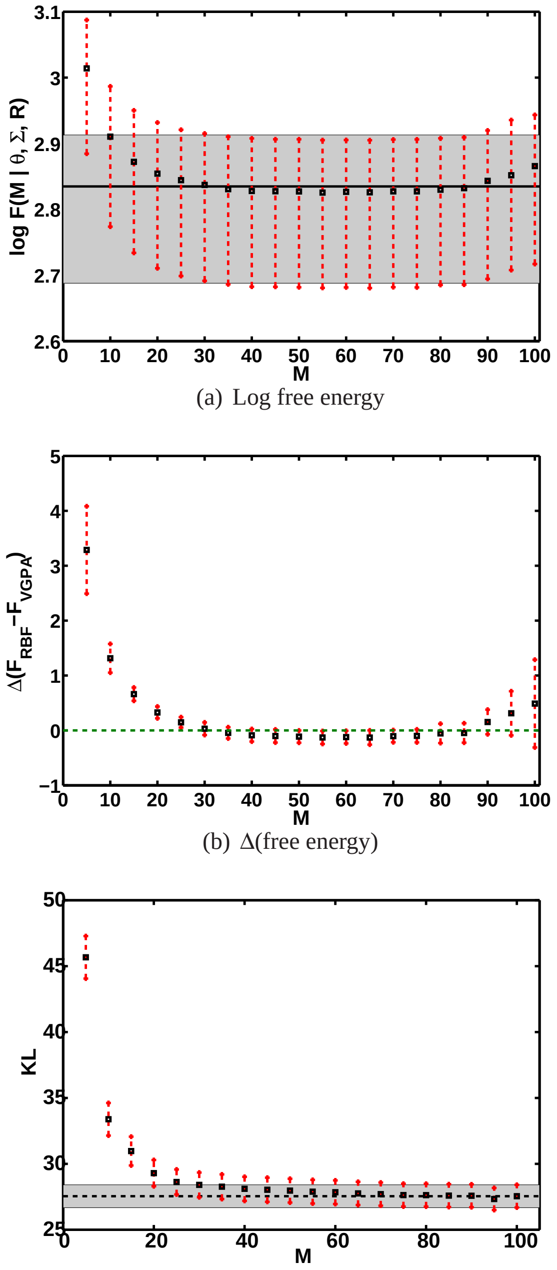
from the RBF approximation algorithm compared with the one from the original VGPA, shown in Figure 5.4(a). The 25, 50 and 75 percentiles from these 100 realisations are plotted when the system has converged to its minimum free energy. Is is apparent that with more than thirty five basis functions per time unit ($M = 35$), the RBF algorithm reaches the same free energy values as the original VGPA.

In addition to Figure 5.4(a), Figure 5.4(b) provides a similar summary, plotting the difference between the free energies of the RBF and the VGPA, as a function of basis functions density, clearly showing that for this system an RBF with 40 basis functions per time unit is sufficient to capture the variation with no detectable loss of information.

The new RBF approximation algorithm is extremely stable, when estimating the state of the systems concerned, and converges to the original VGPA, given a “sufficient” number of basis functions. This is also apparent in comparing the $\mathrm{KL}[p\|q]$ divergence (Kullback and Leibler, 1951), between the approximations q (VGPA, RBF) and the “true” (HMC) posterior p, as shown in Figure 5.4(c), on a typical realisation of the DW system. This KL divergence, which is integrated over the whole time window, is useful to measure the goodness of the RBF approximation which is clearly comparable to the original VGPA. The non-zero value of this divergence is related to the approximation error induced by the Gaussian process approximation to the non-Gaussian posterior distribution.

To address the sensitivity of the RBF approximation to the widths of the Gaussian basis functions and the effect of this on the convergence of the free energy a comparison of the original VGPA, against the RBF algorithm took place, for fixed basis function density equal to forty per time unit $M = 40$, while varying the width λ of the basis functions. This was again repeated for one hundred different realisations of the observation noise (of the DW system) and the summary results are shown in Figure 5.5(b). It is apparent that the performance of the RBF algorithm is very stable for a wide range of λ_i values (note the logarithmic scale on the x-axis). It is possible that this is an effect of the high number of equidistant basis functions that was chosen and which provide good coverage in the time domain. Repeating the same experiment with fewer ($M = 10$, Fig. 5.5(a)) basis functions shows very similar behaviour, although the RBF approximation is unable to match the original VGPA due to having insufficient basis functions. Again, the stable region is wide and flat and only when the value of the width is pushed to the extremes does the algorithm produce instabilities.

Figures 5.6(a), 5.6(b) and 5.6(c) compare the results obtained from the RBF approximation algorithm, on a twenty time unit inference window $T = [0, 20]$ of the L3D system for fixed basis function density, $M = 40$, against the “correct” posterior process obtained from a Hybrid Monte Carlo (HMC) method, on a single realization, given the true parameter setting, as shown on Ta-



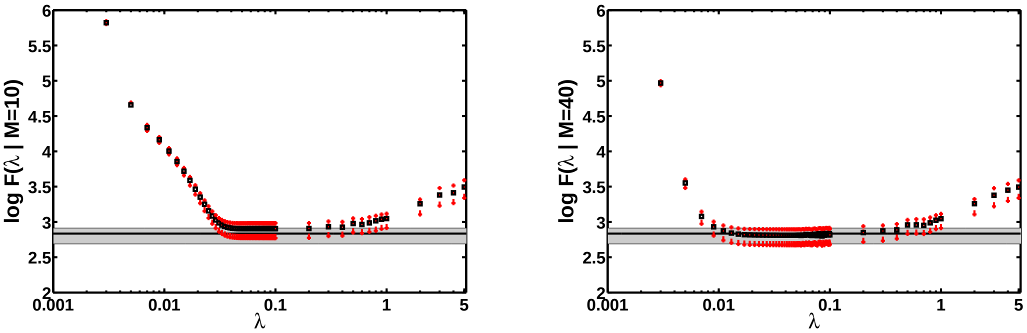
(a) Log free energy

(b) Δ(free energy)

(c) KL divergence

Figure 5.4: (a) Comparison of the log free energy, at convergence, between the RBF algorithm (squares, dashed lines) and the original VGPA (solid line, shaded area) on the DW system. The plot shows the 25, 50 and 75 percentiles (from 100 realisations) of the free energy. (b) The mean value (black squares) and the variance (red dashed vertical lines) of the difference of the free energy between the RBF approximation and the original VGPA, obtained from one hundred realisations of the observations noise. (c) Shows a similar plot (for a single realisation) for the integral of the $\mathrm{KL}[p\|q]$ divergence, between the "true" (HMC) and approximate VGPA (dashed line, shaded area) and RBF (squares, dashed lines) posteriors, over the whole time window $[t_0, t_f]$. All plots are presented as functions of basis function density.

ble 5.2. It is worth noticing that in this case obtaining results using HMC methods was non-trivial and required careful tuning and convergence assessment. More details about sampling with the HMC algorithm are given in Chapter 7.



(a) M = 10

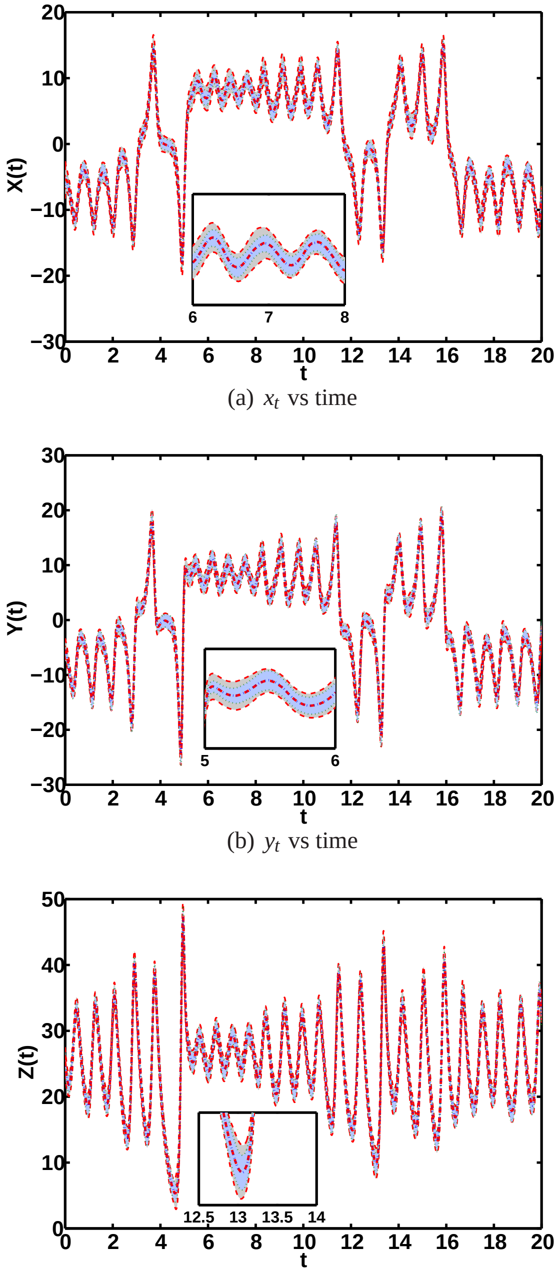
(b) M = 40

Figure 5.5: Comparison of the log free energy, at convergence, between the RBF algorithm (squares, dashed lines) and the original VGPA (solid line, shaded area) on the DW system as a function of the basis function width, λ. The plot shows the 25, 50 and 75 percentiles (from 100 realisations) of the free energy. **Left panel:** The value of the basis function density is fixed to ten per time unit ($M = 10$), whereas in the **Right panel:** the basis function density is increased to forty ($M = 40$) per time unit. For both experiments the parameters $\boldsymbol{\theta}$, $\boldsymbol{\Sigma}$ and $\mathbf{R}$ were identical and kept fixed to their true values. Note also the logarithmic scale on the -x- axis.

Similarly to the DW case, the marginal mean paths on each dimension of the system match the HMC results and the variance of the RBF approximation is again underestimated. However, after having a closer look to the experiments performed it was realised that the underestimation of the variance in higher dimensional systems is not the general case (as it will be seen in Chapter 6). Here this result is explained by the fact that the smoothing window for the HMC results was originally fifty time units, while the RBF approximation algorithm was performed only in the first twenty time units.

To provide robust results illustrating the convergence of the RBF approximation to the original discretized VGPA on this multivariate system thirty different realisations of the observation noise, from a single dataset, were used. Summary statistics from these experiments, on the L3D system, concerning the convergence of the free energy obtained from the RBF approximation algorithm compared with the one from the original VGPA, is shown in Figure 5.7. This again shows that the RBF version is relatively insensitive to the number of basis functions per time unit, above some threshold, and seems to actually produce slightly better estimates in terms of the free energy.

Finally, results are presented for the stochastic Lorenz 40D system. Figure 5.8(a) shows the approximated means for all forty dimensions of the system for a relatively short time window. To obtain these results forty basis functions per time unit were used. Figure 5.8(b) shows the marginal variances for each dimension and Figure 5.8(c) plots the squared difference of the approximated means with the true sample path (see Figure 5.2(c)) showing that for the most part a good estimate of the mean state is produced.



(a) x_t vs time

(b) y_t vs time

(c) z_t vs time

Figure 5.6: A comparison of the approximated marginal means and variances (of a single realisation of the L3D system), between the "correct" HMC estimates (dashed red lines) and the variational RBF ($M = 40$) algorithm (dotted blue lines). The results are plotted separately on each dimension. The noisy observations have been omitted for better illustration of the marginal means. The zoomed sub-plots highlight the underestimation of the variance.

5.5 Results of parameter estimation

This section presents the results for the estimation of (hyper-) parameters of the systems considered, following the same experimental setup as in Section 5.4. The original VGPA approximation can be used to compute a bound on the marginal likelihood Eq. (4.6) and thus compute estimates of (hyper-) parameters, including the system noise and the drift parameters (see Section 4.5). Once

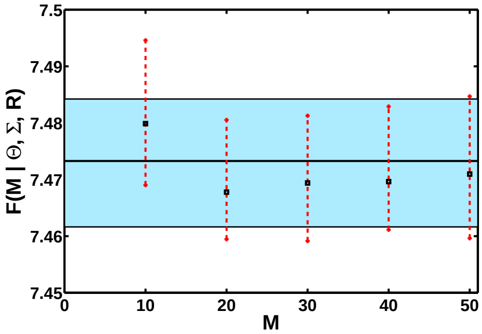
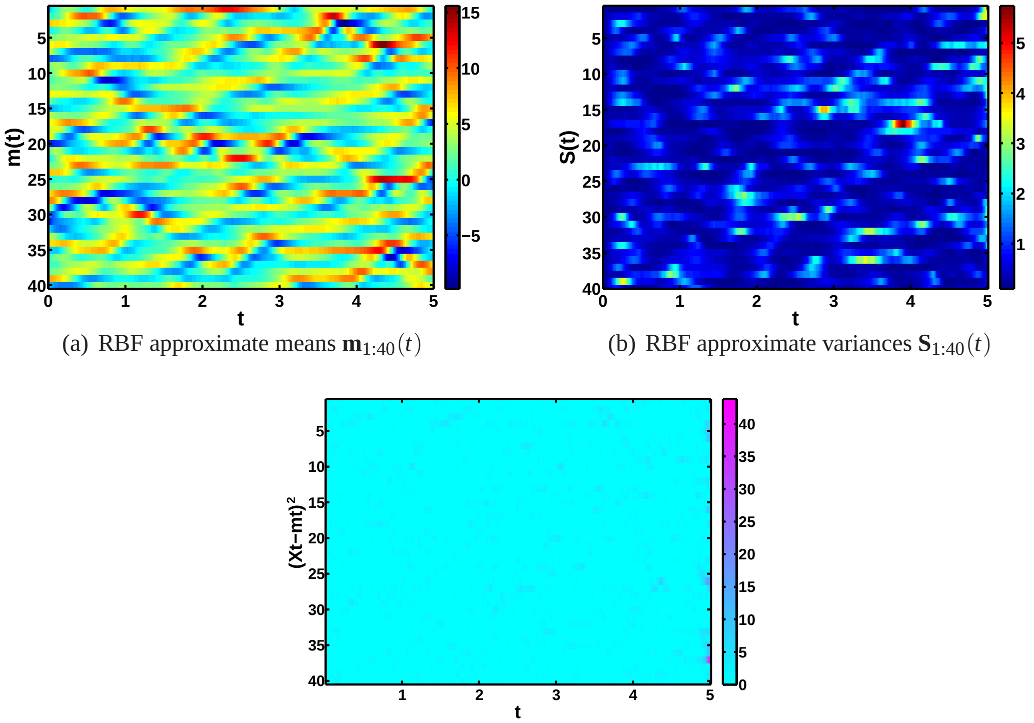


Figure 5.7: Comparison of the log free energy, at convergence, between the RBF algorithm (squares, dashed lines) and the original VGPA (solid line, shaded area) on the L3D system. The plot shows the 25, 50 and 75 percentiles (from 30 different realisations) of the free energy. The log free energy is plotted as a function of basis function densities.



(a) RBF approximate means $\mathbf{m}_{1:40}(t)$

(b) RBF approximate variances $\mathbf{S}_{1:40}(t)$

(c) Squared difference $(\mathbf{x}_t - \mathbf{m}_t)^2$

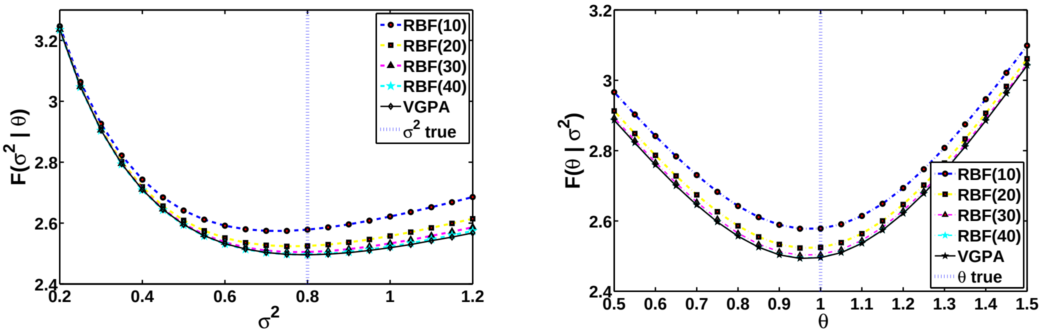
Figure 5.8: Fig. (a) shows all the approximated mean paths (of a single realisation of the L40D system), obtained from the variational RBF ($M = 40$) algorithm. In Fig. (b), the marginal variance around each mean path. Fig. (c) illustrates the squared difference of the approximated means with the true sample path (see Fig. 5.2(c)).

an upper bound has been achieved, one can attempt to estimate the (hyper-) parameters by computing the derivatives of the $\mathcal{L}$agrangian (see Eq. 4.14), with respect to the drift and diffusion parameters:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \text{ and } \frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}}, \tag{5.10}$$

and then employ a conjugate gradient optimisation algorithm (Nabney, 2002).

In the RBF version this is also possible and empirical results show that, at least for the univariate case, this is faster and more robust compared to the original VGPA. The profile log approximate marginal likelihood, for the DW system is shown in Figures 5.9(a) and 5.9(b). Even with a relative small basis function density, the σ^2 and θ minima are very close to those determined by the VGPA. For σ^2 around thirty basis functions ($M = 30$), are needed to reach the same minimum. For the drift (θ) parameter the minimum is almost identical using only ten basis functions ($M = 10$) per time unit. Thus if the primary interest is in the drift parameters one can employ a relatively compact RBF approximation, which provides speed and robustness benefits and still obtain good estimates for the parameters.



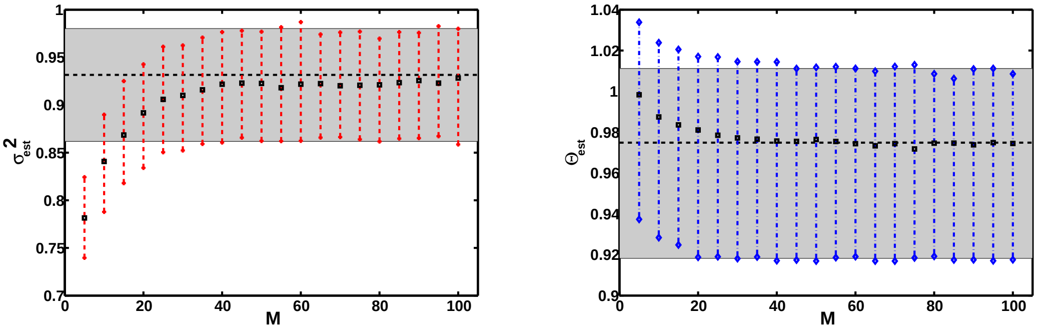
(a) σ^2 profile

(b) θ profile

Figure 5.9: **DW system:** (a) Profile marginal log likelihood for the system noise coefficient σ^2 keeping the drift parameter θ fixed to the true one. (b) as (a) but for the drift parameter θ keeping σ^2 fixed to its true value. Both simulations run for $M = [10, 20, 30, 40]$, basis functions per time unit and compared with the profiles from the VGPA on a typical realisation of the observations. The dotted vertical lines represent the true values of the parameters that generated the data.

The above conclusions are supported by further experiments, on one hundred different realisations of the observation noise on the same dataset. Figures 5.10(a) and 5.10(b), exhibit consistency in the estimates of the maximum marginal likelihood parameters both in value and variability. In addition, the biases that exist in both estimates are shown more clearly and are consistent with the relatively sparse noisy measurements. For the drift parameter Fig. 5.10(b), the bias is relatively small, whereas for the system noise Fig. 5.10(a) it is larger. This could be explained by the fact that in the example considered here there exist two transitions between the two stable states, which for a short time window is rather unlikely, and thus suggests a higher noise variance than is really present.

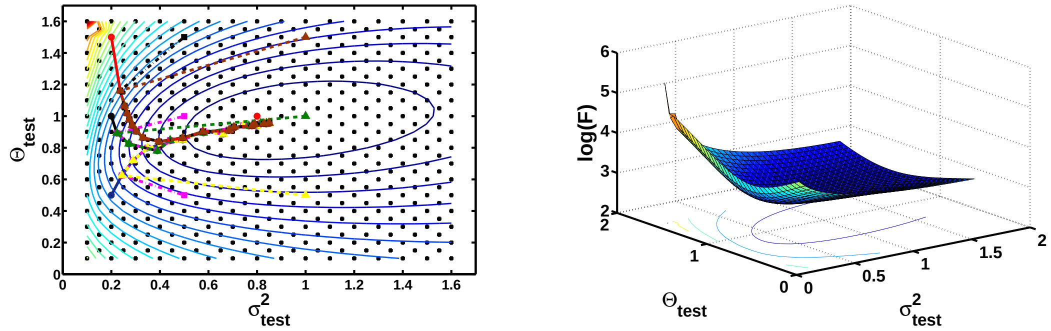
Apart from the conditional estimation results, a series of joint estimation of the parameters was also performed. These start from nine different points on a two dimensional grid, spanning the effective parameter space. Figure 5.11(a) shows a contour plot representing the logarithm of the free energy, and the nine different trajectories of the joint parameter estimation processes. This shows robust behaviour with all trajectories converging to a good approximation, close to the true parameter values, regardless of initialisation.



(a) σ^2 estimation

(b) θ estimation

Figure 5.10: **DW system:** (a) Conditional estimation of the system noise coefficient σ^2 keeping θ to its true value. The comparison is between the results from the RBF algorithm (squares, dashed vertical lines) and the original VGPA (horizontal dashed line, shaded area). The figure shows the 25, 50 and 75 percentiles of the estimated values (from 100 different realisations). (b) as (a) but for the drift parameter θ keeping σ^2 fixed to its true value. Both plots are presented as functions of increasing basis function density.



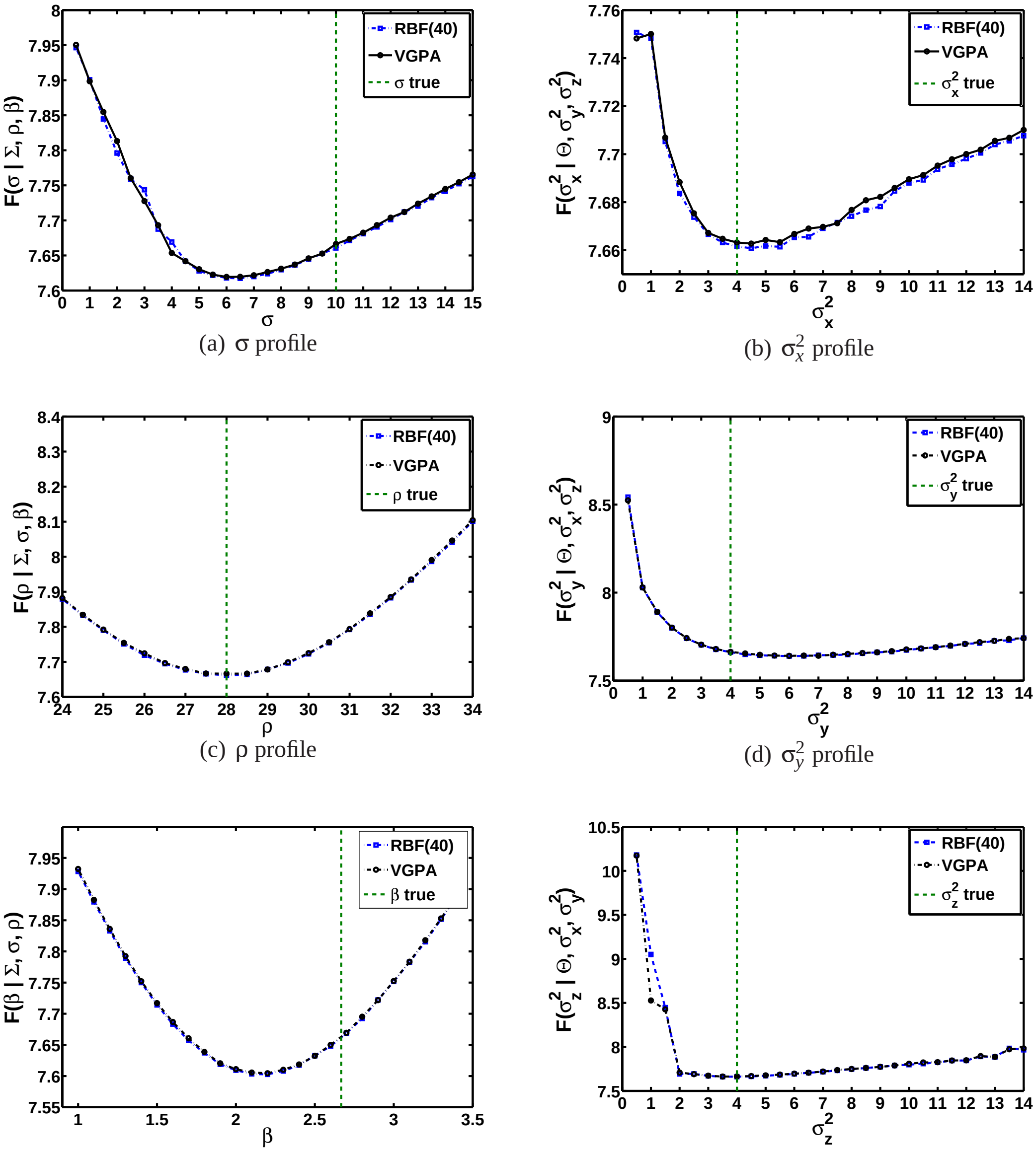
(a) Joint estimation (σ^2 and θ)

(b) Energy profile

Figure 5.11: **DW system:** Contour plot (a) presents the trajectories of nine joint estimations of both σ^2 and θ, from different starting points. Results obtained with $M = 40$ basis function density. (b) shows the log energy profile, in the parameter space.

Results on parameter inference for the Lorenz 3D system are shown in Figure 5.12. All subfigures show clearly that the RBF and original VGPA produce consistent results. The system noise parameters are well identified for the x_t variable Fig. 5.12(b), but not so well identified for the other two variables y_t and z_t, Figures 5.12(d) and 5.12(f) respectively. This is related to the dynamics of the system since the y_t and z_t components both have more complex interaction terms in their evolution equations. Estimating system noise parameters from sparse discrete time, noisy observations remains a significant practical challenge for all parameter inference methods.

Results for drift parameter inference for the Lorenz 3D system show that there is a bias in the estimates for the σ and β parameters, Figures 5.12(a) and 5.12(e) respectively. The source of this bias is not clear, and further work is necessary to investigate whether this is related to systematic error in the variational method, or a more general problem for likelihood based inference in such chaotic dynamical systems. The ρ parameter, as shown in Fig. 5.12(c), is well estimated. It



(a) σ profile

(b) σ_x^2 profile

(c) ρ profile

(d) σ_y^2 profile

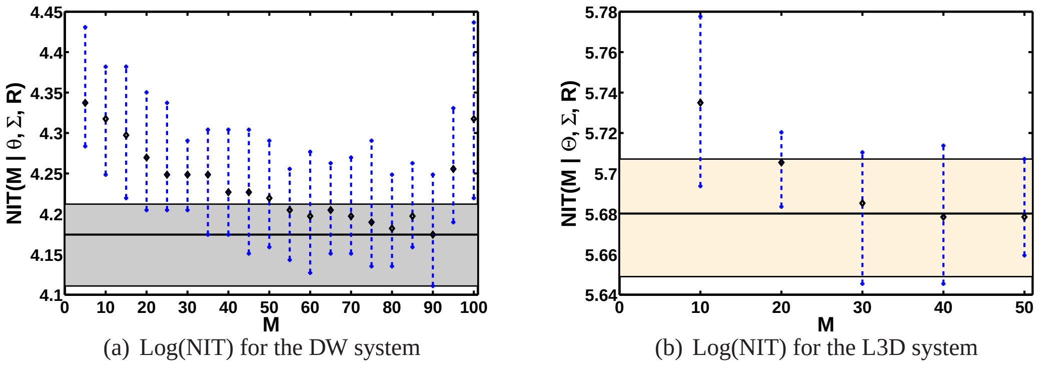
(e) β profile

(f) σ_z^2 profile

Figure 5.12: **L3D system:** Profile approximate marginal log likelihoods for the drift and diffusion parameters $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ (diagonal elements). Each profile is obtained by keeping all the other parameters fixed to their true values. Left column presents the profiles for the drift σ (a), ρ (c) and β (e), while the right column for the system noise variance on each dimension σ_x^2 (b), σ_y^2 (d) and σ_z^2 (f). All simulations run with $M = 40$, basis functions per time unit and are compared with the profiles from the VGPA on a typical realisation of the observations. The dotted vertical lines represent the true values of the parameters that generated the data.

should be stressed that obtaining such profile plots is computationally intensive since it requires minimisation of the free energy for relative long time windows (20 time units for these plots) at a range of parameter settings.

The reduction in the complexity of the algorithm, does not produce a similar reduction in computational time. Figure 5.13(a) compares the log number of iterations of the RBF algorithm needed to reach convergence with the number of iterations from the VGPA. These results are



(a) Log(NIT) for the DW system

(b) Log(NIT) for the L3D system

Figure 5.13: (a) Comparison of the log number of iterations to reach convergence, between the RBF algorithm (diamonds, dashed vertical lines) and the original VGPA (solid horizontal line, shaded area) on the DW system. The plot shows the 25, 50 and 75 percentiles (from 100 realisations). (b) same as (a), but for the Lorenz 3D system from 30 different realizations of the observation noise. Both plots are presented as functions of RBF density.

summaries from 100 different realizations (of the observation noise on a single dataset) of the DW system and one can clearly see that the VGPA, while optimising a larger number of parameters, converges in fewer iterations. Figure 5.13(b) presents similar results but from 30 realisations of the Lorenz 3D system, where the two algorithms are more comparable.

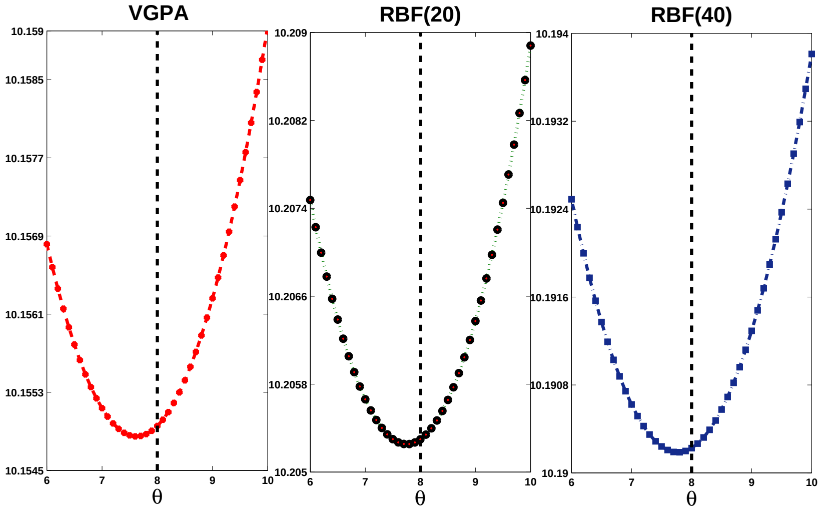


Figure 5.14: **Left panel:** Profile approximate marginal log likelihood, obtained with the VGPA algorithm, for the force (drift) parameter, from a single realisation of the L40D system, keeping the system noise covariance matrix $\boldsymbol{\Sigma}$, to its true value. **Central panel:** Same profile but obtained from the RBF approximation with basis function density twenty per time unit. **Right panel:** Same as central, but with increased basis function density to forty per time unit. In all panels the vertical dashed line represents the true value of the forcing parameter.

Results of parameter inference for the 40 dimensional Lorenz system are shown in Figure 5.14. Here the results show consistency for the RBF approximation and that this is relatively insensitive to RBF density. In all cases the minimum for the profiles is well defined and close to the true value used to generate the trajectory and thus observations. These new results show that both the VGPA and the variational RBF approximation can be applied to relatively high dimensional dynamical systems and can provide reliable estimates of (hyper-) parameters in these dynamical systems.

5.6 Discussion

This chapter presented a new radial basis function approximation that extends the variational Gaussian process approximation (VGPA) algorithm for Bayesian inference for diffusion processes. The new method, is validated by numerical experiments on three non-linear systems. Results show that the new algorithm converges to the original VGPA with a relatively small number of basis functions per time unit. However, it was not possible to provide a principled way to determine this value, and thus theoretically examine the sensitivity of the RBF approximation to this modelling choice.

Different systems *suggested* different optimum values, but in most of the cases less than 40% of the number of parameters required in the original VGPA had to be optimised. This makes the algorithm more stable, however the computation benefits are not as significant as had been hoped, which is related to the more complex (non-linear) optimisation problem when using weights on RBFs as the control parameters in the variational optimisation process.

The algorithms were extended to higher dimensional systems and shown to provide good state and parameter estimates even in a forty dimensional dynamical system. In obtaining these results several numerical challenges related to the sensitivity of the original VGPA to good initialisation were encountered. In the 1D case the RBF version was extremely robust and could be initialised almost arbitrarily, however in the higher dimensional cases some care was required to initialise all the algorithms.

Although the new algorithm is stable with fewer parameters, that was not reflected in a similar reduction in the computational time. This may be related to the fact that the class of basis functions that was chosen (i.e. Gaussian) is not suitable to approximate the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$.

Another argument is that the RBF algorithm still works in a discrete time framework, albeit with an alternative parametrisation. In the original VGPA the control parameters are discretized, with a relatively small time step for numerical stability (e.g. $\delta t = 0.01$). This discretisation is also inherited in the implementation of the RBF version of the algorithm and even though all the time varying basis function maps can be pre-computed off-line, there are still bounds by the limitations of each discretisation scheme.

The experiments have also highlighted issues around initialisation and the computation of the expectations (see Appendix A), required in the free energy. At present two options have been employed: a (*universal*) numerical approximate approach based on the unscented transformation (Julier et al., 2000) and the (*system dependent*) exact analytic calculation of the required moments (Appendix D). Neither is particularly satisfactory – the unscented transformation requires careful tuning to ensure stability and the analytic derivation is time consuming, especially with high dimensional systems and potentially error prone in implementation, although it can be partially automated using symbolic manipulation.

6 Local polynomial extension

CONTENTS

6.1 Foreword

The current chapter proposes another, alternative, re-parametrisation to the previously described VGPA algorithm (Chapter 4), in terms of polynomial approximations. The linear drift $\mathbf{g}_L(\mathbf{x}_t)$ in Eq. (4.7) is defined in terms of $\mathbf{A}_t$ and $\mathbf{b}_t$. These functions, upon discretisation, result in a finite set of discrete time variables that need to be inferred during the optimisation procedure.

In Chapter 5, these time varying functions were approximated with basis function expansions with support over the whole time domain (i.e. $T = [t_0, t_f]$). This allowed a reduction in the total number of control variables in the optimisation step, as well as some prior control over the space of functions admitted as solutions. However, the $\mathbf{A}_t$ and $\mathbf{b}_t$ variational parameters are by construction *discontinuous* at observation times. Thus a large number of basis functions was required to capture the *roughness* at observation times (see Fig. 6.2).

In the same spirit, the solution proposed here to overcome this issue is to define the approximations only between observation times such as, $[t_0, t_{k1}], (t_{k1}, t_{k2}], \ldots, (t_{kM}, t_f]$. This way a function approximation can be defined on each sub-interval (without overlap) and further reduces the total number of parameters to be optimised. Although simple in concept, this approach is shown to be very robust and able to recover good estimates of the states and (hyper-) parameters on the systems tested.

6.1.1 Chapter outline

The chapter begins with Section 6.2, where the new suggested polynomial extension is explained in detail for the general multivariate case. The new approach is tested on artificial data generated from a variety of systems, as described in Chapter 3. The experimental setup is given in Section 6.3.1, followed by results on state and parameter estimation in Sections 6.3.2 and 6.3.3, respectively. The Lorenz '96 system, due to its relatively higher dimensionality, comparing to the other systems tested here, is treated as a special case in Section 6.3.4. The chapter ends with a discussion.

6.2 Polynomial approximation of the variational parameters

The variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$ are represented by a finite set of discrete time variables. The size of that set scales proportionally to the length of the time window of inference, the dimensionality of the data (state vector $\mathbf{x}_t$) and the time discretisation step, as defined in Eq. (5.3).

Substituting $\mathbf{A}_t$ and $\mathbf{b}_t$ with polynomials, defined locally on each sub-interval, the following expressions are obtained:

$$\tilde{\mathbf{A}}_t^j = \boldsymbol{A}_0^j + \boldsymbol{A}_1^j \times t + \cdots + \boldsymbol{A}_{Mo}^j \times t^{Mo} \,, \tag{6.1}$$

$$\tilde{\mathbf{b}}_t^j = \boldsymbol{b}_0^j + \boldsymbol{b}_1^j \times t + \cdots + \boldsymbol{b}_{Mo}^j \times t^{Mo} \,, \tag{6.2}$$

where $\tilde{\mathbf{A}}_t^j$ and $\tilde{\mathbf{b}}_t^j$ are the approximating functions defined on the j'th sub-interval, $\boldsymbol{A}_i^j \in \Re^{D\times D}$ and $\boldsymbol{b}_i^j \in \Re^D$ are the i'th order coefficients of the j'th polynomial and $i \in \{0, 1, \ldots, Mo\}$, with Mo representing the order of the polynomial.

It is important to distinguish from the case where the polynomials are fitted between the actual *measurable values* (e.g. interpolation with cubic splines). Here they are rather fitted between *observation times*. Note also that the order of the polynomials between $\tilde{\mathbf{A}}_t^j$ and $\tilde{\mathbf{b}}_t^j$, or even between the j'th polynomial of each approximation, need not to be the same; nevertheless in the absence of any additional information about the functions, or lack of any theoretical guidance, a practical approach is taken to suggest the same order of polynomials, under the condition that they provide sufficient flexibility to capture the *discontinuity* of the variational parameters at observation times, as shown in Figure 6.1.

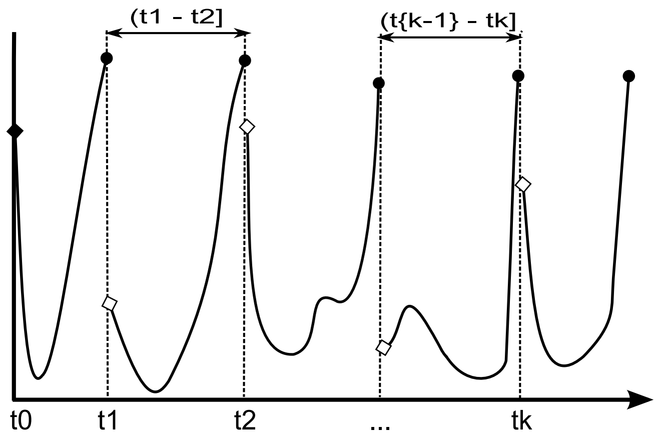


Figure 6.1: An example of the *local* polynomial approximation, on a univariate system. The vertical dashed lines represent the times the observations occur and each polynomial is defined *locally* between two observation times. The filled diamond and circles indicate closed sets, while the clear diamonds define open sets. Note that only the first polynomial is defined in closed set from both sides, to avoid overlapping.

The new expression for the $\mathcal{L}$agrangian (see Equation 4.14), for the j'th sub-interval thus becomes:

$$\tilde{\mathcal{L}}^j = \tilde{\mathcal{F}}^j(q(\mathbf{x}_t), \boldsymbol{\theta}, \boldsymbol{\Sigma}) - \int_{t\in T^j} \Big(\boldsymbol{\lambda}_t^\top (\dot{\mathbf{m}}_t + \tilde{\mathbf{A}}_t^j \mathbf{m}_t - \tilde{\mathbf{b}}_t^j) + \mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t + \tilde{\mathbf{A}}_t^j \mathbf{S}_t + \mathbf{S}_t \tilde{\mathbf{A}}_t^{j\top} - \boldsymbol{\Sigma})\}\Big)\, dt \,, \tag{6.3}$$

where $T^j \subset T$, or $T = \{T^1 \cup \cdots \cup T^j \cup \cdots \cup T^J\}$, with $J \geq 1$, being the total number of disjoint sub-sets.

The expressions for the polynomial approximations, Eq. (6.1 and 6.2), can be presented more compactly using matrix notation, which simplifies the presentation and is used from this point forward:

$$\tilde{\mathbf{A}}_t^j = \boldsymbol{A}^j \times \boldsymbol{p}^j(t) \text{ and } \tilde{\mathbf{b}}_t^j = \boldsymbol{B}^j \times \boldsymbol{p}^j(t)\ . \tag{6.4}$$

Schematically these matrix - vector products can be seen as:

$$\tilde{\mathbf{A}}_t^j \overset{\text{reshape to}}{\leftarrow} \begin{pmatrix} A_1^j(t) \\ A_2^j(t) \\ \vdots \\ A_{D^2}^j(t) \end{pmatrix} = \underbrace{\begin{pmatrix} A_{1,0}^j & A_{1,1}^j & \cdots & A_{1,Mo}^j \\ A_{2,0}^j & A_{2,1}^j & \cdots & A_{2,Mo}^j \\ \vdots & \vdots & \ddots & \vdots \\ A_{D^2,0}^j & A_{D^2,1}^j & \cdots & A_{D^2,Mo}^j \end{pmatrix}}_{\boldsymbol{A}^j} \times \underbrace{\begin{pmatrix} 1 \\ t \\ \vdots \\ t^{Mo} \end{pmatrix}}_{\boldsymbol{p}^j(t)} .$$

Here $A_{r,i}^j$ represents the r'th (scalar) component of the $\boldsymbol{A}_i^j$ coefficient in the j'th sub-interval. Effectively, the $\boldsymbol{A}_i^j$ weights have been reshaped in column vectors and packed together in one matrix of size $D^2 \times (Mo+1)$, (similar to the RBF case). For the $\tilde{\mathbf{b}}_t^j$ a similar procedure is followed, only here things are simpler because the $\boldsymbol{b}_i^j$ coefficients are already vectors, so there is no need to reshape them. That yields:

$$\tilde{\mathbf{b}}_t^j \leftarrow \begin{pmatrix} b_1^j(t) \\ b_2^j(t) \\ \vdots \\ b_D^j(t) \end{pmatrix} = \underbrace{\begin{pmatrix} b_{1,0}^j & b_{1,1}^j & \cdots & b_{1,Mo}^j \\ b_{2,0}^j & b_{2,1}^j & \cdots & b_{2,Mo}^j \\ \vdots & \vdots & \ddots & \vdots \\ b_{D,0}^j & b_{D,1}^j & \cdots & b_{D,Mo}^j \end{pmatrix}}_{\boldsymbol{B}^j} \times \underbrace{\begin{pmatrix} 1 \\ t \\ \vdots \\ t^{Mo} \end{pmatrix}}_{\boldsymbol{p}^j(t)} ,$$

where $b_{r,i}^j$ represents the r'th (scalar) component of the $\boldsymbol{b}_i^j$ (vector) coefficient.

Equation (6.4) shows that the vectors $\boldsymbol{p}^j(t)$ can be precomputed off-line for all predefined discrete time domains, reducing the computational complexity of estimating the coefficients of the polynomials. $\boldsymbol{p}^j(t)$ is precomputed and stored column-wise in a matrix, as shown on Table 6.1. Thus the reconstruction of the approximate variational parameters $\tilde{\mathbf{A}}_t^j$ and $\tilde{\mathbf{b}}_t^j$, for their whole time domain, can be done by a simple matrix - matrix multiplication, such as $\tilde{\mathbf{A}}_t^j = \boldsymbol{A}^j \times \boldsymbol{\Pi}^j(t)$, where the matrix $\boldsymbol{\Pi}^j(t)$, is defined as on Table 6.1.

The number of coefficients for both variational parameters $\tilde{\mathbf{A}}_t$ and $\tilde{\mathbf{b}}_t$ is:

$$L_{poly} = (D+1) \times D \times (Mo+1) \times J\ , \tag{6.5}$$

variables, where D is the system dimension, Mo is the order of the polynomials and J is the total number of disjoint sub-intervals (i.e. the number of observation times increased by one). Usually, it is anticipated that $L_{poly} \ll N_{total}$, thus making the optimisation problem smaller.

$$\mathbf{\Pi}^j(t) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ t_{k+\delta t} & t_{k+2\delta t} & t_{k+3\delta t} & \cdots & t_{k+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{k+\delta t}^{Mo} & t_{k+2\delta t}^{Mo} & t_{k+3\delta t}^{Mo} & \cdots & t_{k+1}^{Mo} \end{pmatrix}$$

Table 6.1: Example of $\mathbf{\Pi}^j(t)$ matrix, defined on $\boldsymbol{T}^j = (t_k, t_{k+1}]$. Note that because the time interval is discretised and defined on an open set from the left side, the first point of evaluation is $t_{k+\delta t}$, instead of t_k.

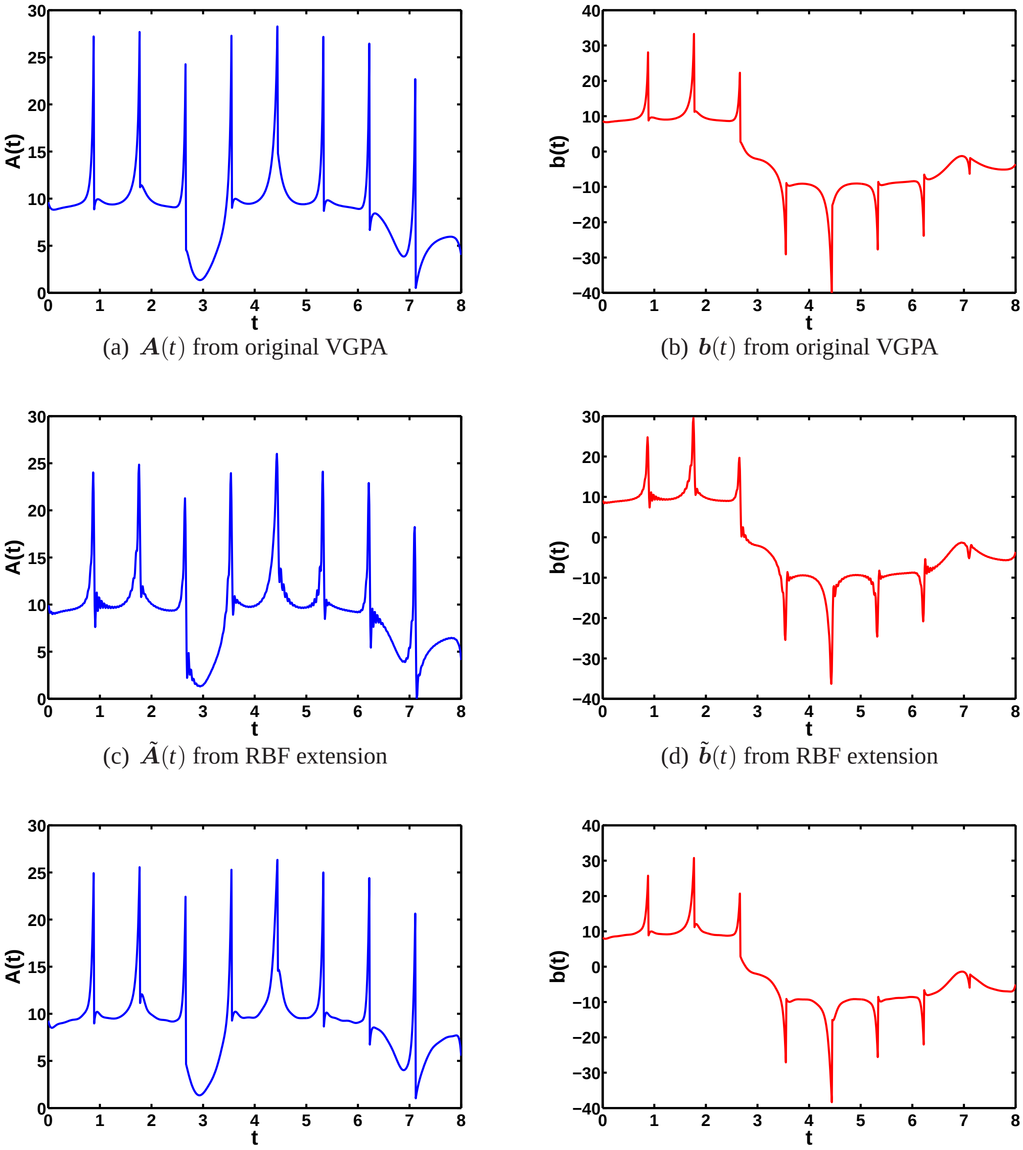
The original VGPA algorithm, used a scaled conjugate gradient (SCG) algorithm (Nabney, 2002), to minimize Eq. (4.14) with respect to the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$. The same procedure is used here computing the gradients of the approximate Lagrangian Eq. (6.3), with respect to the coefficients $\boldsymbol{A}^j$ and $\boldsymbol{B}^j$, of the re-parametrized variational parameters, for each sub-interval (details can be found in Appendix C). As in the RBF extension (Chapter 5), to further improve computational efficiency and stability a modified Gram-Schmidt orthogonalisation is applied (Golub and van Loan, 1996), to the rows of the pre-computed $\mathbf{\Pi}^j(t)$ matrices, as shown in Table 6.1, on each sub-interval separately. In practice this orthogonalisation dramatically reduces the number of iterations required for the algorithm to reach convergence.

Figure 6.2, demonstrates how the new proposed methodology better approximates the linear and offset parameters ($\mathbf{A}_t$ and $\mathbf{b}_t$), of the original VGPA (Figures 6.2(a) and 6.2(b)), compared to the RBF extension as presented in the previous chapter. The large number of basis functions that are used for this example (forty per time unit) makes the approximations to worsen causing fluctuations close to the observation times (Figures 6.2(c) and 6.2(d)). The new polynomial scheme suggested here, thanks to the locality of the approximation (there is no overlapping between the polynomials), achieves a better fit of the original parameters (Figures 6.2(e) and 6.2(f)), producing smoother results. Notice that although the RBF extension uses forty basis function per time unit, the new polynomial extension, with 9'th order polynomials can approximate the variational parameters better close to observation times, compared to the original VGPA results.

The proposed solution has an additional advantage over the original VGPA algorithm in that when solving the ODEs for the marginal means and covariances of the approximate Gaussian process Eq. (4.12 and 4.13) one can apply high order solvers, such as Runge-Kutta 2nd/4th order schemes by using the *exact* mid-points of $\tilde{\mathbf{A}}_t$ and $\tilde{\mathbf{b}}_t$, computed through the polynomial functions, i.e. evaluating

$$\tilde{\boldsymbol{A}}^j(t+0.5\delta t) = \boldsymbol{A}^j \times \boldsymbol{p}^j(t+0.5\delta t) \text{ and } \tilde{\boldsymbol{b}}^j(t+0.5\delta t) = \boldsymbol{B}^j \times \boldsymbol{p}^j(t+0.5\delta t),$$

rather than approximating them. In Figure 6.3, when the time discretisation step is relatively small



(a) $\boldsymbol{A}(t)$ from original VGPA

(b) $\boldsymbol{b}(t)$ from original VGPA

(c) $\tilde{\boldsymbol{A}}(t)$ from RBF extension

(d) $\tilde{\boldsymbol{b}}(t)$ from RBF extension

(e) $\tilde{\boldsymbol{A}}(t)$ from LP extension

(f) $\tilde{\boldsymbol{b}}(t)$ from LP extension

Figure 6.2: Left column presents the variational linear parameter $\mathbf{A}_t$, from a DW simulation with one observation per time unit on a $T = [0, 8]$ time window, for the original VGPA algorithm (a) and the new RBF (c) and polynomial (e) extensions. Right column presents similar results for the bias parameter $\mathbf{b}_t$.

(e.g. $\delta t = 0.001$), both the VGPA and LP[1] provide similar profile free energy results. The profiles show the value of the free energy at algorithm convergence as a function of the drift parameter value, for a fixed diffusion variance and are used later to demonstrate parameter estimation where they are explained in more detail. When the time step increases the new LP approximation remains smoother thus making the minimum clearer.

[1] LP from here onwards is a shorthand notation for the new local polynomial extension.

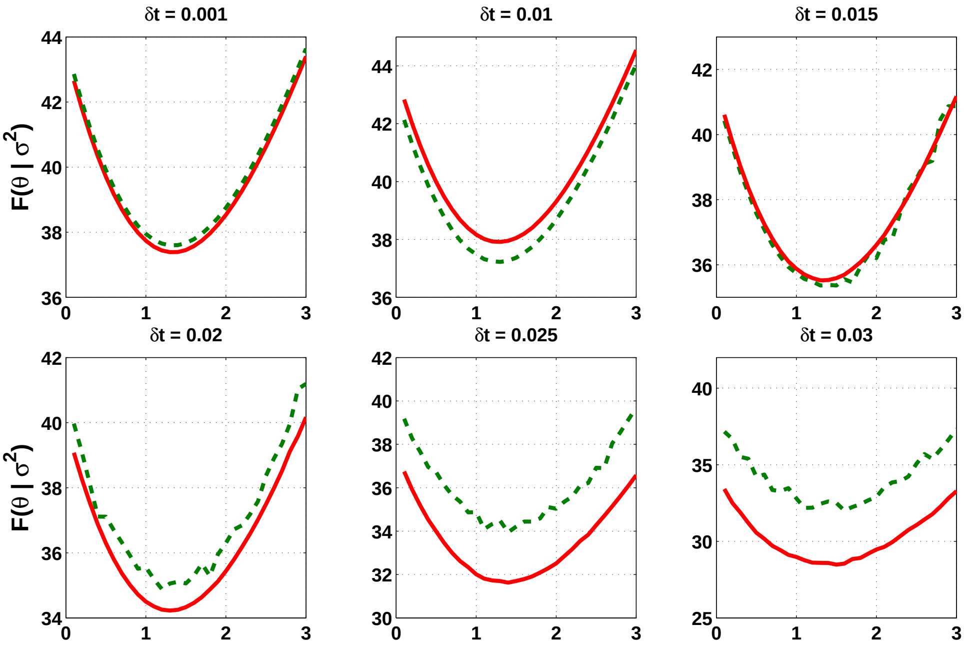


Figure 6.3: Marginal profiles of the variational free energy, at convergence, as a function of the drift parameter θ, given the system noise σ^2. The continuous line represents the profile from the LP approximation of second order (i.e. $Mo = 2$), while the dashed line represents the same profile but with the original VGPA framework. The results are from a single realisation of the OU process (see Chapter 3) and both algorithms use the Runge-Kutta 2nd order integration method. The time discretisation step ranges from $\delta t = 0.001$ (top left), to $\delta t = 0.03$ (bottom right).

6.3 Numerical simulations

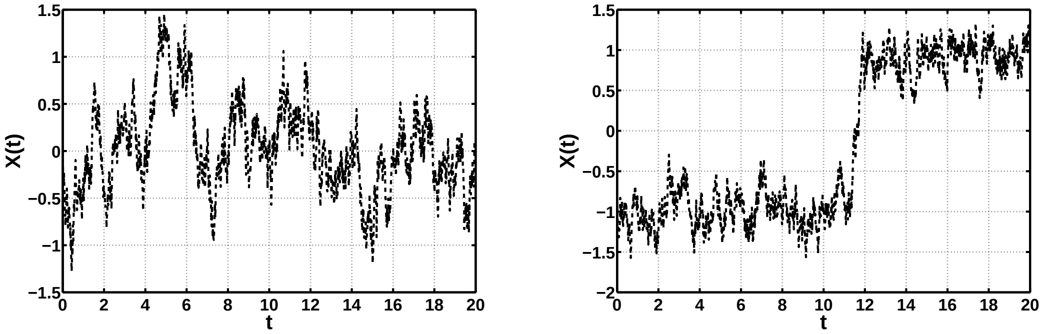
Before proceeding in exploring the convergence properties of the new LP algorithm, compared to the original VGPA framework, this section establishes the experimental setup that is used in the following sections. The systems considered here are the one-dimensional OU Fig. 6.4(a) and DW Fig. 6.4(b), and the three-dimensional L3D (as reviewed in Chapter 3).

6.3.1 Experimental setup

In the numerical experiments a fixed inference window of twenty time units (i.e. $T = [0,20]$) was considered for all systems and the time discretisation was set to $\delta t = 0.01$ to ensure numerical stability. Table 6.2 summarizes the *true* parameter values, that generated the sample paths for the following simulations.

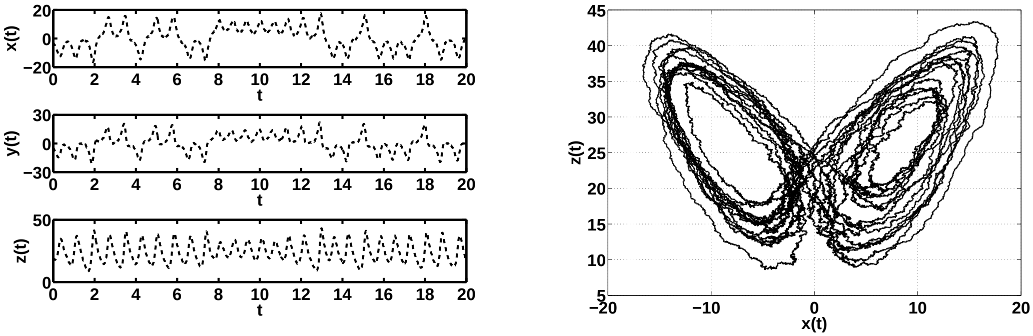
In a similar strategy to Apte et al. (2007), the discretisation is applied only in the posterior approximation; the inference problem is derived in an infinite dimensional framework (continuous time sample paths), as shown in Chapter 4. The Euler-Maruyama representation of the prior process (Eq. 4.1), leads to the following discrete time analogue:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{f}(t, \mathbf{x}_k; \boldsymbol{\theta})\ \delta t + \sqrt{\Sigma \delta t}\ \xi_k\ , \tag{6.6}$$



(a) OU sample path (b) DW sample path

Figure 6.4: Typical examples of OU (a) and DW (b) sample paths. These sample paths will be used as the histories in the experimental simulations, that produced the observations.



(a) Lorenz 3D simulation (b) Lorenz 3D in $x_t - z_t$ plane

Figure 6.5: (a) A typical realisation of the stochastic Lorenz ’63 system as time series in each dimension. (b) The same data but in $x_t - z_t$ plane where the effect of the random fluctuations is more clear.

where $\boldsymbol{\xi}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and the positive *infinitesimal dt* in Eq. (4.1), has now been replaced by a positive *finite* number δt. Moreover, this expression can be used to provide approximate sample paths (in terms of discretising a stochastic differential equation) from the prior process (Higham, 2001; Kloeden and Platen, 1999). This first order approximation imposes a suitably small discretisation step δt, if good accuracy is to be achieved.

6.3.2 Results of state estimation

The presentation of the experimental simulations begins with results for the OU process. Fig. 6.6, shows the results from the LP approximation of the VGPA algorithm, of polynomial order $Mo = 5$. For this example the observation density of 2 observations per time unit (hence 40 in the whole time domain $T = [0, 20]$), with $Mo = 5$ and $J = 41$, produces a set of $L_{poly} = 492$ coefficients to be inferred, compared to $N_{total} = 4000$ in the original VGPA framework. This is roughly 12.3% of the initial optimisation problem. For this system since the initial state $x_0 = 0$, is fixed in this simulations, as mentioned earlier, one can use the induced non-stationary covariance kernel function Eq. (3.7) and compute the exact posterior process. Comparing the results obtained from the

System	t_0	t_f	δt	$\boldsymbol{\theta}$	$\boldsymbol{\Sigma}$	N_{obs}	$\mathbf{R}$
OU	0	20	0.01	2	1	2	0.04
DW	0	20	0.01	1	0.8	2	0.04
L3D	0	20	0.01	$[10, 28, 2.6667]$	6	10	2

Table 6.2: Experimental setup that generated the data (trajectories and observations). Initial times (t_0) and final times (t_f) define a fixed time window of inference, whilst δt is the time discretisation step. $\boldsymbol{\theta}$ are the parameters related to the drift function, while $\boldsymbol{\Sigma}$ and $\mathbf{R}$ represent the noise (co)-variances of the stochastic process and the discrete observations respectively. In the multivariate system these covariance matrices are diagonal. N_{obs} represents the number of i.i.d. observations *per time unit* (i.e. observation density), which are taken at equidistant time instants.

LP approximation with the results from a GP regression smoother with the OU kernel the match is excellent, as expected for a linear system, where the approximation is theoretically optimal (in the limiting case as $\delta t \rightarrow 0$).

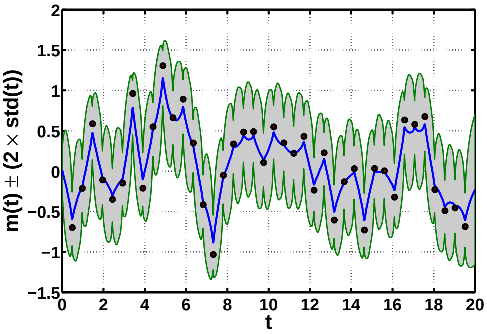
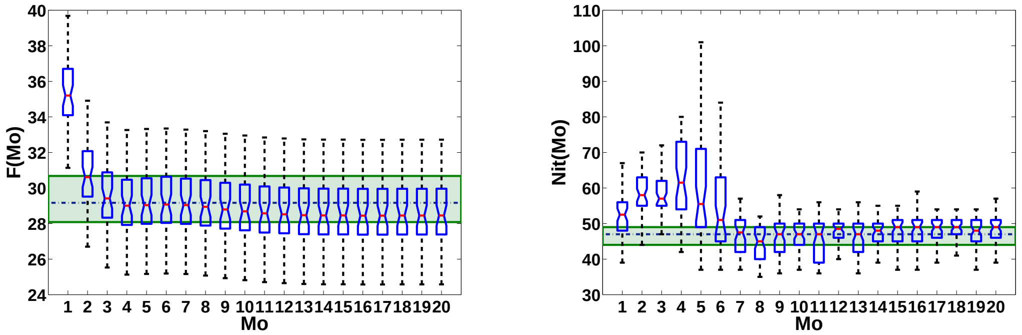


Figure 6.6: Marginal values of the means (solid line) and variances (shaded area) obtained by the LP approximation of 5'th order on a single realisation of the OU system. The results from the GP regression, on the same observation set, are visually indistinguishable and are omitted. The circles indicate noisy observations.

To provide a robust demonstration of the consistency of the results of the LP approximation, with respect to the original discretized VGPA, fifty different realisations of the observation noise, from a single trajectory, were used. The order of the polynomials was increased to explore convergence of the LP to the original VGPA. Summary statistics from these experiments, on the OU system, concerning the convergence of the free energy obtained from the LP approximation algorithm compared with the one from the original VGPA is shown in Figure 6.7(a). The median, the 25'th and 75'th percentiles are plotted in boxplots, while the extended vertical dashed lines indicate the 5'th and 95'th percentiles, from these 50 realisations, when the system has converged to its free energy minimum. For this example, with only second order polynomials (i.e. $Mo = 2$), the LP algorithm reaches the same free energy values as the original VGPA.

Figure 6.8(a) compares the results obtained from the LP approximation with 5'th order polyno-



(a) Variational free energy (b) Number of SCG iterations

Figure 6.7: (a) The median and the 25'th to 75'th percentiles in boxplots of the variational free energy, from fifty realisations of the observation noise, as a function of the increasing order of polynomials *Mo*, keeping the drift and diffusion parameters fixed to their true values. Extended vertical dashed lines indicate the 5'th and 95'th percentiles. The horizontal dashed (blue) line represents the 50'th percentile of the free energy obtained from the original VGPA on the same 50 realisations and the shaded area encloses the 25'th to 75'th percentiles. (b) Summaries from the same experiment concerning the number of iterations both algorithms needed to converge to optimality. Again, the horizontal lines (and shaded area) represent results obtained for the original VGPA, while boxplot results from the LP approximation, as in (a).

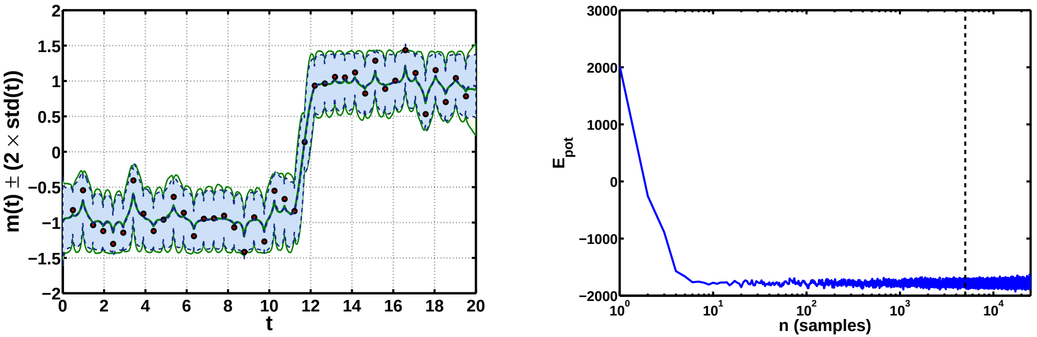
mials, on a single realisation of the DW system, to the outcomes of a Hybrid Monte Carlo (HMC)[1] sample from the posterior process, using the true values for the drift and diffusion parameters. The HMC solution is assumed here to provide a *reference solution* to the smoothing problem. The setting, for the DW example, is $25,000$ iterations of which the first $5,000$ are considered as *burn-in* and discarded. Each iteration generates 80 posterior sample paths of the system with artificial time $\delta\tau = 0.01$, and the last one is considered as the candidate sample path. In total $2,000,000$ sample paths are generated which are sampled uniformly to produce only $20,000$ samples from which to compute the marginal mean and variance as shown in Figure 6.8(a). The convergence results of this simulation are shown in Figure 6.8(b). Even though there exist recently proposed MC sampling algorithms, such as the *generalised HMC* as suggest by Alexander et al. (2005) which speed up the convergence of the Markov chain, here a rather classical hybrid Monte Carlo is implemented, as was first introduced by Duane et al. (1987).

Although the variance of the LP approximation is slightly underestimated, the mean path matches the HMC results and the time of the transition between the two wells is tracked accurately. The variational approximation as shown in Chapter 4 is likely to underestimate the variance of the approximating process (Minka, 2005) as is often the case when the expectation in the KL divergence is taken with respect to the approximating distribution[2] in Eq. (4.4).

Figures 6.9(a) and 6.9(b), present results comparable to Figures 6.7(a) and 6.7(b), but for the DW system. Again 50 different realisations of the observation noise from a single trajectory

[1] The HMC algorithm is reviewed briefly in Section 7.2.3.

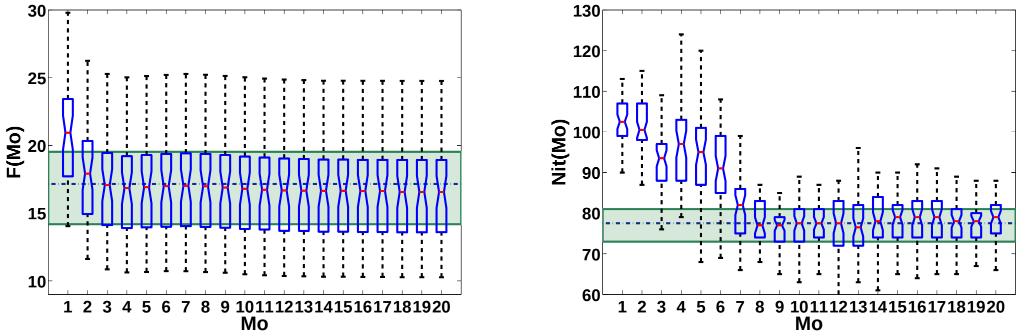
[2] That is $\mathrm{KL}[q_t \| p_t]$ instead of computing $\mathrm{KL}[p_t \| q_t]$, where p_t is the true posterior while q_t is the approximate one.



(a) HMC vs LP (b) Potential energy trace

Figure 6.8: (a) Comparison of the approximate marginal mean and variance (of a single DW realisation), between the "correct" HMC posterior estimates (solid green lines and light shaded area) and the LP approximation, of 5'th order, (dashed blue lines and dark shaded area). The circles indicate noisy observations. (b) Trace of the potential energy (horizontal axis is in log-space), of the Hamiltonian,in the HMC posterior sampling. The vertical dashed line, indicates the end of the burn in period and the beginning of the posterior sampling.

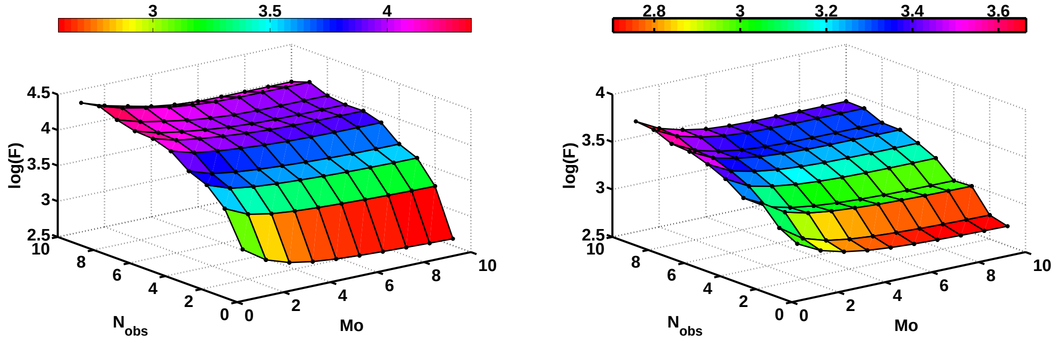
were generated and both LP approximation and VGPA algorithms were applied, given the true parameter values for the drift and diffusion coefficients. The summaries from these runs show the consistency of the LP approximation, when applied to non-linear systems. The algorithm exhibits stability and slightly outperforms the original VGPA framework, in terms of minimizing the free energy, although this has a very minor impact in terms of solving the ODEs (Eq. 4.12, 4.13) to produce the marginal means and variances as shown in Figure 6.8(a).



(a) Variational free energy (b) Number of SCG iterations

Figure 6.9: (a) Similar to Fig. 6.7(a), but from fifty different realizations of the observation noise of the DW system. (b) Again, similar to Fig. 6.7(b), but for the DW system.

However, when the LP approximation is applied one must be aware that the complexity of the algorithm (i.e. numbers of degrees of freedom), scales not only with the order of the imposed polynomial, but also with the frequency of the measured values (i.e. observation density) as shown in Eq. (6.5). Thus, to address the sensitivity of the LP approximation, as both these quantities vary, the algorithm was tested for $1 \leq N_{obs} \leq 10$ and $1 \leq Mo \leq 10$, on both OU and DW systems as shown in Figures 6.10(a) and 6.10(b), respectively. At each point on the grid, the result from thirty different realisations of the observation noise were averaged and presented. The behaviour of the LP approximation is similar in both systems tested, and confirms the initial belief that when a system is very frequently observed, one can apply even a linear polynomial approximation ($Mo = 1$), between observation times to approximate the variational parameters.



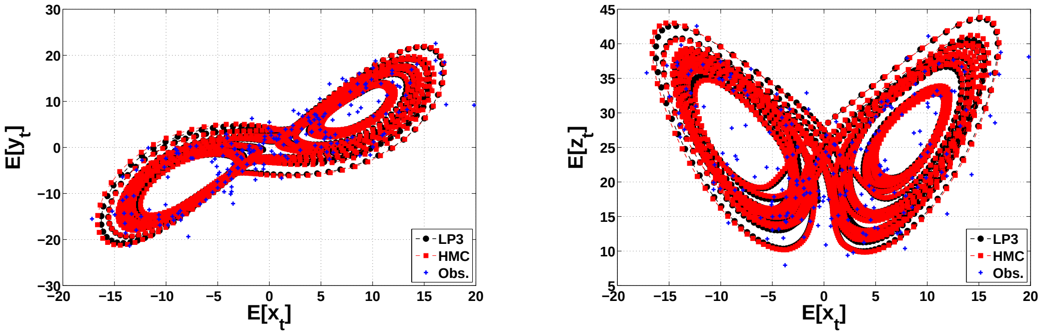
(a) *OU system* (b) *DW system*

Figure 6.10: (a) The log average free energy, at convergence, from thirty different realisations of the observation noise, of a single OU trajectory, as a function of both observation density and order of polynomials (i.e. $log(F(N_{obs}, Mo))$). Figure (b), repeats the same experiment but for the DW system.

To provide a more complete assessment of how this new LP approximation approach to the VGPA algorithm scales with higher dimensions the same experiments were repeated on a multivariate system, namely the Lorenz '63 (L3D). Figures 6.11(a) and 6.11(b), show the approximated mean paths obtained with a 3'rd order LP algorithm, against the posterior mean paths computed using HMC, in *xy* and *xz* planes respectively, from a single realisation of the stochastic L3D shown in Figure 6.5(a). The observation density for this example was relatively high ($N_{obs} = 10$, per time unit), hence it was possible to relax the order of the polynomials to $Mo = 3$.

In this example, unlike the previous case of the DW, the LP approximation overestimates the marginal variance (Figure 6.12(b)) compared with the estimates obtained by using HMC. However, the same effect is also observed when applying the original VGPA framework, hence this is not an artefact of the polynomial approximation but rather of the variational framework.

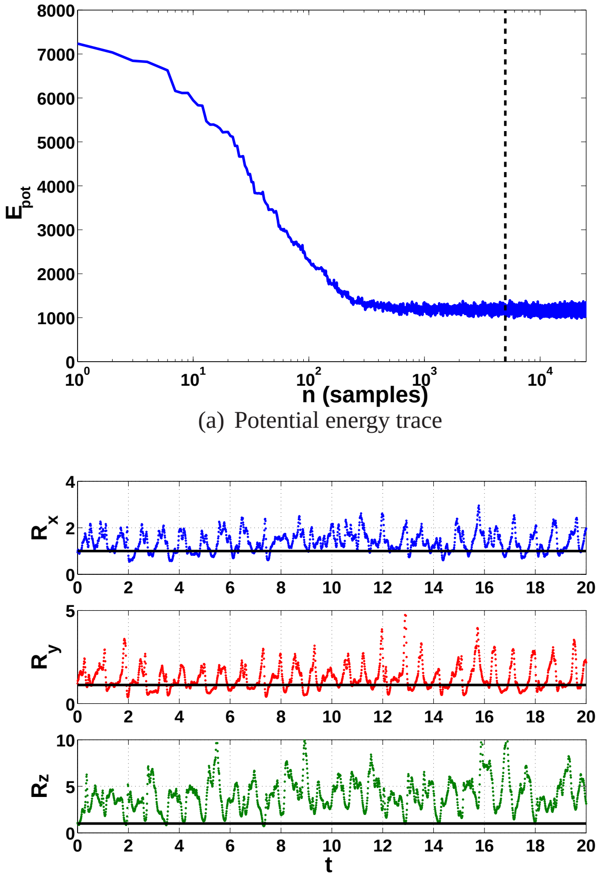
The tuning of the HMC sampling scheme was similar to the one used to obtain the posterior estimates for the DW system, only in this case a smaller artificial time step was necessary to correctly sample the posterior process. In total $25,000$ iterations of the HMC algorithm were used, with the first $5,000$ considered *burn-in*. Each iteration integrated the artificial Hamiltonian dynamics (Eq. 7.14) for 50 iterations, where only the last one was the candidate sample path. The artificial time step was $\delta\tau = 0.004$. Sampling from high dimensional distributions with the HMC is not a trivial task. Continuous time *sample paths*, which when discretised result in a large number of random variables that need to be jointly sampled at each iteration is challenging. For the L3D system considered here, the dimensionality of the discretised sample path is $N_{rv} = 6003$ (i.e. one needs to sample jointly N_{rv} random variables at each iteration). The trace of the potential energy



(a) *xy* plane (b) *xz* plane

Figure 6.11: The marginal means, obtained from the LP approximation and the HMC sampling in *xy* (a) and *xz* (b) planes respectively, on a single realisation of the L3D (see Fig. 6.5(b)). In both plots, the dots (black) are the results from the LP approximation (of 3'rd order), while the squares (red) are results from HMC. Crosses (blue) indicate the noisy observations. The $E[\cdot]$ notation that appears in the figures axis represents *expected* value.

of the Hamiltonian (for the L3D example), is presented in Fig. 6.12(a).

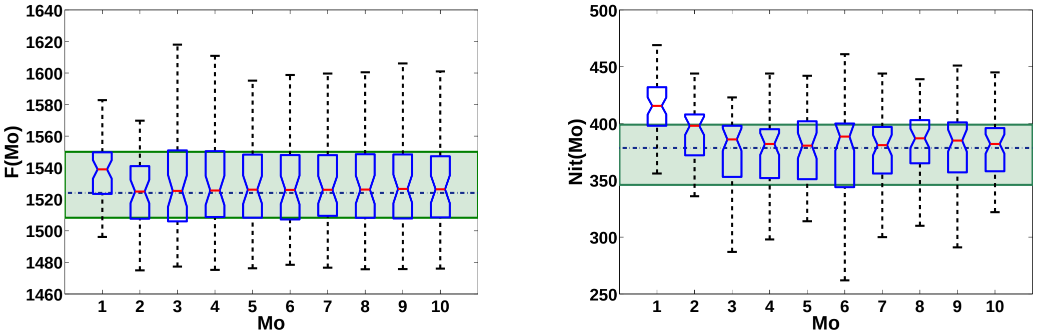


(a) Potential energy trace

(b) Ratios in marginal variance vs time

Figure 6.12: (a) Trace of the potential energy of the Hamiltonian in the HMC posterior sampling of the L3D example. The vertical dashed line, indicates the end of the burn in period and the beginning of the posterior sampling. Notice the logarithmic scale on the horizontal axis. (b) The ratios, in each dimension of the L3D, between the LP approximate variance to the variance obtained by the HMC sampling (i.e. $\frac{var[LP]}{var[HMC]}$), as functions of time. The overestimation from the LP approximation is apparent in all three dimensions.

The performance of the new polynomial framework scales well for this multivariate system. As shown in Figures 6.13(a) and 6.13(b), when comparing the minimisation of the free energy and the number of iterations to reach convergence, the LP approximation is very stable and fully converges to the original VGPA with only $Mo = 2$ order of polynomial. The experiments were extended up to $Mo = 20$, and showed similar outcomes although with higher computational cost and are omitted from the plots. The observation density considered (i.e. $N_{obs} = 10$) implies that $Mo = 9$ is the limit where both algorithms LP and VGPA optimise the same number of parameters. For values of $Mo > 9$, the LP becomes more demanding in computational resources. However, when tested with $Mo = 3$, $L_{poly} = 9,648$ whilst $N_{total} = 24,000$ hence achieving 59.8% reduction in optimised variables.



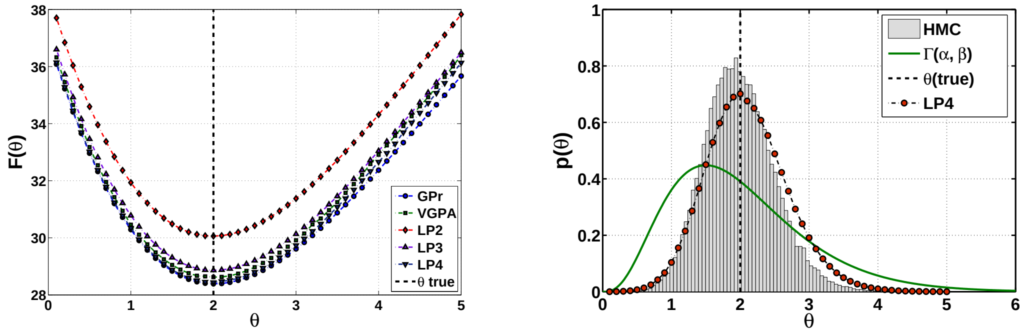
(a) Variational free energy (b) Number of SCG iterations

Figure 6.13: (a) Boxplots of the free energy attained from 50 realisations of the observation noise (on a single L3D sample path) as a function of the order of polynomials Mo. The horizontal dashed line (and the solid ones above and below) represent the 25, 50 and 75 percentiles from the VGPA free energy on the same data sets. (b) Presents a similar plot but for the number of iterations, in the SCG optimisation routine, that convergence was achieved. In both plots the extreme values (outliers) have been removed for clearer presentation.

The reduction in the memory requirements of the algorithm does not produce a similar reduction in computational time. Figures 6.7(b), 6.9(b) and 6.13(b) compare the number of iterations of the LP algorithm to reach convergence with the number of iterations from the VGPA. These results are summaries from 50 different realizations (of the observation noise on a single trajectory) of the OU, DW and L3D systems respectively, and show that the VGPA algorithm, while optimising a larger number of parameters, still converges in slightly fewer iterations.

6.3.3 Results of parameter estimation

The new LP algorithm is able to estimate the (hyper-) parameters of the aforementioned dynamical systems, in the same way as in the original VGPA algorithm. Chapter 4, described two ways of performing this task. First by constructing discrete approximations to the posterior distribution of the parameters and second by providing Maximum Likelihood type-II point estimates. Both approaches are based on the upper bound that the *variational free energy* provides to the true marginal likelihood (Eq. 4.6). In this section the focus is on estimating the drift parameters $\boldsymbol{\theta}$ and diffusion coefficient $\boldsymbol{\Sigma}$, although estimation of the prior distribution, over the initial state (i.e. $\mathcal{N}(\mu_0, \tau_0)$) and the noise related to the observations $\mathbf{R}$ are straightforward extensions.

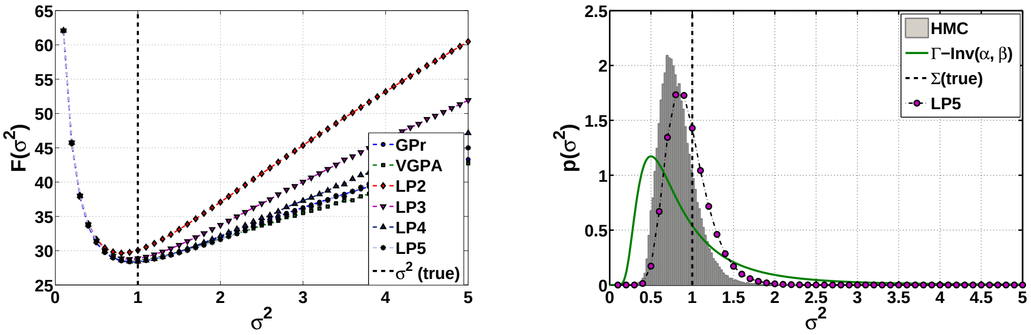


(a) θ marginal profile (b) Posterior distribution p(θ)

Figure 6.14: *OU system:* (a) The profile marginal likelihood of the drift parameter θ, keeping the system noise Σ fixed to its true value, obtained by the GP regression (blue circles) with the OU kernel, which gives the exact likelihood, against the original VGPA algorithm (green squares) and the new LP extension with different order of polynomials. (b) The histogram of the posterior samples obtained with the HMC. The continuous green line shows the $\mathcal{G}(4.0, 0.5)$ prior of the (hyper)-parameter θ, while the red circles connected with the dot-dashed line represent the discrete approximation to the posterior distribution obtained by the point estimates of the LP algorithm with 4'th order polynomials. Both the HMC posterior sample histogram and the LP approximation have been normalized, such that the area they define sums to unity. In both figures the vertical dashed line represents the true parameter value that generated the data.

Figure 6.14(a), compares the profile of the approximate marginal likelihood, of the OU drift parameter, obtained with the original variational framework and the local polynomial approximation, on a typical realisation. For this system the "true" marginal likelihood can be obtained using a Gaussian process regression smoother (with OU kernel function). Also the LP framework converges to the original VGPA when 4'th order polynomials are employed, which is consistent with the state estimation results in Fig 6.7(a). The minimum of the profile can be well identified with only 2'nd order polynomials, which suggests that for the drift parameter, in this example, the bound on the true likelihood does not need to be very precise, if a point estimator is sought.

Figure 6.14(b), shows the results from the LP (of 4'th order) discrete approximation to the posterior distribution of the drift parameter θ using a $\mathcal{G}(4.0, 0.5)$ prior. Here the results are compared with $80,000$ posterior samples (presented as a histogram), obtained from four independent Markov chains ($20,000$ samples per chain), using HMC sampling. The same prior distribution (continuous green line) is used in both cases and in addition the results are presented such that the areas defined by the histogram and the approximate discrete estimates (red circles), sum to one. Although the results, for both algorithms, are slightly biased the LP algorithm provides a better approximation because for a linear system, such as the OU, the variational Gaussian process yields an optimal approximation while the HMC approximation remains subject to finite sample effects.



(a) σ^2 marginal profile

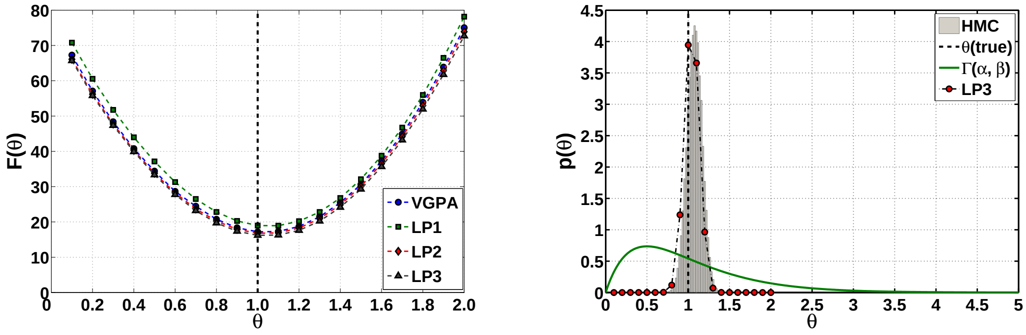
(b) Posterior distribution $p(\sigma^2)$

Figure 6.15: *OU system:* (a) Plot similar to Fig. 6.14(a) only for the system noise variance σ^2 and keeping the drift θ fixed to its true value. Again, the results of the GP regression represent the exact marginal likelihood. (b) As Fig. 6.14(b), only the continuous line now is the $\mathcal{G}^{-1}(3.0, 2.0)$ prior of the (hyper-) parameter σ^2.

Figures 6.15(a) and 6.15(b), show similar profile and posterior results, but for the OU system noise coefficient σ^2. It is apparent that for this parameter the LP method needs higher order polynomials to match the results from the original VGPA. All methods locate the minimum of the profile at a smaller value than the true one. Furthermore, both methods seem to deviate from the true likelihood (blue circles), as the value of this parameter becomes more distant from the true value that generated the data. The same bias effect can also be seen in Figure 6.15(b), where the LP method (5'th order) is compared with the HMC posterior sampling. However, MCMC methods for sampling this parameter can be problematic due to the high dependencies between the system noise σ^2 and the states of the system x_t, which results in slow rates of convergence (Roberts and Stramer, 2001; Golightly and Wilkinson, 2006). Again the same $\mathcal{G}^{-1}(3.0, 2.0)$ prior (continuous green line), was used for both algorithms.

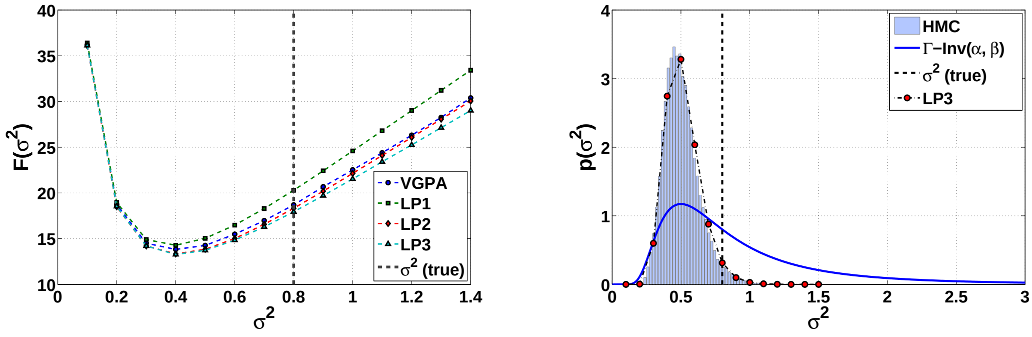
Similarly, the approximate posterior distributions and profile likelihoods, for a single realisation of the DW system are presented for the drift θ in Figures 6.16(a) and 6.16(b) and for the diffusion coefficient σ^2 in Figures 6.17(a) and 6.17(b). Here there is no method to compute the exact likelihood, hence the only comparison is between the profiles obtained from the VGPA algorithm against those obtained with the LP. For both parameters θ and σ^2, the results are almost identical with 3'rd order polynomials. Both estimates are biased, the drift towards a higher value, while the noise towards a smaller value, but these biases are consistent with those seen in the HMC posterior samples.

The profiles of the drift parameter vector $\boldsymbol{\theta} = [\sigma\ \rho\ \beta]^\top$ for the *L3D* system are shown in Fig. 6.18(a) where the original VGPA algorithm (red circles) is plotted against the LP approximation, with 2'nd order polynomials (green squares). The results are almost indistinguishable and



(a) θ marginal profile (b) Posterior distribution p(θ)

Figure 6.16: *DW system:* (a) The profile approximate marginal likelihood of the drift parameter θ, keeping the system noise σ^2 fixed to its true value, obtained by original VGPA algorithm (blue circles) and the new LP extension with different order of polynomials. (b) The histogram of the posterior samples obtained using the HMC. The continuous green line shows the $\mathcal{G}(2.0, 0.5)$ prior of the (hyper-) parameter θ, whilst the red circles connected with the dot-dashed line represent the approximate posterior distribution obtained by the discrete estimates of the LP algorithm with 3'rd order polynomials. Both the HMC posterior sample histogram and the LP point estimates have been normalized, such that the area they define sums to one.

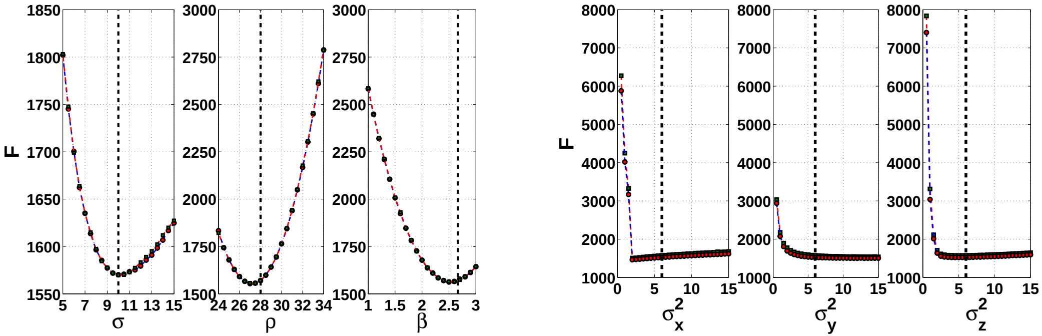


(a) σ^2 marginal profile (b) Posterior distribution $p(\sigma^2)$

Figure 6.17: *DW system:* (a) Plot similar to Fig. 6.16(a) only for the system noise variance σ^2 and keeping the drift θ fixed to its true value. (b) As in Fig. 6.16(b), only the continuous line now is the $\mathcal{G}^{-1}(3.0, 2.0)$ prior of the (hyper-) parameter σ^2. Again the areas that both algorithms define (HMC and LP) have been normalized. In both figures the vertical dashed line represent the true parameter value that generated the data.

the minimum values are well estimated for all parameters. Fig. 6.18(b), presents similar profiles but for the diagonal elements of the Σ matrix (i.e. σ_x^2, σ_y^2 and σ_z^2). Although both the VGPA and the LP (3'rd order) exhibit identical behaviour unlike the drift parameters the system noise profiles are not as informative. Only the first dimension 'x', shows a clear minimum, although strongly biased towards a smaller value (the true values are indicated with vertical dashed lines). The third dimension 'z', shows a weak minimum, i.e. there is quite flat region around the minimum value and the second dimension 'y', does not possess a minimum within the range of values explored.

Figure 6.19 (upper three panels), presents the posterior estimates of the *L3D* drift vector $\boldsymbol{\theta}$, obtained from the HMC algorithm. The lower three panels present the approximate posterior



(a) $\boldsymbol{\theta}$ marginal profiles (b) $\boldsymbol{\Sigma}$ marginal profiles

Figure 6.18: *L3D system:* (a) The profile approximate marginal likelihood for all three parameters of the *L3D* drift vector. From left to right the profiles for σ, ρ and β obtained from the original VGPA algorithm (red circles) are compared against those obtained with the LP with 2'nd order polynomials (green squares). (b) As before but for the system noise, on each dimension (σ_x^2, σ_y^2 and σ_z^2). Here the LP approximation uses 3'rd order polynomials. The vertical dashed lines indicate the true values of the parameters that generated the datasets.

distributions (discrete estimates) from the LP algorithm. Both algorithms used the same prior distributions $p_0(\sigma) = \mathcal{G}(20, 0.5)$, $p_0(\rho) = \mathcal{G}(56, 0.5)$ and $p_0(\beta) = \mathcal{G}(6, 0.5)$. Nonetheless, the comparison between the upper and lower panels is not straightforward because the approximate posterior distributions obtained with the LP algorithm are conditional, in the sense that the two other drift parameters are kept fixed to their true values, whereas the posterior distributions from the HMC are obtained jointly (i.e. all the drift parameters are sampled simultaneously). The results from the LP method show weak biases towards smaller values in all parameters, which is consistent with the HMC results, except the σ (drift) parameter (first column) which the LP approximation estimates more accurately.

6.3.4 Stochastic Lorenz '96 (40D)

In this section the application of the new LP variational approximation framework is illustrated in a forty dimensional system, namely the Lorenz '96 (L40D). An example of this system is given in Figure 6.20(a), where all forty dimensions are shown for a time period of ten units $T = [0, 10]$.

Figure 6.20(b), shows the approximate marginal mean $\mathbf{m}_t$ and variance $\mathbf{S}_t$, of three selected dimensions from the $L40D$ system. The mean paths are reasonably smooth and the variances are broad enough to enclose the observations. Similar results were also obtained for the other dimensions of the system.

Finally the new approach was compared against the original VGPA algorithm, in producing conditional profiles for the forcing (drift) parameter $\boldsymbol{\theta}$ (see Figure 6.21(a)) and system noise coefficients $\boldsymbol{\Sigma}$ (see Figure 6.21(b), for the system noise in the the 20'th dimension). Both algorithms produce smooth profiles, with the new approach identifying the minimum slightly better.

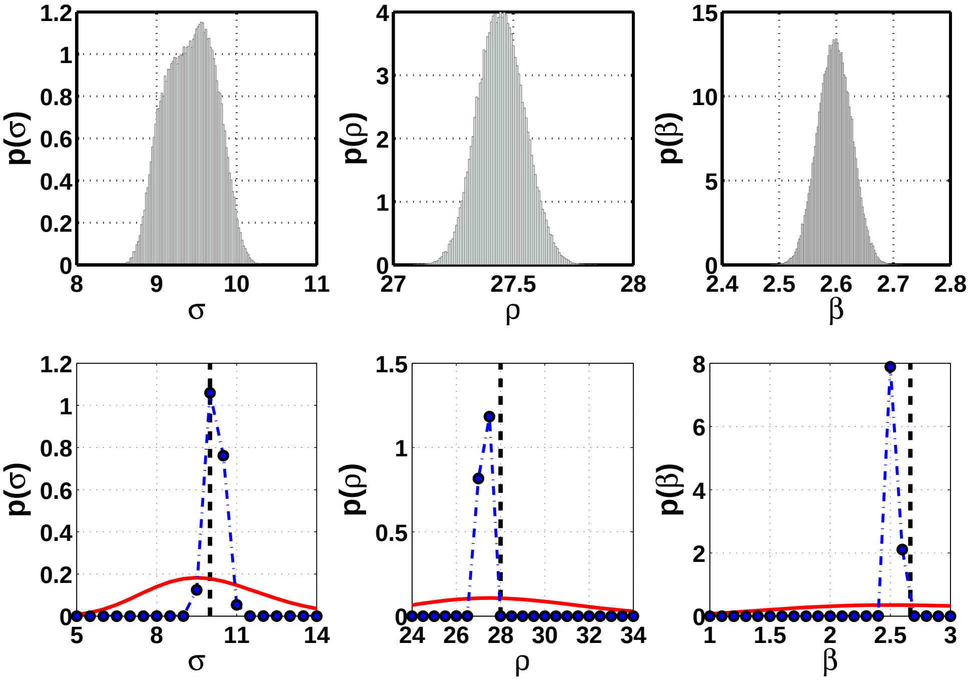
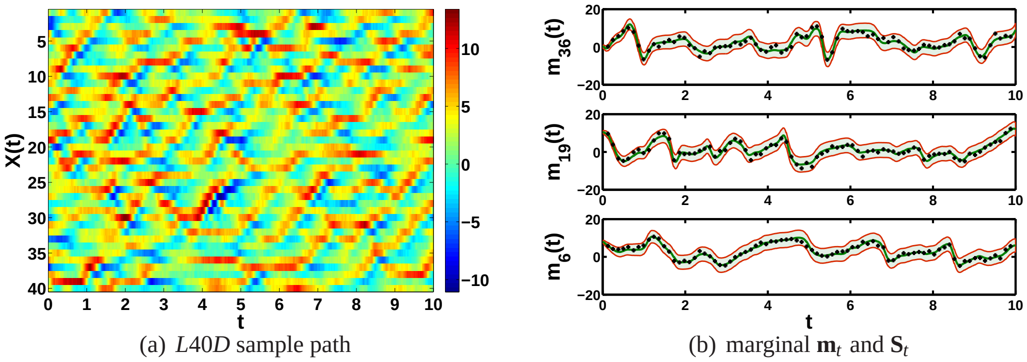


Figure 6.19: *L3D system:* The upper three panels, starting from left to right, present the joint posterior HMC samples for the drift parameters σ, ρ and β. The lower three panels, following the same order, show the approximate posterior distributions (blue dots connected with the dot-dashed line) obtained from the LP algorithm with 2'nd order polynomials. The continuous lines represent the *Gamma* prior distributions that were used. Notice that the priors are very broad. In all the above results the system noise is assumed to be known and fixed to its true value.

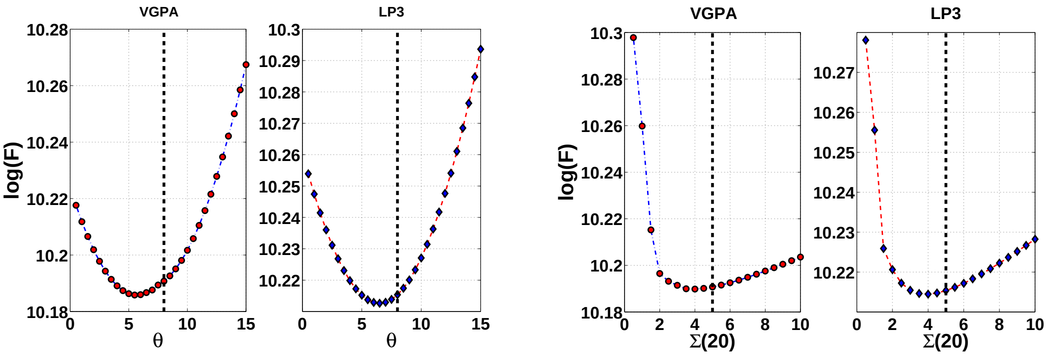


(a) *L40D* sample path

(b) marginal $\mathbf{m}_t$ and $\mathbf{S}_t$

Figure 6.20: *Lorenz 40D:* In (a) all forty dimensions (top to bottom) of a ten units time-window ($T = [0, 10]$), of the stochastic Lorenz 40*D* system, used for the experiments. (b) presents three examples (3'rd, 19'th and 36'th dimension) of the marginal means (solid green line) and variances (shaded light green area) obtained with the *LP* algorithm (3'rd order), at convergence. The crosses indicate the noisy observations. Similar result were also acquired for the remaining dimensions.

However, more important is that these results were obtained by achieving a significant reduction of 67.6% in optimisation space. For this example, with eight observations per time unit (hence $J = (8 \times 10) + 1 = 81$) and third order polynomials (hence $Mo = 3$), one needs to infer $L_{poly} = 531,360$ variables, comparing to $N_{total} = 1,640,000$, of the original VGPA.



(a) $\boldsymbol{\theta}$ profile (b) $\boldsymbol{\Sigma}$ profile

Figure 6.21: *Lorenz 40D:* In (a) the approximate marginal profile log likelihood of the drift parameter θ, obtained with the original *VGPA* algorithm (left panel, red circles) is compared against the one obtained with the *LP* algorithm with 3'rd order polynomials (right panel, blue diamonds). In this example the system noise covariance matrix $\boldsymbol{\Sigma}$ is fixed to its true value. (b) presents similar results but for the conditional estimation of the system noise on the 20'th dimension, assuming the drift is known. Similar profiles were also generated for other dimensions. In all sub-plots the vertical dashed lines represent the true values of the parameters that generated the data.

6.4 Discussion

This chapter has introduced an alternative parametrisation of the VGPA algorithm. This new approach uses local polynomials to approximate the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$ of the linear drift approximation Eq. (4.7) to control the complexity of the algorithm and reduce the number of variables need to be optimized. The LP algorithm is validated on a range of different systems to test its convergence behaviour w.r.t. the original VGPA and shows excellent stability. In most of the examples 3'rd order polynomials are required to match the original algorithm, although the order is likely to increase as the observations become more sparse (i.e. the time between observations increases).

Despite the notable reduction in optimized variables the LP approach does not produce similar results in computational time. This is mostly because the new gradients of the cost function, Equation (6.3), w.r.t. the coefficients of the polynomial approximations, have to be computed separately in each sub-interval where each polynomial is defined (see Appendix C). In the current implementation priority was not given to the computational cost, hence a simple serial approach was chosen. However, a parallel implementation in which the necessary gradients are computed simultaneously is straightforward and could dramatically reduce the execution time, especially when treating long time windows.

The new LP algorithm can be used to construct, computationally cheap, discrete approximations to the posterior distribution of the (hyper-) parameters $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ (Section 4.5) and it shows that it can match the results of the HMC sampling rather well, in the examples tested.

Another advantage with the LP framework is that different classes of polynomials can be used. This approach was explored here using mostly orthogonal classes of polynomials, such as Chebyshev and Legendre. However the results were not significantly different in the systems explored here and hence were omitted.

Although the application of this variational approach to the forty dimensional Lorenz ’96 system (L40D) is very encouraging, there is still an open question on how these methods can be applied to very high dimensional models (such as those used for numerical weather prediction). The LP approximation is a step towards that direction. In most of the examples presented here the computational resources were reduced by more than 60% (in terms of optimizing variables) compared to the original VGPA. By imposing further assumptions on the Gaussian process approximation (e.g. by defining a special class of linear drift functions) it is possible to control the complexity of the posterior variational approximation and reduce the number of variables even further.

7 Comparison with other methods

CONTENTS

7.1 Foreword

Chapter 6 introduced a new extension of the VGPA algorithm, by approximating the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$ of the linear drift $\mathbf{g}_L(\mathbf{x}_t)$, with polynomials that were defined between each pair of observations. The convergence properties of this approach, with respect to the original VGPA, were tested and it was proven (experimentally), that this new approach can produce similar results to the original framework with a significant reduction in the number of optimised variables. In addition the new approach showed beneficial characteristics when estimated the model parameters (discrete approximations to the posterior distributions).

Moreover, the original VGPA algorithm can also be used to provide point estimates of the (hyper-) parameters, as shown in Chapter 4, within a gradient based estimation technique (pseudocode in Table 4.2). The same dual optimisation approach can also be used with the LP approximation framework, without any change in the implementation of the code, since the re-parametrisation of the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$, affects only the smoothing algorithm (see inner loop, Table 4.1), while leaving the outer loop unaffected. In fact, the new approach is more flexible, because it is possible to adjust the bound of the variational algorithm to the marginal likelihood, by tuning the order of the polynomial approximation.

The aim of this chapter is two fold: **(a)** to describe briefly, a range of different methodologies that were implemented which solve the state and parameter estimation problem, in dynamical systems, from a Bayesian point of view, as reviewed in Chapter 2 and **(b)** to present a comprehensive study in comparing, empirically, the aforementioned estimation methods with the new LP approximation framework in terms of estimating the (hyper-) parameters of three dynamical systems.

7.1.1 Chapter outline

The chapter begins with a brief description of different estimation methods that were implemented to compare with the original VGPA and the LP extension on state and parameter estimation problems. Next, the methods are applied on a single example of the DW system highlighting the different performances. However, the main contribution of this chapter is presented in Section 7.4, where an extensive empirical study compares the LP method with a dual-UnKF and a weak constraint 4D-Var method in estimating the parameters of three dynamical systems, namely the OU, DW and L3D. The results are summarised and discussed in the final section.

7.2 Methods implemented

This section briefly describes a range of estimation techniques that were implemented to assess the original VGPA and the LP extension when estimating the systems states and (hyper-) parameters on a range of dynamical systems. The methods have already been introduced in Chapter 2, therefore here a small description will be given along with some implementation details.

7.2.1 Ensemble Kalman filter / smoother

In general, ensemble filters are based on the intuition that it is easier to approximate a probability distribution than it is to approximate a non-linear function. Therefore, ensemble methods use a Monte Carlo approach and propagate forward in time a number of states (the ensemble) through the exact model. This ensemble (typically of size $O(10^2)$), represents the state's distribution and its first two moments, the mean and the covariance, are typically used as the summary statistics. The (predictive) mean and covariance are given by:

$$\mathbf{m}_t^F = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_t^n \,, \tag{7.1}$$

$$\mathbf{S}_t^F = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}_t^n - \mathbf{m}_t^F)(\mathbf{x}_t^n - \mathbf{m}_t^F)^\top \,, \tag{7.2}$$

where N is the size of the ensemble, $\mathbf{x}_t^n \in \Re^D$ represents the n'th ensemble member (at discrete time 't') and $\mathbf{m}_t^F \in \Re^D$ and $\mathbf{S}_t^F \in \Re^{D\times D}$ are the ensemble mean and covariance respectively. The superscript F indicates that these are the *Filtered* mean and covariance, in contrast with the *Smoothed* versions that will be introduced shortly.

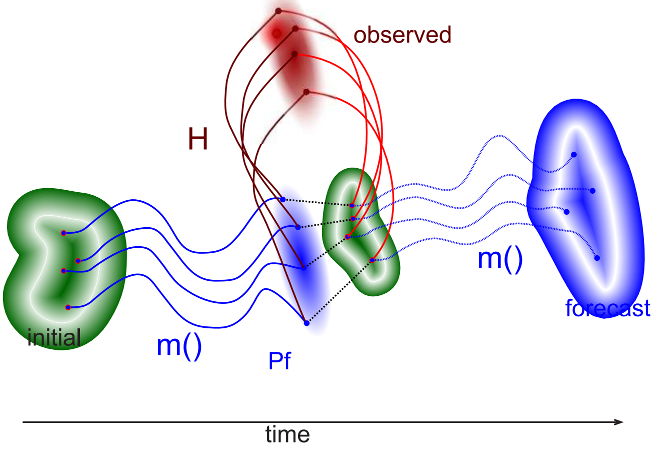


Figure 7.1: The initial ensemble of particles (or system states), is propagated forward in time using the exact model $m()$. Then in the light of observation each ensemble (forecast) member is updated and a new initial ensemble is created for the next propagation.

Figure 7.1 shows an example of the ensemble Kalman filter in practice. The algorithm proceeds as follows: initially an ensemble of particles $\{\mathbf{x}_{t=0}^n\}_{n=1}^N$, is created by sampling from some prior distribution. Then each member of the ensemble is propagated forward in time through the full non-linear model $m(\cdot)$ to create the *forecast* ensemble (i.e. $\{\mathbf{x}_{t+1}^n\}_{n=1}^N = \{m(\mathbf{x}_t^n)\}_{n=1}^N$). This propagation is made until there is an observation. At observation times 't_k' each ensemble member is mapped through the general observation operator $h(\cdot)$ (or $\mathbf{H}$ if is assumed linear for simplicity) to the observation space and updated given the observation $\mathbf{y}_k$. Ideally, one could use an ensemble of observations, so that each particle is updated by a different observation. This would ensure the covariance structure of the ensemble is maintained in agreement with the observation's error covariance $\mathbf{R}$ (Burgers et al., 1998). However, generating an ensemble of observations would be too costly, therefore only a single measurement $\mathbf{y}_k$ is used instead.

The ensemble Kalman smoother (EnKS), is a natural extension of the EnKF only instead of assimilating the observations sequentially up to time 't' it uses all available observations within the predefined time window. Two predominant approaches for smoothing are (**a**) the *two-filter* smoother and (**b**) the *forward-backward* smoother. The first approach uses a linear combination of two independent filters which run in forward and backward directions. However a common mistake with this approach is the use of the inverse forward dynamics to obtain the backward dynamical model, which does not in general lead to the correct result (Klaas et al., 2006). On the contrary, the second smoothing approach requires a separate backward filter which recursively computes corrections to the forward pass.

Many implementations of an EnKS have been introduced in the literature such as Evensen and van Leeuwen (1999); van Leeuwen (2001). Here the smoothing approach is based on the Rauch-Tung-Striebel smoother (Rauch et al., 1965) (forward-backward type) and implementation details can be found in Sarkka (2008).

7.2.2 Unscented Kalman filter / smoother / dual estimation

The unscented Kalman filter (UnKF) is in the same spirit as the ensemble Kalman filter. The main difference is that instead of maintaining a (possibly large) randomly generated ensemble, it rather chooses deterministically a set of typically ($N = 2D + 1$) particles (or sigma points), where D is the dimensionality of the state vector $\mathbf{x}_t$. These sigma points capture essential information about the first two moments of the distribution that they approximate. The predictive mean and variance are given by weighted sums of the sigma points as follows:

$$\mathbf{m}_t^F = \sum_{n=0}^{N-1} w_{(mean)}^n \mathbf{x}_t^n \,, \tag{7.3}$$

$$\mathbf{S}_t^F = \sum_{n=0}^{N-1} w_{(cov)}^n (\mathbf{x}_t^n - \mathbf{m}_t^F)(\mathbf{x}_t^n - \mathbf{m}_t^F)^\top \,, \tag{7.4}$$

where $w^n_{(mean|cov)} \in \Re$ defines the n'th weight. Assuming that the state vector $\mathbf{x}_t$ (at time 't') has mean $\mathbf{m}^F_t$ and covariance $\mathbf{S}^F_t$, the selection of sigma points for the next time instant 't+1' is done according to the following rule:

$$\chi^0_t = \mathbf{m}^F_t \ , \tag{7.5}$$

$$\chi^n_t = \mathbf{m}^F_t + \left(\sqrt{(D+\lambda)\mathbf{S}^F_t}\right)^n ,n \ = 1,\ldots,D \, , \tag{7.6}$$

$$\chi^n_t = \mathbf{m}^F_t - \left(\sqrt{(D+\lambda)\mathbf{S}^F_t}\right)^{n-D} ,n \ = D+1,\ldots,2D \, , \tag{7.7}$$

$$\mathbf{x}^n_{t+1} = f(\chi^n_t) \ ,n \ = 0,\ldots,2D \, , \tag{7.8}$$

where $f(\cdot)$ is the non-linear transformation (system dynamics) and the exponent n (on the right hand side) indicates the n'th column of the square matrix. The weights for the means and the covariances need not be the same. These are selected (usually) only once at the beginning of the estimation procedure as follows:

$$w^0_{(mean)} = \lambda/(D+\lambda) \ , \tag{7.9}$$

$$w^0_{(cov)} = \lambda/(D+\lambda) + (1-\alpha^2+\beta) \ , \tag{7.10}$$

$$w^n_{(mean|cov)} = 1/(2(D+\lambda)) \ ,n \ = 1,\ldots,2D \ \text{,with} \tag{7.11}$$

$$\sum_n w^n_{(mean|cov)} = 1 \ , \tag{7.12}$$

where $\lambda = \alpha^2(D+\kappa)-D$, is a scaling parameter, $\alpha \in \Re$ determines the spread of the sigma points around the mean, $\kappa \in \Re$ is a secondary scaling parameter (usually set to zero) and $\beta \in \Re$ is used to incorporate prior knowledge of the distribution of $\mathbf{x}$. For further details concerning on the choice of the sigma points and the tuning of the weights we refer to van der Merwe (2004).

As discussed in Chapter 2, this method utilizes a technique known as the "*unscented transformation*", to estimate the states of the dynamical system considered and was primarily introduced, as an alternative to the extended Kalman filter (EKF), to address its linearisation limitations. The UnKF has been extended to model parameter estimation problems in Wan and van der Merwe (2000) and Wan et al. (2000). Two approaches were taken: **(a)** augmenting the state vector with the model parameters and then applying a single filter recursion to estimate both of them *jointly* and **(b)** using two separate filters, one to estimate the system states, given the current estimates for the parameters, and one to estimate the model parameters given the current state estimates. In the latter approach the two filters are run in parallel and are known as the *dual filter*.

In this work, for estimating the states of a system a version of an unscented Kalman filter and smoother were implemented as proposed in van der Merwe and Wan (2001) and Sarkka (2008), respectively. These papers not only describe the proposed algorithms but also provide detailed pseudocode which guided the implementation.

For the estimation of the model parameters a dual unscented Kalman filter (dual UnKF), similar to the one used by Gove and Hollinger (2006) to assimilate net CO_2 exchange between the surface and the atmosphere, is implemented. Again for more implementation details the reader is referred to van der Merwe and Wan (2001).

7.2.3 Hybrid Monte Carlo

The HMC algorithm Duane et al. (1987), is a Markov chain Monte Carlo (MCMC) technique that combines Hamiltonian molecular dynamics with the Metropolis-Hastings accept / reject criterion to sample from complex distributions. In this setting the HMC algorithm proposes a new configuration (or a new sample path) by sampling from the posterior distribution Eq. (4.3).

The algorithm begins with an initial (discrete time) sample path $\mathbf{x}^j = \{x_k^j\}_{k=0}^N$, where $j > 0$ is the step in the iterative procedure and proposes a new sample path $\mathbf{x}^{j+1} = \{x_k^{j+1}\}_{k=0}^N$. This is done by simulating, forward in time, a fictitious time deterministic system:

$$\frac{dx_k^j}{d\tau} = p_k \; , \tag{7.13}$$

$$\frac{dp_k}{d\tau} = -\frac{\partial \mathcal{H}(x_k^j, p_k)}{\partial x_k^j} \; , \tag{7.14}$$

where $p_k \sim \mathcal{N}(0,1)$ are the fictitious momentum variables assigned to each state variable x_k, resulting in a finite size random vector $\mathbf{p} = \{p_k\}_{k=0}^N$. The Hamiltonian of the system $\mathcal{H}(\mathbf{x},\mathbf{p})$ is:

$$\mathcal{H}(\mathbf{x},\mathbf{p}) = E_{pot} + E_{kin} \; , \tag{7.15}$$

where $E_{pot} = -\ln p_{post}(\mathbf{x}_{0:N}|\mathbf{y}_{1:K})$, is the potential energy given by the negative log-posterior distribution of the target density (see Equation 4.3), associated with the dynamics of the system (SDE) including the observations and $E_{kin} = \frac{1}{2}\mathbf{p}\mathbf{p}^\top$ is the kinetic energy.

In practice, the deterministic Equations (7.13) and (7.14), are discretised with a time step $\delta\tau$ and numerically solved with a *leapfrog* integration scheme:

$$\mathbf{p}(\tau + 0.5\varepsilon) = \mathbf{p}(\tau) - 0.5\varepsilon\nabla_{\mathbf{x}} L(\mathbf{x}(\tau)) \; , \tag{7.16}$$

$$\mathbf{x}(\tau + \varepsilon) = \mathbf{x}(\tau) + \mathbf{p}(\tau + 0.5\varepsilon) \; , \tag{7.17}$$

$$\mathbf{p}(\tau + \varepsilon) = \mathbf{p}(\tau + 0.5\varepsilon) - 0.5\varepsilon\nabla_{\mathbf{x}} L(\mathbf{x}(\tau + \varepsilon)) \; , \tag{7.18}$$

where ε is the artificial step size and $\nabla_{\mathbf{x}} L(\mathbf{x}) = -\frac{\partial \mathcal{H}(\mathbf{x},\mathbf{p})}{\partial \mathbf{x}}$. For full details of the algorithm concerning ergodicity of the chain and detailed balanced, see Neal (1996).

Finally, once the chain has converged to its stationary distribution, a large number of (discretised) sample paths is collected and the mean and covariance is computed as:

$$\mathbf{m}_t^S = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_t^n \, , \tag{7.19}$$

$$\mathbf{S}_t^S = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}_t^n - \mathbf{m}_t^S)(\mathbf{x}_t^n - \mathbf{m}_t^S)^\top \, , \tag{7.20}$$

where $\mathbf{m}_t^S \in \Re^D$ and $\mathbf{S}_t^S \in \Re^{D\times D}$ are respectively the *Smoothed* mean and covariance at time 't'.

7.2.4 Full weak constraint 4D-Var

As described earlier (Chapter 2), the *4D-Var* method minimizes a cost function that measures the distance of the most probable trajectory from the observations, within a predefined time window of inference. In most operational implementations the model equations are assumed perfect (strong constraint), or that the errors are sufficiently small to be ignored. In this work the model is assumed to be known only approximately, hence allowing for model error to exist in the problem formulation. This formulation is known as “*weak constraint 4D-Var*”.

Tremolet (2006), describes different variations of this algorithm, with the one closer to our approach denoted in his work, as “$4D-Var_x$”, where the subscript “x” denotes the control variable in the optimisation procedure. In the 4D-Var method implemented here since every (discrete in time) system state $\mathbf{x}_k$ is a control variable is also referred as “*full weak constraint 4D-Var*”.

Although the original 4D-Var method is well studied for estimating the states of a system, not much work has been done in estimating model parameters. Navon (1997) provides a useful review for parameter estimation, in the context of meteorology and oceanography. Here a dual approach is followed, similar to the LP approximation algorithm. The estimation framework is based on an outer / inner optimisation loop. The inner loop estimates the most probable trajectory, given the current estimates for the drift and diffusion parameters and subsequently the outer loop, conditioning on the most probable trajectory, updates the estimates of the parameters by taking a gradient descent step. The cost function to optimize is given by:

$$J_{cost} = J_{\mathbf{x}_0} + J_f + J_{obs} + J_{hp} + C_{\mathbf{\Sigma}} \, , \tag{7.21}$$

where $J_{\mathbf{x}_0}$, is the contribution of the prior over the initial state $\mathbf{x}_{k=t0}$, J_f is the influence of the model equations (drift function), J_{obs} is the contribution of the observations, J_{hp} comes from the priors over the (hyper-) parameters and $C_{\mathbf{\Sigma}}$ is a constant value that depends on the system noise coefficient $\mathbf{\Sigma}$. In practice, one needs to compute the gradients of the cost function with respect to the control variables (i.e. $\nabla_{\mathbf{x}_{0:N}} J_{cost}$), for estimating the most probable trajectory (inner loop) and then the gradients of the cost function with respect to the (hyper-) parameters (i.e. $\nabla_{\boldsymbol{\theta}} J_{cost}$ and $\nabla_{\mathbf{\Sigma}} J_{cost}$), for updating their values in the outer optimization loop.

Constructing the weak constraint 4D-Var cost function

In a Bayesian framework, if one is interested in estimating the system states $\mathbf{x}$ as well as the model parameters [1] $\mathbf{\Theta}$, then one is interested in the joint posterior distribution of the states and the parameters, given the observations (i.e. $p(\mathbf{x},\mathbf{\Theta}|\mathbf{y})$). Via Bayes rule this posterior is given by:

$$\begin{aligned} p(\mathbf{x},\mathbf{\Theta}|\mathbf{y}) &= \frac{p(\mathbf{y}|\mathbf{x},\mathbf{\Theta})p(\mathbf{x}|\mathbf{\Theta})p(\mathbf{\Theta})}{p(\mathbf{y})} \;, \\ &\propto p(\mathbf{y}|\mathbf{x},\mathbf{\Theta})p(\mathbf{x}|\mathbf{\Theta})p(\mathbf{\Theta}) \;, \end{aligned} \tag{7.22}$$

where $p(\mathbf{y}|\mathbf{x},\mathbf{\Theta})$ is the likelihood of the observations given the current state of the system and the (hyper-) parameters, $p(\mathbf{x}|\mathbf{\Theta})$ is the prior distribution over the system states given the the (hyper-) parameters, $p(\mathbf{\Theta})$ is the prior over the (hyper-) parameters and $p(\mathbf{y})$ is the marginal likelihood.

Having discretise the continuous time sample path $\{\mathbf{x}_t,\, t_0 \le t \le t_f\}$, using the Euler-Maruyama method (Kloeden and Platen, 1999), the next step is to compute the following joint posterior distribution of the states with the desired (hyper-) parameters:

$$p(\mathbf{x}_{0:N},\mathbf{\Theta}|\mathbf{y}_{1:K}) \propto \underbrace{p(\mathbf{y}_{1:K}|\mathbf{x}_{0:N},\mathbf{\Theta})}_{likelihood}\underbrace{p(\mathbf{x}_{0:N}|\mathbf{\Theta})}_{prior}\underbrace{p(\mathbf{\Theta})}_{prior} \;, \tag{7.23}$$

where $N = |t_0 - t_f|/\delta t$, is the total number of discrete state variables and K denotes the total number of observations.

Likelihood of the observations

Assuming that the measurements are i.i.d. with zero mean and covariance matrix $\mathbf{R}$, the likelihood expression for the observations yields:

$$\begin{aligned} p(\mathbf{y}_{1:K}|\mathbf{x}_{0:N},\mathbf{\Theta}) &= \prod_{k=1}^{K} \mathcal{N}(\mathbf{y}_k - \mathbf{x}_{t_k}|\mathbf{R}) \;, \\ &= \prod_{k=1}^{K} (2\pi)^{-D/2}|\mathbf{R}|^{-1/2}\exp\{-0.5(\mathbf{y}_k - \mathbf{x}_{t_k})^\top\mathbf{R}^{-1}(\mathbf{y}_k - \mathbf{x}_{t_k})\} \;, \\ &= \left[(2\pi)^{-D/2}|\mathbf{R}|^{-1/2}\right]^K \exp\{-0.5\sum_{k=1}^{K}(\mathbf{y}_k - \mathbf{x}_{t_k})^\top\mathbf{R}^{-1}(\mathbf{y}_k - \mathbf{x}_{t_k})\} \;, \end{aligned} \tag{7.24}$$

where the dependency on $\mathbf{\Theta}$ comes through the sample path $\mathbf{x}_{0:N}$ and all the assumptions about the state and observation vector dimensions are the same as introduced in Chapter 4. In addition the observations $\mathbf{y}_k$ are assumed direct measurements of the states $\mathbf{x}_{t_k}$, therefore the observation operator $h(\cdot)$ is omitted.

[1] Within the current framework Θ includes all the parameters in the drift and the system noise covariance matrix (i.e. $\Theta = \{\boldsymbol{\theta},\mathbf{\Sigma}\}$).

Prior distribution over the states

Using the assumption that the process is Markovian, the prior distribution of the states is given by:

$$\begin{aligned}
p(\mathbf{x}_{0:N}|\boldsymbol{\Theta}) &= p(\mathbf{x}_0)\prod_{k=0}^{N-1} p(\mathbf{x}_{k+1}|\mathbf{x}_k)\ , && (7.25)\\
&= p(\mathbf{x}_0)\prod_{k=0}^{N-1} \mathcal{N}(\mathbf{x}_{k+1}|\mathbf{x}_k + \mathbf{f}(\mathbf{x}_k;\boldsymbol{\theta})\delta t, \boldsymbol{\Sigma}\delta t)\ , && (7.26)\\
&= p(\mathbf{x}_0)\left[(2\pi)^{-D/2}|\boldsymbol{\Sigma}\delta t|^{-1/2}\right]^N \times \\
&\prod_{k=0}^{N-1} \exp\{-0.5(\delta\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k;\boldsymbol{\theta})\delta t)^\top(\boldsymbol{\Sigma}\delta t)^{-1}(\delta\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k;\boldsymbol{\theta})\delta t)\}\ , && (7.27)\\
&= p(\mathbf{x}_0)\left[(2\pi)^{-D/2}|\boldsymbol{\Sigma}\delta t|^{-1/2}\right]^N \times \\
&\exp\{-0.5\delta t\sum_{k=0}^{N-1}(\frac{\delta\mathbf{x}_{k+1}}{\delta t} - \mathbf{f}(\mathbf{x}_k;\boldsymbol{\theta}))^\top\boldsymbol{\Sigma}^{-1}(\frac{\delta\mathbf{x}_{k+1}}{\delta t} - \mathbf{f}(\mathbf{x}_k;\boldsymbol{\theta}))\}, && (7.28)
\end{aligned}$$

where $\delta\mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\delta t = t_{k+1} - t_k$. The initial state $\mathbf{x}_0$, is chosen to be normally distributed such as $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\tau}_0, \boldsymbol{\Lambda}_0)$. Notice also the unusual scaling of the system noise coefficient $\boldsymbol{\Sigma}$, with the time increment δt. This comes from the discrete version of the SDE (Equation 6.6), where the scaling is necessary to achieve the limit of the diffusion process as $\delta t \to 0$ (see Chapter 2).

Prior distribution over the parameters

For this prior density it is assumed that the parameters have no dependencies between them, hence their joint density can be expressed as the product of their marginal densities:

$$\begin{aligned}
p(\boldsymbol{\Theta}) &= p(\boldsymbol{\theta}, \boldsymbol{\Sigma})\ ,\\
&= p(\boldsymbol{\theta})p(\boldsymbol{\Sigma})\ , && (7.29)
\end{aligned}$$

where $p(\boldsymbol{\theta})$ is the prior marginal distribution of the drift parameters and $p(\boldsymbol{\Sigma})$ is the same but for the system noise coefficient. The derivations of these expressions are not further extended because these densities can be parametrized with any distribution of choice. In this 4D-Var setting the same prior distributions as in the HMC and the variational framework are used. That is $p(\boldsymbol{\theta}) = \mathcal{G}(\alpha,\beta)$ and $p(\boldsymbol{\Sigma}) = \mathcal{G}^{-1}(a,b)$.

J_{cost} - Total cost function

It is common practice in optimisation when one wants to find the minimum (or maximum), of a cost function to look for the minimum (or maximum) of the logarithm of the cost function (due to the monotonicity of the logarithmic function). Hence instead of maximizing the posterior

$p(\mathbf{x}_{0:N}, \mathbf{\Theta}|\mathbf{y}_{1:K})$, one can minimize the negative $\ln p(\mathbf{x}_{0:N}, \mathbf{\Theta}|\mathbf{y}_{1:K})$, which has some nice characteristics. Therefore, the complete cost function is given by:

$$\begin{aligned}
J_{cost} = \quad - \quad & \underbrace{\ln p(\mathbf{x}_0)}_{J_{\mathbf{x}_0}} + \underbrace{0.5\delta t \sum_{k=0}^{N-1} \left(\frac{\delta \mathbf{x}_{k+1}}{\delta t} - \mathbf{f}(\mathbf{x}_k;\boldsymbol{\theta}) \right)^{\top} \mathbf{\Sigma}^{-1} \left(\frac{\delta \mathbf{x}_{k+1}}{\delta t} - \mathbf{f}(\mathbf{x}_k;\boldsymbol{\theta}) \right)}_{J_f} \\
+ \quad & \underbrace{0.5 \sum_{k=1}^{K} (\mathbf{y}_k - \mathbf{x}_{t_k})^{\top} \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{x}_{t_k})}_{J_{obs}} \underbrace{- \ln p(\boldsymbol{\theta}) - \ln p(\mathbf{\Sigma})}_{J_{hp}} \\
+ \quad & \underbrace{0.5\,(K\ \ln|\boldsymbol{R}| + N\ \ln|\mathbf{\Sigma}\delta t| + K\ N\ D\ \ln(2\pi)\ )}_{C_{\mathbf{\Sigma}}}\,,
\end{aligned} \tag{7.30}$$

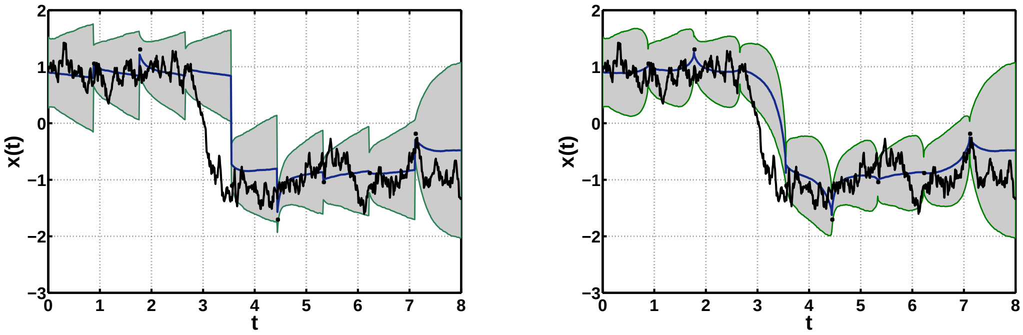
where $K > 0$ is the total number of observations, $N > 0$ is the number of the discrete time states and $D > 0$ is dimensions of the system states and observations. It is worth noticing that unlike most implementations of weak constraint 4D-Var, where the term $C_{\mathbf{\Sigma}}$ is omitted (because in state estimation this constant vanishes), in this setting it is important to include it if one wants to perform the estimation of the system noise $\mathbf{\Sigma}$ properly.

7.3 State estimation

This section compares the estimation methods described in Section 7.2, on a typical DW realisation. This is done for illustration purposes to demonstrate the different results obtained with each algorithm. The time window is $T = [0, 8]$ with discretisation step $\delta t = 0.01$. The drift and diffusion parameters are set to $\theta = 1$, and $\sigma^2 = 0.8$ and $K = 8$ observations are measured (one per time unit) with noise variance $R = 0.04$. For simplicity, all filtering and smoothing algorithms (except the HMC), start with identical fixed initial values for the marginal means and variances ($\mathbf{m}_0 = \mathbf{x}_0$ and $\mathbf{S}_0 = 0.1$).

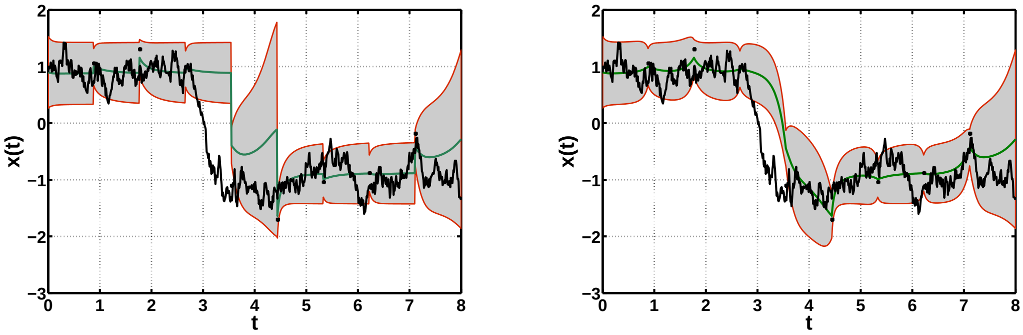
Figures 7.2(a) and 7.2(b) present the results of the ensemble Kalman filter and smoother respectively, with 5000 ensemble members. The reason for choosing so high ensemble size is to provide a smooth solution. As expected, in the absence of observations the EnKF increases the variance fast and only at observation times it reduces it rapidly. The EnKS does better, producing a smoother approximation to the mean path and does not overestimate the uncertainty like the EnKF. Observe that both algorithms at the final time are identical, as should be expected. Nevertheless, both algorithms perform poorly in tracking the transition between the two wells. Ensemble Kalman methods provide an effective means to approximate the evolution of the probability distributions for non-linear dynamics, however as discussed in Miller et al. (1999), problems still remain in properly tracking transitions in systems with multi-modal statistics.

Similar remarks can be made for Figures 7.3(a) and 7.3(b), where the results from the unscented methods are illustrated. In general the estimation of the variance is more conservative,



(a) Ensemble Kalman filter (b) Ensemble Kalman smoother

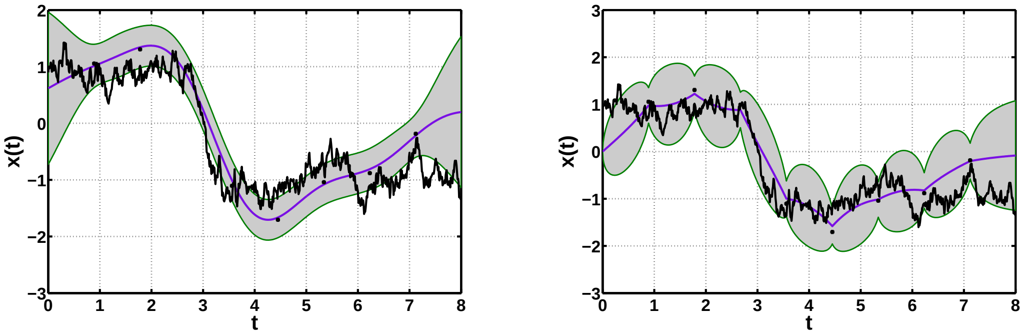
Figure 7.2: Application of the ensemble Kalman filter (a) and smoother (b) to a single dataset (8 noisy observations) of a typical DW realisation. Both algorithms use $5,000$ ensemble members to approximate the filtering and smoothing distribution respectively. Continuous (smooth) blue lines indicates the mean paths and the shaded areas the variances. In both plots the true history that generated the observed data is plotted on top of the predictive results (black rough trajectory).



(a) Unscented Kalman filter (b) Unscented Kalman smoother

Figure 7.3: (a) and (b), same as 7.2(a) and 7.2(b), but with the unscented algorithms.

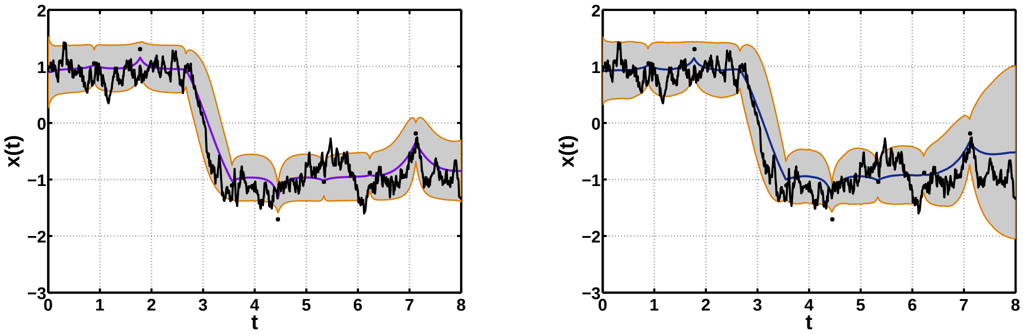
compared to their ensemble counterparts (Figs. 7.2(a) and 7.2(b)) and only at the transition time is the variance overestimated.



(a) GPr - Squared Exponential kernel (b) GPr - Ornstein-Uhlenbeck kernel

Figure 7.4: Gaussian process regression (GPr) smoothing. For (a) the squared exponential (SE) kernel is used, whereas for (b) the stationary OU kernel. The dataset is the same as Fig. 7.3(a).

Although not described thoroughly in this thesis, Gaussian Process Regression (GPR) smoothing is a common tool in machine learning to perform inference in unobserved data. Rasmussen and Williams (2006), provide a detailed study on this subject and also give implementation details. Here, the GPR is used with two different kernels, these are the Squared Exponential kernel $K(x_t, x_s) = \exp\{-0.5|x_t - x_s|^2\}$ and the non-stationary OU kernel Eq. (3.7). Both Figures 7.4(a) and 7.4(b), show that the GPR produce very smooth results for the approximate mean paths. Clearly both of the kernels used are not appropriate to perform inference for this system. Note also how the GPR using the OU kernel overestimates the posterior covariance. However, this is not a surprising result as GPR does not assume any dynamics in the underlying system that generated the observed data. The only case with the GPR can be used to perform exact inference in SDEs is the OU system (as shown in Chapter 3), where the transition pdf is given by the OU kernel.



(a) VGPA algorithm

(b) Markov chain Monte Carlo

Figure 7.5: Original VGPA (a) vs HMC sampling (b). The HMC for this example is considered as the reference solution.

The last two figures of this section (Figs. 7.5(a) and 7.5(b)), provide the results from the original version of the VGPA algorithm in contrast with the HMC posterior sampling algorithm. For this example, the HMC is assumed to provide the reference solution. Compared to all the above methods, the VGPA seems to provide a very good approximation to the posterior process. The variance is slightly underestimated, compared to the HMC results, but the mean path matches the HMC mean path rather well and the transition is tracked accurately. An obvious difference with the HMC is at the end of the time window (after the last observation) where the mean paths start to deviate and the variance is strongly underestimated. This is explained by the fact that the linear approximation that is imposed by the VGPA, in the absence of observations, restricts the algorithm and prevents it from “seeing” the other well of the system. The VGPA focuses on the correct posterior mode as long as there are available observations and when the observations stop it remains in the last visited mode keeping the variance appropriately fixed. On the contrary the HMC in the absence of observations becomes “confused” and is unsure on which mode to stay, therefore it returns to the true mean of the equilibrium process (which for this system is $\mathbf{x}_t = 0$, see Figure 3.2(a)) and its variance diffuses in both wells.

7.4 Parameter estimation

This section presents an empirical comparison of the marginal and joint estimation of the drift $\boldsymbol{\theta}$ and diffusion coefficient $\boldsymbol{\Sigma}$, using the UnKF, 4D-Var and LP methodologies in two distinct asymptotic regimes: **(a)** *infill asymptotics*, where the observations are sampled more and more densely, within a fixed time domain (i.e. $N_{obs} \to \infty$, while $T = [t_0, t_f]$) and **(b)** *increasing domain asymptotics*, where the observation density remains fixed, whilst the time window of inference increases (i.e. N_{obs} = const. and $T \to \infty$).

7.4.1 Infill asymptotic behaviour

Before proceeding a few issues need to be clarified concerning the presentation of the results. As mentioned earlier the variational LP approximation method and the weak constraint 4D-Var based algorithm, provide point estimates of the (hyper-) parameters, in a gradient based optimisation framework. The dual unscented Kalman filter approach provides mean estimates (of the parameters), as a function of time. To make the results of the dual UnKF more comparable with those from the other two methods the collection of the mean estimates is treated as a (filtered) distribution and then estimates of its moments, such as the mean value (Hansen and Penland, 2007), are calculated. An example of this procedure is shown in Figure 7.6, where the dual UnKF is applied to estimate the drift parameter of the DW system, on a single data set. As a general rule, only the second half of the time period is used to estimate the mean values. The rational is that in these controlled experiments[1] there is no need to average over the whole time window because the initial estimated value is wrong, hence the filter is allowed to converge around a value. The second remark has to do with the quantities that are plotted. In order to provide a more general analysis thirty different observation noise realisations were created, for each observation density. The results are presented as summary statistics, illustrated using the 25'th, 50'th (or median value) and the 75'th percentile of the estimated values from each algorithm.

The conditional[2] drift estimation of the OU and DW systems is shown in Figures 7.7(a) and 7.7(b) respectively. The results for the OU system show that the LP approximation has a small increasing trend and settles to a higher value, compared with 4D-Var, although this higher value is also seen in the HMC posterior estimates of this parameter (Fig. 6.14(b)). Also both algorithms narrow the range of estimates, as the observation density increases (the error bars are closer to the median value), as one would expect. On the other hand the results from the UnKF based algorithm,

[1] Here it is implied that the true values that generated the data are known a priori and also the initial values of the estimation process are deliberately wrong but close to the true one.

[2] This term is used to signify that all the other parameters, such as the system and observation noises ($\boldsymbol{\Sigma}$ and $\mathbf{R}$), are assumed known and fixed to their true values.

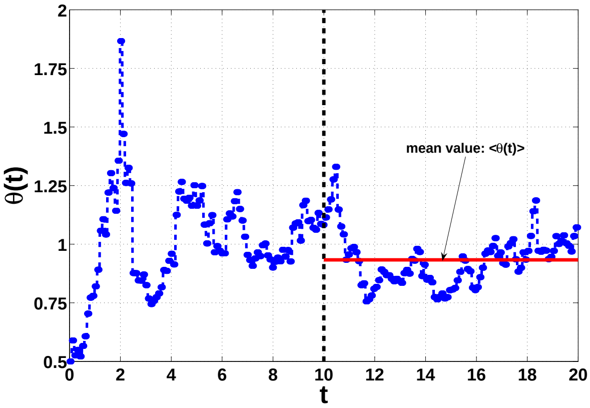
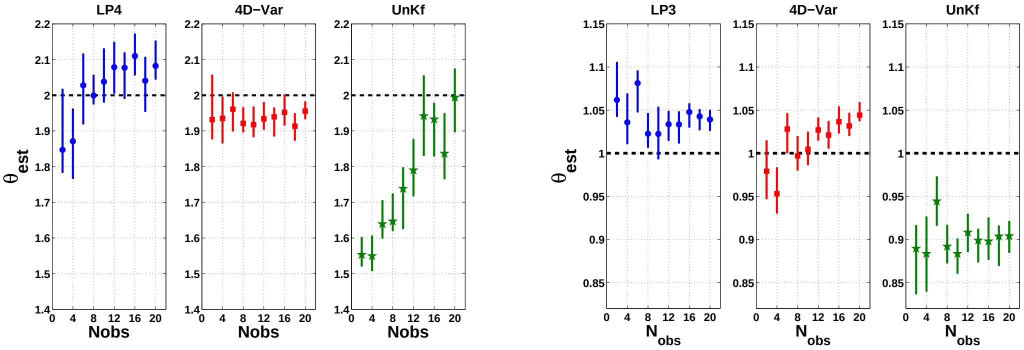


Figure 7.6: An example of mapping the results from the application of the dual UnKF algorithm applied to a single trajectory, estimating the DW drift parameter, to a point estimate (mean value). The blue circles indicate the ensemble mean estimates as a function of time, while the continuous red line is the mean value of these estimates over the period used for averaging. The vertical dashed line marks the beginning of the time window where the average takes place.

show a more steep trend and only when the system is highly observed are the estimates close to the true generating value. Here, as in all the experiments that follow, all three algorithms were initialized with the same value for the parameter(s) that were estimated.

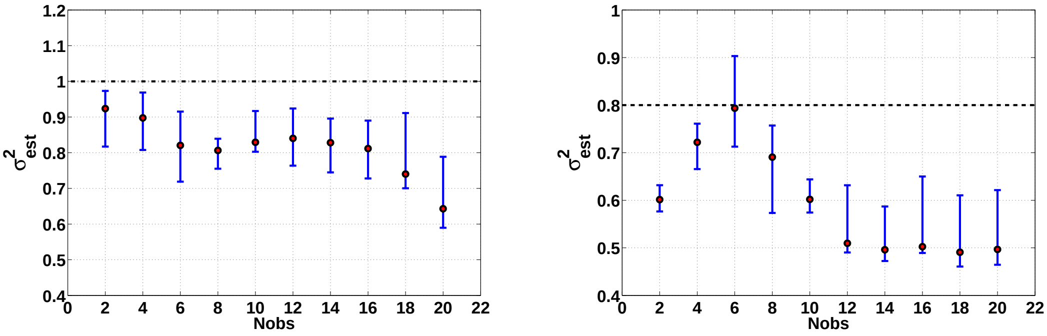


(a) θ_{OU} estimation (b) θ_{DW} estimation

Figure 7.7: *Drift (conditional) estimation:* (a) Presents the summary statistics (25'th, 50'th and 75'th percentiles) after estimating the drift parameter θ from thirty different realizations, of the observation noise, on the OU system keeping the system noise coefficient σ^2 fixed to its true value. The left panel (blue) presents the results from the *LP* algorithm, while the middle (red) and the right (green) the results from the *(full) weak-constrained 4D-Var* and the *dual UnKF* respectively. In (b) the same estimation experiment is repeated but for thirty different realizations, of the observation noise, of the DW system. All estimation results are presented as functions of increasing observation density.

For the DW system the algorithms were more stable, in the sense that they converge to a stable value and there are no major trends as in the OU case. The results from all methods are biased either towards higher values (LP and 4D-Var), or lower values (UnKF). Once again the LP algorithm bias matches the HMC posterior estimates as shown in Fig. 6.16(b). Although the results from the dual UnKF seem inferior compared to the other two algorithms, it should be recalled that this is a filter estimation, which means that it “sees” the observations sequentially, only up to the current time and does not take into account the future observations.

Figures 7.8(a) and 7.8(b), present the results of estimating the system noise σ^2, of the OU and DW systems. This shows only the estimates obtained from the LP approximation method. The other methods, although applied to the same datasets, were unable to provide good estimates, hence were omitted. It is obvious that the estimation for the OU system is stable, while for the DW the process needs to be well observed (e.g. $N_{obs} \geq 10$), before converging to a value. Both plots show consistency with the HMC posterior estimates presented in Chapter 6.



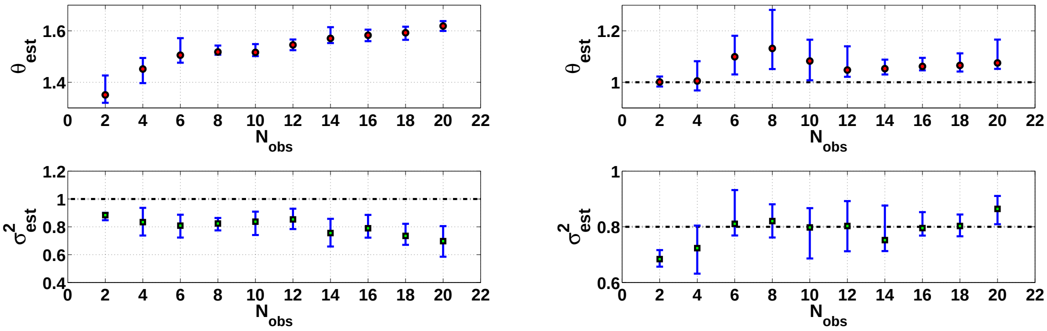
(a) σ^2_{OU} estimation

(b) σ^2_{DW} estimation

Figure 7.8: *Noise (conditional) estimation:* (a) shows the conditional estimation of the system noise coefficient σ^2, keeping θ at its true value. The plot presents the 50’th percentile (red circles) and the 25’th to 75’th percentiles (blue vertical lines). (b) repeats the same experiment but for the DW system. All results were obtained with the *LP* method (3’rd order) and are presented as functions of increasing observation density.

The experiments on the uni-variate systems conclude with the joint estimation of the drift parameter θ and the system noise coefficient σ^2. Figures 7.9(a) and 7.9(b), summarize the results attained from the LP approximation method. The drift estimation for the OU system, shows a significant bias to smaller values (compared with the conditional estimation of Fig. 7.7(a)), where the bias was towards a higher value. These estimates become more confident as the observation density increases (smaller error bars). Meanwhile, the estimation of the OU diffusion noise is consistent with the conditional outcomes. Unlike the OU system, the DW shows consistent estimation for the drift parameter and a surprising improvement of the system noise estimation. In these plots, in contrast to the conditional ones, there cannot be a direct reference to the posterior HMC estimates, because here the parameters are estimated simultaneously, while the results of the HMC, in Chapter 6, were obtained by fixing the parameters that are not estimated to their true values.

Next the conditional estimation of the drift vector $\boldsymbol{\theta}$, of the L3D system is considered (Figure 7.10). It is clear that in this example the 4D-Var estimation method (middle column), performs better and produces more stable results. The LP algorithm when tested with 4 and 6 observations



(a) OU system (b) DW system

Figure 7.9: *Joint estimation:* In (a) the drift and diffusion coefficient, of the OU system, are estimated jointly. The left upper panel shows the results for θ, while the left lower panel for σ^2. The results are summaries (25'th, 50'th and 75'th percentiles) from thirty different observation realizations. (b) shows the same joint estimation but for the DW system. The right upper panel shows the results for θ, while the right lower panel for σ^2. All results were obtained with the *LP* method (3'rd order) and are presented as functions of increasing observation density.

per time unit seems to be under-sampled and the state estimation (inner loop of the optimisation procedure) does not actually converge to the optimal posterior process. Therefore, the parameter estimates are no longer reliable. When the process is observed more frequently (e.g. $N_{obs} \geq 8$), it produces more stable results. The dual UnKF estimation results are reliable, with the exception of the ρ parameter (third column, second row), which is very biased with sparse observations. However, all parameters asymptotically converge close to the true values, as the observation density increases.

Similar to the univariate systems, the conditional estimation of the system noise coefficient $\mathbf{\Sigma}$, was feasible only with the variational LP approximation algorithm. Because the covariance matrix is assumed diagonal (see Eq. 4.1), one needs to estimate only the three diagonal components, which correspond to the noise added in each dimension of the L3D dynamical equations (see Eq. 3.9). Figure 7.11 suggests that to estimate this very important parameter one has to have dense observations. For the L3D system all three dimensions are observed. Components σ_x^2 and σ_z^2 converge close to the true values roughly after 16 observations, per time unit, while the σ_y^2 parameter converges to a higher value. These results are in agreement with the approximate marginal profiles produced earlier (Fig. 6.18(b)).

To conclude with the *infill asymptotics* section, the application of the newly proposed LP approximation framework is demonstrated to the joint estimation of the drift and diffusion matrix of the L3D system. In total six (hyper-) parameters (σ, ρ, β, σ_x^2, σ_y^2 and σ_z^2), are estimated as shown in Figure 7.12. The asymptotic behaviour is similar to that observed when estimating the parameters conditionally, which gives some level of confidence that the algorithm is stable. The general message is that good estimates can be achieved when the system is well observed.

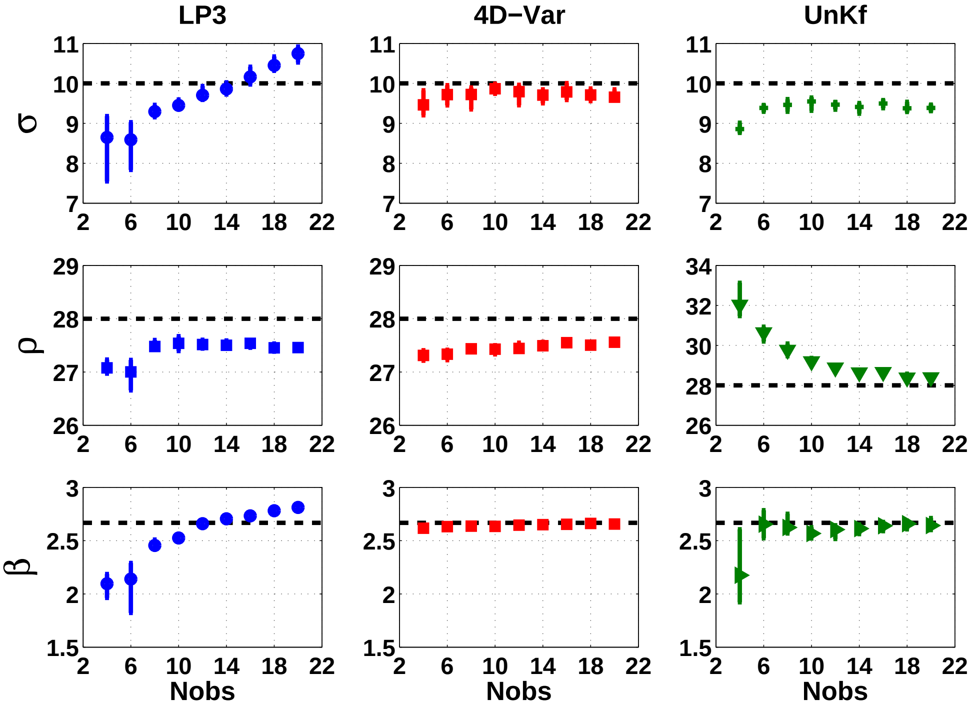


Figure 7.10: *Drift (conditional) estimation:* The infill asymptotic results for the *L*3*D* drift parameter vector $\boldsymbol{\theta}$. The summary results when seen horizontally compare the same drift parameter but with different estimation methods, while vertically the results are presented for the same estimation method but for all three parameters (σ, ρ and β). The methods tested, from left to right are the *LP* algorithm (3'rd order), the *(full) weak-constraint 4D-Var* and the *dual UnKF* accordingly. In all sub-plots the horizontal dashed lines indicate the true values of the drift parameters that generated the observed trajectories. Where possible the *y-axis* was kept the same for all plots to make comparison easier. All algorithms were tested on the same thirty different realisations of the observation noise.

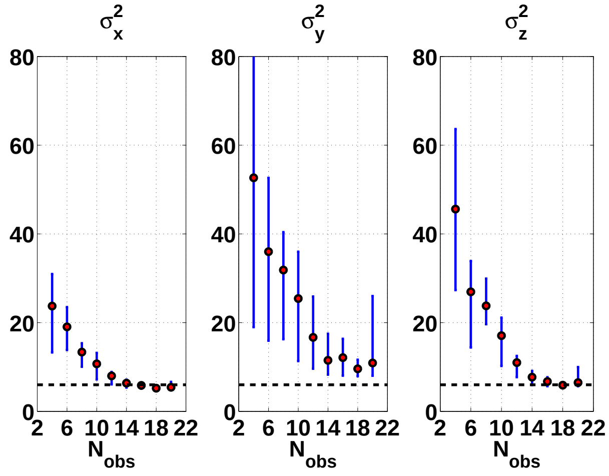


Figure 7.11: *Noise (conditional) estimation:* Summary results (25'th, 50'th and 75'th percentiles) from thirty different observation realizations, of the *L*3*D* system, when estimating conditionally the system noise coefficient matrix Σ. The results were obtained using the *LP* algorithm (3'rd order) and are presented as functions of increasing observation density. The estimation of the noise is presented separately in each dimension *x*, *y* and *z* from left panel to right accordingly.

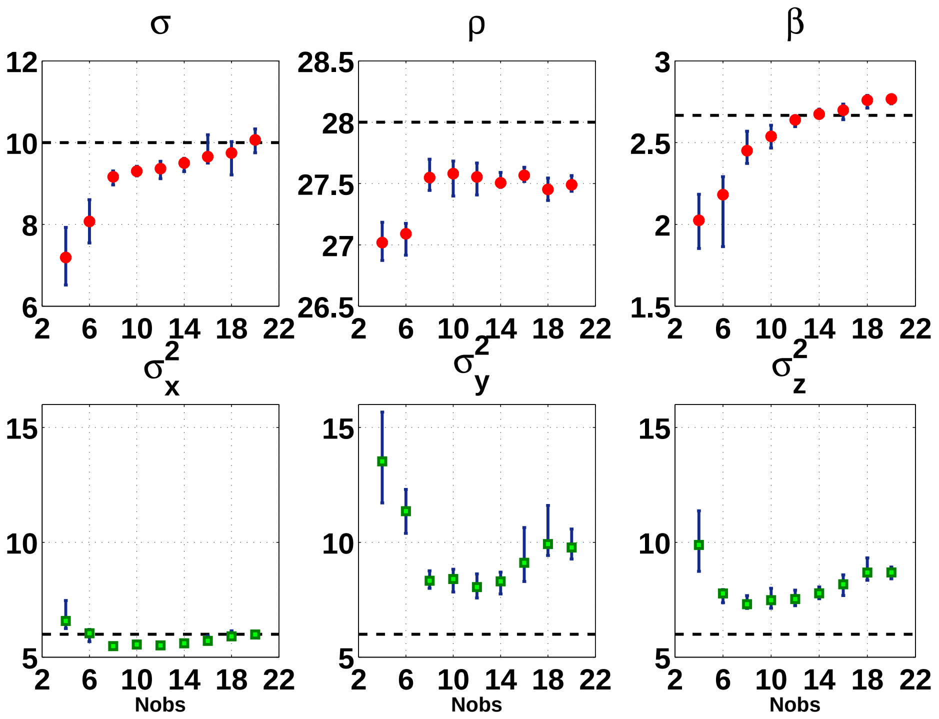


Figure 7.12: *Joint estimation:* The summary results (25'th, 50'th and 75'th percentiles) when estimating jointly the drift parameters σ, ρ and β (upper three panels), and the system noise coefficients σ_x^2, σ_y^2 and σ_z^2 (lower three panels), of the *L3D* system. The same dataset of the thirty different realisations of the observation noise is used, as in the previous experiments.

7.4.2 Increasing domain asymptotic behaviour

This section discusses another important asymptotic property; when the observation *density* remains fixed, but the duration that an event (or the random process) is observed, increases. To explore this behaviour new extended sample paths were created for all the dynamical systems considered in our previous simulations and then the total time-window was split into smaller, but equal, time intervals.

To be more precise, an example is given on the DW system. Figure 7.13, presents a sample path (or history) of the DW system with time-window $T_{total} = [0, 50]$. The next step consists of measuring the history with fixed observation density (e.g. $N_{obs} = 2$). Then the total time-window is divided in five sub-domains, of ten time units and create five time-windows ($T_{10} = [0, 10]$, $T_{20} = [0, 20]$, $\cdots$, $T_{50} = [0, 50]$), including the observations from the previous steps. Finally, the estimation methods are applied on each sub-interval, by introducing the new observations incrementally.

Figures 7.14(a) and 7.14(b), show the results of the conditional drift estimation for the OU and the DW systems respectively, as the time-window of inference increases. As in the *infill asymptotic* simulations, thirty different realizations of the observation noise were generated and the results are presented as summary statistics of the estimation outcomes. Because the number of

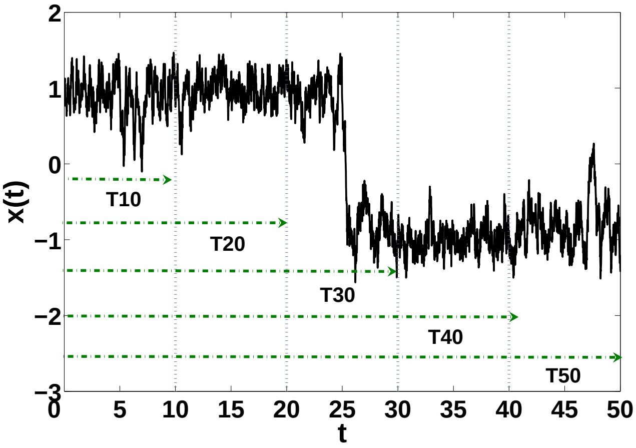
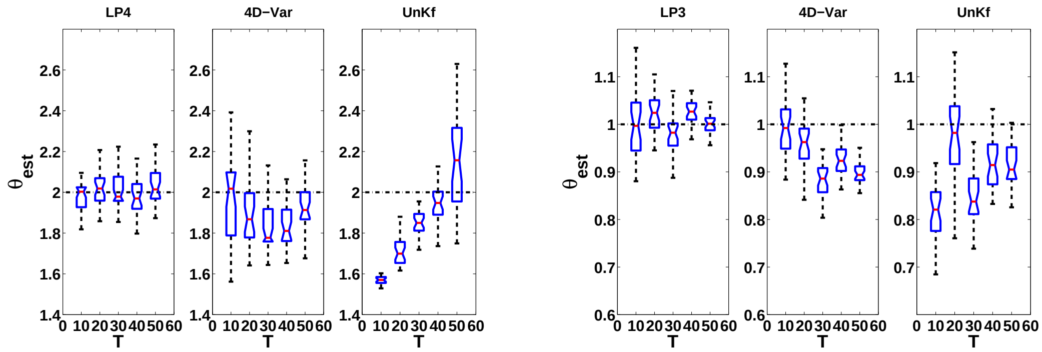


Figure 7.13: A typical example of a *DW* sample path with an extended time-window that is used for the *increasing domain* asymptotic behaviour of the algorithms. The vertical dotted lines split the total time window in five time domains starting from $T_{10} = [0,10]$ to $T_{50} = [0,50]$, which are presented to the estimation methods incrementally.

simulations performed are fewer than in the previous case all the results are presented as boxplots which provide a richer presentation. It is apparent that in this type of asymptotic convergence, the LP approximation algorithm is remarkably stable with results that are very close to the ones that generated the data. The drop under the true value (as indicated by the horizontal dashed line), in the DW example (Fig. 7.14(b)), for the third time window (i.e. $T_{30} = [0,30]$), can be explained by the fact that the transition between the two wells happens between the 24'th to 26'th time units, as shown in Figure 7.13. However, when the time-window increases further the algorithm recovers back to the previous value. For the same example, the 4D-Var method starts with a higher estimated value but after the transition occurs it settles to a lower value. A similar behaviour can also be observed for the UnKF results, were the method approaches the true value, although it becomes less confident (larger error bars), which was somewhat unexpected behaviour.

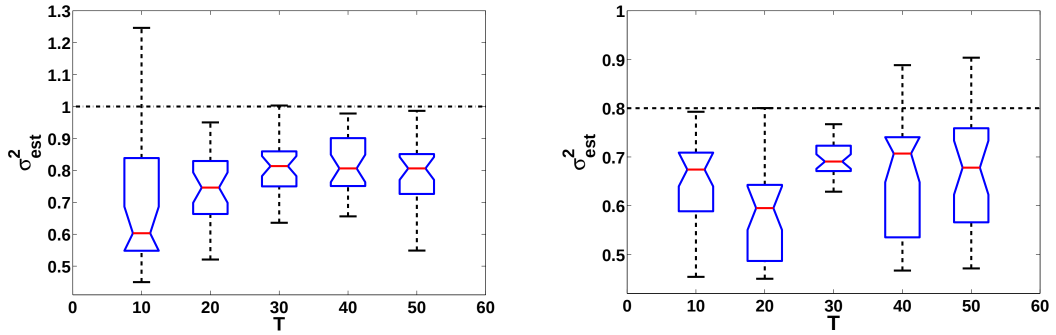
The conditionally estimated diffusion coefficients are presented in Figures 7.15(a), for the OU and 7.15(b), for the DW. Here only the LP approximation method was used, as in the previous section. The estimates, for both examples, are stable and improve as the time window increases. Especially for the DW, the results get closer to the true value after the transition has been observed (T_{30}). In a similar way, the results for the joint estimation of the drift θ and diffusion σ^2, are consistent and presented in Figures 7.16(a) and 7.16(b).

This section concludes with the results of the L3D system. Figure 7.17, presents the summaries of the jointly estimated drift parameter vector $\boldsymbol{\theta} = [\sigma\ \rho\ \beta]^\top$, conditional on the system noise matrix $\boldsymbol{\Sigma}$ set to its true value, from all three estimation methods. All algorithms are stable and produce good results, with the 4D-Var having the smallest bias. Once again, the 4D-Var and UnKF methods failed to provide stable results when estimating the system noise coefficients, hence only results



(a) θ_{OU} estimation (b) θ_{DW} estimation

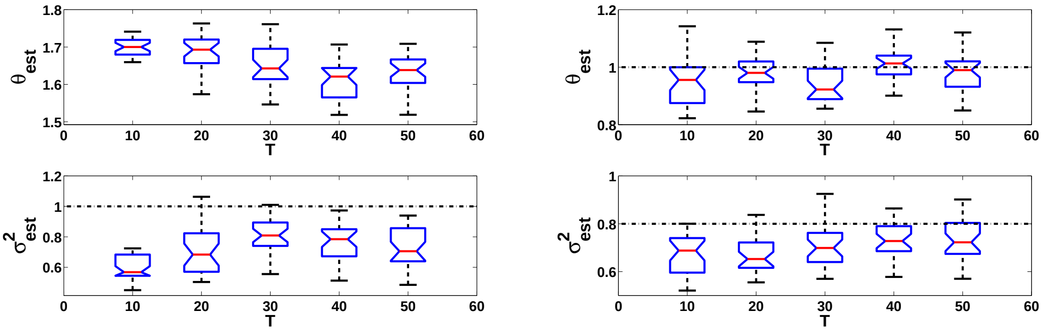
Figure 7.14: *Drift (conditional) estimation:* (a) Presents the summary statistics (boxplots) after estimating the drift parameter θ from thirty different realizations, of the observation noise, on the OU system keeping the system noise coefficient σ^2 fixed to its true value. The left panel presents the results from the *LP* algorithm, while the middle and the right the results from the *(full) weak-constrained 4D-Var* and the *dual UnKF* respectively. In (b) we repeat the same estimation experiment but for thirty different realizations, of the observation noise, of the DW system. All estimation results are presented as functions of increasing time domain, keeping the observation density fixed.



(a) σ^2_{OU} estimation (b) σ^2_{DW} estimation

Figure 7.15: *Noise (conditional) estimation:* (a) shows the conditional estimation of the system noise coefficient σ^2, keeping θ to its true value. The plot presents boxplots (5'th, 25'th, 50'th, 75'th and 95'th percentiles), from thirty different realizations, of the observation noise, of the OU system. (b) repeats the same experiment but for the DW system. All results were obtained with the *LP* method (3'rd order) and are presented as functions of increasing time domain, keeping the observation density fixed.

from the LP method are shown. The joint estimation of the noise coefficients σ_x^2, σ_y^2 and σ_z^2, conditional on the drift vector $\boldsymbol{\theta}$ being fixed to it true value, are illustrated at Figure 7.18, where it was necessary to observe with quite high density ($N_{obs} = 18$). In addition, the joint estimation of all the (hyper-) parameters, of the L3D system, as the time-window of inference increases, is shown in Figure 7.19. The results are in accordance with the conditional estimates, although the observation density was set to ten observations, per time unit (i.e. $N_{obs} = 10$).



(a) OU system (b) DW system

Figure 7.16: *Joint estimation:* In (a) the drift and diffusion coefficient, of the OU system, are estimated jointly. The left upper panel shows the results for θ, while the left lower panel for σ^2. The boxplots present summaries from thirty different observation realizations. (b) shows the same joint estimation but for the DW system. The right upper panel shows the results for θ, while the right lower panel for σ^2. All results were obtained with the *LP* method (3'rd order) and fixed observation density to two per time unit ($N_{obs} = 2$).

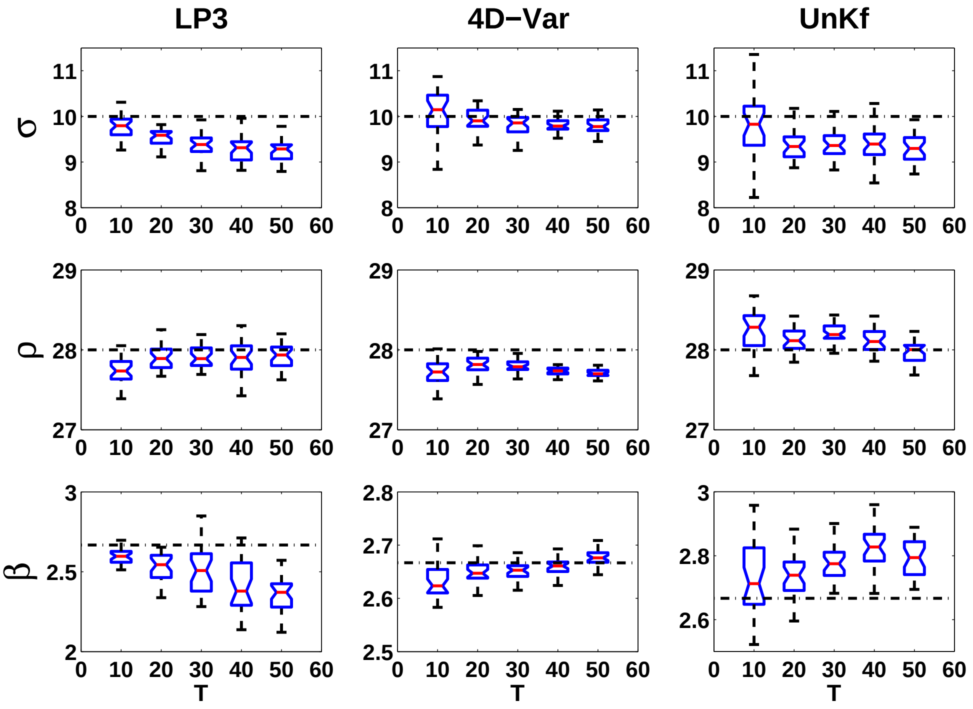


Figure 7.17: *Drift (conditional) estimation:* This plot compares the increasing domain asymptotic results (fixed observation density), when estimating the *L3D* drift parameter vector $\boldsymbol{\theta}$. The summary results when seen horizontally compare the same drift parameter with different estimation methods, while vertically the results are presented for the same estimation method and all three parameters (σ, ρ and β). The methods tested, from left to right are the *LP* algorithm (3'rd order), the *(full) weak-constrained 4D-Var* and the *dual UnKF* accordingly. In all sub-plots the horizontal dashed lines indicate the true values of the drift parameters that generated the history sample. Where possible the *y-axis* was kept the same for all plots comparing the same parameter to make the comparison easier.

7.5 Discussion

The methods implemented and presented here, cover all the main categories that deal with the inference problem from a Bayesian perspective (Chapter 2). In exploring the infill and increasing

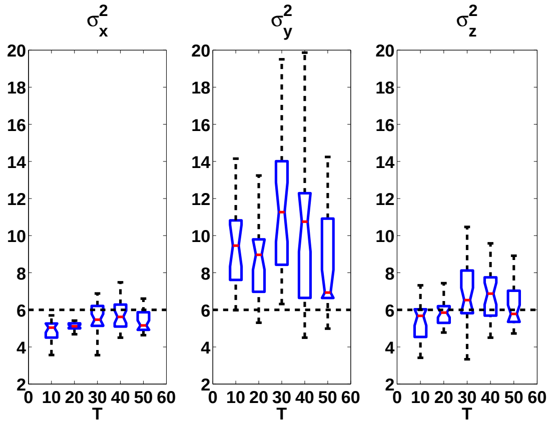


Figure 7.18: *Noise (conditional) estimation:* Summary results (boxplots) when estimating jointly the noise coefficients σ_x^2, σ_y^2 and σ_z^2, of the *L3D* system. The results were obtained with the *LP* method (3'rd order) and presented as functions of increasing time domain, keeping the observation density fixed ($N_{obs} = 18$).

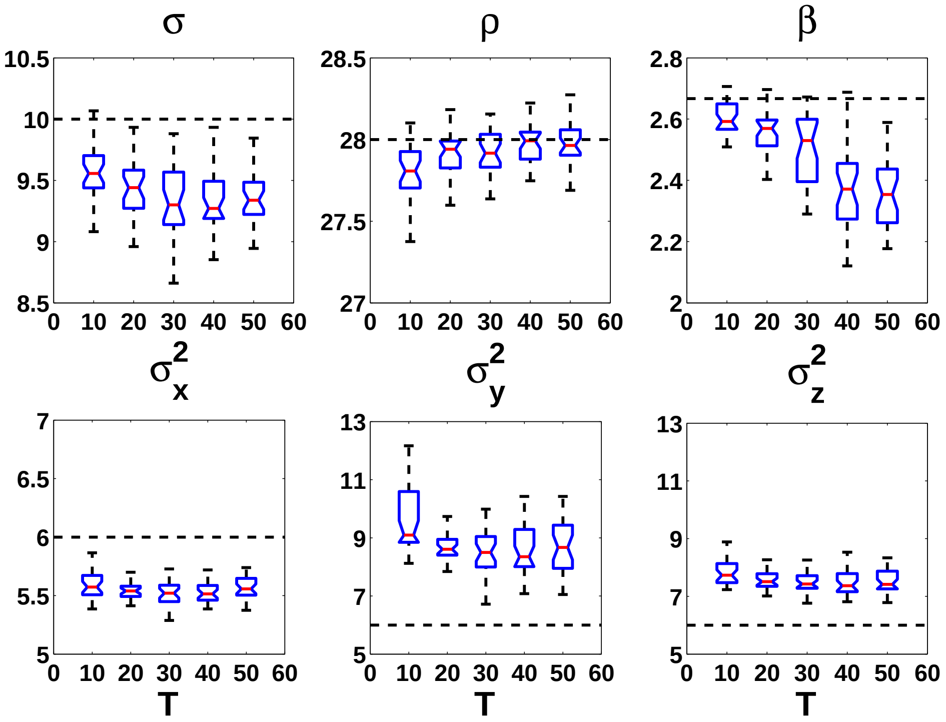


Figure 7.19: *Joint estimation:* Summary results (boxplots) when estimating jointly the drift parameters σ, ρ and β (upper three panels), and the system noise coefficients σ_x^2, σ_y^2 and σ_z^2 (lower three panels), of the *L3D* system. The results were obtained with the *LP* method (3'rd order) and are presented as functions of increasing time domain, keeping the observation density fixed ($N_{obs} = 10$).

domain behaviour when estimating the parameters of the OU, DW and L3D, all methods show biases and the response was different over the range of the systems.

The methods are largely comparable with the dual UnKF being less stable and slightly more biased. LP and weak constraint 4D-Var are more comparable (since both provide smoothing solutions to the inference problem) but there was no clear preference for a specific algorithm, except in the case of estimating the system noise parameters $\mathbf{\Sigma}$. In this case both 4D-Var and UnKF failed to provide satisfactory results, giving the LP a clear advantage. A particular difficult case is the noise estimation of the L3D system where the process has to be observed very frequently. Yet it is not clear whether this relates to the chaotic behaviour of the system rather the inability of the variational algorithm to identify these parameters.

Comparing the results on the two asymptotic regimes reveals that *increasing domain* is more promising than *infill* and suggests that in order to identify a model parameter, is better to observe an event constantly over a large period of time, rather than observe it more densely in a short period of time. It should be emphasised that the results shown for these experiments are obtained by applying the estimation algorithms on many realisations of the observation noise from a single trajectory of each dynamical system. The way that these conclusions generalise for other trajectories has yet to be answered.

An interesting question that is raised is how the parameter estimates are affected if the process is not observed uniformly (at equidistant times), as was the case here, but rather with different densities over different periods of time. An example, on a DW trajectory, would be the estimation of the system noise $\mathbf{\Sigma}$ by having more frequent observations around the transition time than the rest of the sample path.

8 Conclusions

CONTENTS

"As for me, all I know is that I know nothing."
— Socrates, Greek philosopher.

8.1 Foreword

This thesis has developed two new algorithms for approximate inference in partially observed diffusion processes, based on extending the recently proposed variational Gaussian process algorithm (VGPA), in terms of radial basis functions (RBF) and local polynomial (LP) approximations. Both extensions were tested on artificial datasets and are shown to converge to the original VGPA algorithm, given a sufficient number of basis functions or order of the polynomials, respectively. Although simple in concept, the LP method is a natural extension of the RBF approximation and from a theoretical point of view possess a more appropriate approximation of the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$. This was shown in practice where the number of optimised variables was reduced even further than the RBF extension while retaining an excellent approximation of the aforementioned parameters. This chapter concludes the thesis by summarising some key points of each chapter. At the end some further research directions are discussed along with the limitations of the proposed algorithms.

8.2 Summary

Stochastic differential equations have gained increasing attention the last decades, applied to the modelling of real world systems with many applications in physics, finance, environmental sciences, engineering and systems biology. Typically they describe the temporal evolution of a state vector of a dynamical system based on the (assumed) physical laws of the real system, including a driving noise process. The noise process can be thought of in various ways. It often represents processes not included in the model, but present in the real system. Chapter 1, describes briefly the difference between modelling a system with ordinary and stochastic differential equations. Moreover the rational for adopting a Bayesian paradigm for the developed algorithms is also highlighted.

The important class of diffusion processes is reviewed in Chapter 2, which can be seen as solutions of the aforementioned SDEs. Some basic definitions on stochastic processes are given with a few fairly simple examples and then the inference problem, addressed in this thesis, is explained. It is made clear why the estimation of model parameters with the classical Maximum-Likelihood estimation framework is a challenging task and how this leads to the development of approximate estimation techniques to tackle this problem. The relevant literature is reviewed, mainly from a Bayesian perspective, although for the sake of completeness the major non-Bayesian techniques are also highlighted.

Chapter 3 summarises and describes the dynamical systems that were chosen to test the algorithms developed. These vary from univariate linear (OU), to forty dimensional non-linear (L40D). The model equations of each system are defined properly and characteristic examples are given.

Providing approximate solutions to very difficult problems is not a new idea in Machine Learning. Chapter 4 describes an algorithm for approximate inference in partially observed diffusion processes, following the variational paradigm which approximates an intractable probability distribution 'p_t' by another one 'q_t' that belongs to a family of tractable distributions. This is done by minimising the $\mathrm{KL}[q_t \| p_t]$ divergence (Kullback and Leibler, 1951), between the true posterior process and the approximate one (i.e. between probability measures over continuous time sample paths). Unlike most other variational methods, a fully factorized posterior density $q(\mathbf{x}) = \prod_i q_i(\mathbf{x}_i)$ (such as the one assumed by the *Naive Mean Field* theory), does not make sense in an infinite (Markovian) setting. Moreover, as argued in Apte et al. (2007) it is preferable and mathematically correct to define the inference problem in an infinite dimensional (space of sample paths) setting and then look for efficient ways to discretise it. If the discretisation occurs first, so that the inference problem is never written down in continuous time, it may lead to a non-optimal approximation of the required infinite dimensional problem.

Another issue which relates to the Gaussian processes approximation, assumed by the variational framework presented here, is the ability of the proposed algorithm to perform prediction. To be more precise, the current approximation algorithm performs smoothing within a predefined time window of inference $T = [t_0, t_f]$, given a finite set of discrete noisy observations. However, in the absence of observations the variational algorithm here seems to stick to one mode of the (true) posterior distribution. An example can be seen in the stochastic DW case (see Figure 7.5(a)), where in the absence of observations the VGPA algorithm remains in one well of the system, not being able to "see" the other well, due to the uni-modal approximation and its uncertainty remains fixed. On the contrary the HMC algorithm Figure 7.5(b), when tested on the same dataset, after the last observation is "unsure" of where it should be therefore returns to the true mean of the system ($\mathbf{x}_t$ = 0), and its variance diffuses in both wells.

The continuous time inference problem, as discussed above, when discretised results in a set of discrete time variables that need to be optimised, during the minimisation of the KL divergence. In order to reduce the number of variables that need to be optimised and control the complexity of the algorithm Chapters 5 and 6, introduce two new approximation of the VGPA algorithm. The former, in terms of basis function expansions defined on the whole time window of inference and the latter in terms of polynomials defined locally between each pair of observations. Both frameworks are derived and presented for the general multivariate case and their convergence properties, compared to the original VGPA algorithm, are examined on a range of dynamical systems as defined in Chapter 3. The general message, from both extensions, is that they are able to well approximate the results of the VGPA and also have beneficial characteristics when estimating the SDE parameters. It is shown that when estimating the drift parameters (single point estimates), the bound on the marginal likelihood (from the free energy) does not need to be very tight. Therefore, a relatively low order parametrisation of the new extensions can be used where the number of optimised parameters is lower than within the original variational framework. However, for the estimation of the system noise, which is also of great importance, a more accurate bound on the marginal likelihood must be provided.

In comparing the two new extensions, the LP method seems more appropriate for estimating the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$, mainly due to the way that the polynomials are defined. As shown in Figure 6.2, these variational parameters are discontinuous at observation times. This effect resulted in the requirement for a high number of basis functions in the RBF extension to capture this roughness. In contrast, the local polynomials do not face the same problem because they can be defined only between each pair of observations, therefore reducing the number of coefficients that need to be estimated even further.

Finally, to present a more complete study, the methods here are compared against other well known estimation techniques that cover all the main categories that deal with the Bayesian inference problem. For state estimation a range of ensemble and unscented Kalman filters and smoothers are implemented along with standard Gaussian process regression smoothers and results are given in Chapter 7.3. The application of the methods to toy example systems is very promising. On the one hand, the variational approach employed here is consistent, i.e. the solution is identical to the exact solution when the stochastic process is a Gaussian one (e.g. OU system). On the other hand, the method is able to cope with strongly non-linear systems (e.g. DW system), in contrast to most approximate state-of-the-art techniques. For the case of parameter estimation the LP approach is compared against a powerful hybrid Monte Carlo sampling algorithm, a weak constraint formulation of 4D-Var (well known in the data assimilation community) and a fast dual estimation technique based on the unscented Kalman filter. The estimation of parameters is examined asymptotically in two different regimes: **(a)** infill asymptotics, where the time window is fixed and the density of the observations increases and **(b)** increasing domain asymptotics, where the observation density remains fixed but the time window increases. Of course the word "asymptotic" here is slightly an abused term because the results are experimental and not theoretical. Therefore, the limits where $N_{obs} \rightarrow \infty$ and $T \rightarrow \infty$ are practically never reached. The results are biased, however these are in accordance with the HMC posterior sampling approach, where in most cases is assumed to provide the reference solution.

8.3 Future directions

- **Multiplicative noise** – Although the original variational Bayesian algorithm (Chapter 4), along with the two new extensions presented in Chapters (5 and 6), are defined on a Gaussian process approximation to the posterior process, in fact this Gaussian assumption restricts the applicability of the algorithm and suggests further research directions which interest the author. The algorithms presented and developed here assume additive noise ($\mathbf{\Sigma}$) in the process. However, a more realistic approach would be to treat systems where the noise varies with the states of the system (i.e. $\mathbf{\Sigma}(\mathbf{x}_t)$). This case, of multiplicative noise, cannot be treated under the current framework. The reason is that if the noise in the true process is assumed state depended then also the approximate process must be modelled by the same noise function (remember that if the two processes p_t and q_t do not have the same noise coefficient then the $\mathrm{KL}[q_t \| p_t]$ divergence is infinite). However, this would result in an approximate process that is guaranteed to be non-Gaussian (the product of two random variables, even if both of them are Gaussian, is not Gaussian). Therefore, different families of approximate processes must be sought.

- **Higher order solutions for the SDE** – To perform numerical simulations the continuous time framework of Chapter 4, has to be discretised. While the use of the standard Euler-Maruyama discretisation method here simplifies the presentation of the algorithm at the same time it imposes a small time step δt if good accuracy is to be achieved. This makes the problem more computationally demanding because a larger number of parameters has to be inferred. Nevertheless, the choice of the time discretisation is not unique. The impact of using different time discretisations, such as the Milstein scheme (Kloeden and Platen, 1999), is an open problem.

- **Application to very high dimensional systems** – So far the variational methods presented here can be applied only to toy models, or relatively low dimensional systems. The application of the algorithms to real dynamical systems with many degrees of freedom, such as the ones used for numerical weather prediction (Kalnay, 2003), is a challenge. The LP approximation algorithm (Chapter 6), reduces the number of optimising variables by 60% in most of the cases tested here (comparing to the original VGPA). Therefore, a step towards the direction of treating very high dimensional system was taken. Another benefit of the current variational framework is that one can control further the approximation (linear drift function of the approximate process $\mathbf{g}_L(\mathbf{x}_t)$), by imposing a specific structure to the linear parameter in the drift $\mathbf{A}_t$. This direction has to be further explored.

- $\mathbf{R} \rightarrow 0$ **asymptotic behaviour** – Chapter 7 compares the local polynomial extension with a range of other well known estimation techniques, on estimating the (hyper-) parameters of the tested SDEs, on two different asymptotic regimes. In the first case the time window of inference T, is kept fixed and the number of observations N_{obs} increases. The second case keeps the distance between the observations fixed and the time window increases. Of the two examples the latter (i.e. increasing domain asymptotics), proved experimentally more appropriate in estimating the model parameters. In both cases the error level on the observations was kept fixed to a relatively small value (compared to the total space or manifold that the stochastic process occupies). However, another interesting asymptotic regime is that of keeping the time window and the density of the observations fixed and letting the observation noise vary. It has been observed that in the extreme case where the noise on the observations is very high, the algorithms perform poorly when estimating the states of the system. The question that arises naturally is how the estimation of the parameters is affected for different levels of the observation noise.

- **Observation operator and noise assumptions** – Usually, to keep the notation and the presentation of an algorithm simple two important assumptions take place that should not be restrictive in the range of the applications that can be covered by the proposed algorithm. These are **(a)** a linear observation operator $h(\cdot)$ and **(b)** independent and identically distributed (i.i.d.) observations corrupted by Gaussian noise. In this thesis the same assumptions were also followed with the addition that the observations were further assumed direct measurements of the true system states (i.e. $h(\mathbf{x}_t) = \mathbf{x}_t$), when validating the algorithms on artificial datasets, although the original VGPA framework does not restrict the observation operator to be linear. Nevertheless, if the algorithms are to be tested on real observations then the methods developed here must be able to treat non-linear observation functions. The second issue that deals with non-Gaussian error statistics for the observables is more of a general statement, because in practice the Gaussian distribution models the errors in the observations adequately.

- **Computational issues** – The future directions related to computational issues are three-fold. The first has to do with the optimisation method that was chosen to solve the constraint optimisation problem of minimising the KL divergence to make the algorithm converge to its optimal posterior process. In the description of the VGPA algorithm in Chapter 4, the approximation problem was formulated in a Lagrangian framework, where the necessary ordinary differential equations that give the predictive marginal (at time 't') mean and variance (i.e. $\dot{\mathbf{m}}_t$ and $\dot{\mathbf{S}}_t$), were constraints to be satisfied. Therefore, a Lagrangian cost function was formulated (Equation 4.14) and its stationary points were to be determined. That formulation inevitably introduced more parameters (i.e. the Lagrange multipliers $\boldsymbol{\Psi}_t$ and $\boldsymbol{\lambda}_t$), which also need to be estimated. Emphasis in this thesis was given to finding approximate solutions of the original VGPA, rather than proposing a new algorithm. However, a new formulation of the original variational parameter might have beneficial characteristics by avoiding the need to compute more parameters.

The second issue is related to the previous one and has to do with the way that the marginal means and covariances are calculated. Ideally, the ODEs (Equations 4.12 and 4.13) that provide these quantities should be solved in continuous time. However, as shown in Appendix E, these ODEs do not have a closed form solution, therefore numerical methods required in obtaining the marginal mean and variance, at time ’t’. In the current framework a *forward sweep* solves the ordinary differential equations, given a fixed set of values, of the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$. The LP approximation, under this setting, provides slightly more accurate solutions by applying high order integration methods, such as the Runge-Kutta 2’nd order, without approximating the necessary mid-points that required by the integration scheme. Meanwhile, the solution of these ODEs is time-consuming and in very high dimensional systems almost impossible because the matrix giving the marginal covariance $\mathbf{S}_t$ (of size $D \times D$, where D is the systems dimension) has to be updated for each discrete time. Therefore, an alternative re-parametrisation of the original variational framework might also allow a more efficient way of computing these quantities.

The last issue relates to the LP approximation extension. As was shown, in almost all the cases where the LP algorithm compared with the original VGPA, the latter was computationally more efficient not only in actual execution time but also in the number of iterations that both algorithms need to converge. The (slightly) higher number of iterations in the optimisation routine can be explained by the fact that the LP approximation tries to constrain the available functions accepted as solutions, therefore it might take more iterations until a solution satisfies this criterion. On the other hand the original VGPA is free to optimise all the parameters unconditionally. The speed of actual execution time was not possible to capture accurately because all the simulations took place on different machines (or computer clusters), for which the author had no control over the other processes that run on these machines. Nevertheless, in practice the LP extension is slower in its present implementation due to the fact that the gradients of the cost function with respect to the optimised parameters are computed serially (i.e. for each sub-interval separately). Moreover, the algorithm could benefit from a parallel implementation of these computations because in theory these gradients are not dependent to each other.

- **Implementation** – Although it may seem of less importance, the practical application and broad acceptance of a newly proposed algorithm relies in its ease of implementation. Currently the VGPA algorithm, including the two new extensions (RBF and LP), remains quite complex. The pseudocode given in Chapter 4 sketches the outline of both state and parameter estimation procedures. A MATLAB implementation is available, however further work is necessary in order to provide more guidance and make the algorithms more generic and easily applicable.

8.4 Epilogue

The methods developed here propose a novel variational Bayesian treatment of the dynamic data assimilation problem. The initial motivation (and desire) is to make the algorithms applicable and computationally efficient to very high dimensional real-world dynamical systems. These dynamical models are currently treated deterministically, although there is increasing appreciation that a full stochastic treatment is necessary for progress to be made on probabilistic forecasting. This work is a promising step towards methods that will be able to treat such large, complex models in a fully probabilistic way. This concludes the thesis.

Bibliography

Y. Ait-Sahalia. Transition densities for interest rate and other non-linear diffusions. *Journal of Finance*, 54:1361–1395, 1999.

Y. Ait-Sahalia. Maximum Likelihood Estimation of Discretely Sampled Diffusions: A closed form approximation approach. *Econometrica*, 70(1):223–262, 2002.

F. J. Alexander, G. Eyink, and J. Restrepo. Accelerated Monte Carlo for optimal estimation of time-series. *Journal of Statistical Physics*, 119:1331–1345, 2005.

A. Apte, M. Hairer, A. Stuart, and J. Voss. Sampling the posterior: An approach to non-Gaussian data assimilation. *Physica D*, 230:50–64, 2007.

C. Archambeau, D. Cornford, M. Opper, and J. Shawe-Taylor. Gaussian Process Approximations of Stochastic Differential Equations. In *Journal of Machine Learning Research, Workshop and Conference Proceedings*, volume 1, pages 1–16, 2007.

C. Archambeau, M. Opper, Y. Shen, D. Cornford, and J. Shawe-Taylor. Variational Inference for Diffusion Processes. In C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Annual Conference on Neural Information Processing Systems (NIPS)*, volume 20, pages 17–24. The MIT Press, 2008.

N. Benoudjit, C. Archambeau, A. Lendasse, J. Lee, and M. Verleysen. Width optimization of the Gaussian kernels in Radial Basis Function Networks. In *ESANN, Proceedings*, pages 425–432, April 2002.

A. Beskos, O. Papaspiliopoulos, and G. Roberts. Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6):1077–1098, 2006a.

A. Beskos, O. Papaspiliopoulos, G. Roberts, and P. Fearnhead. Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes. *Journal of Royal Statistical Society*, 68(3):333–382, 2006b.

B. M. Bibby and M. Sorensen. Martingale estimating functions for discretely observed diffusion processes. *Bernoulli*, 1:17–39, 1995.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

M. W. Brandt and P. Santa-Clara. Simulated likelihood estimation of diffusions with an application to exchange rate dynamics in incomplete markets. *Journal of Financial Econometrics*, 63:161–210, 2002.

D. S. Broomhead and D. Lowe. Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

G. Burgers, P. J. van Leeuwen, and G. Evensen. On the analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126:1719–1724, 1998.

A. Dembo and O. Zeitouni. Parameter estimation of partially observed continuous time stochastic processes via the EM algorithm. *Stochastic Processes and their Applications*, 23:91–113, 1986.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

J. C. Derber. A variational continuous assimilation technique. *Monthly Weather Review*, 117: 2437–2446, 1989.

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

S. Duane, A.D. Kennedy, B. J. Pendeleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, September 1987.

D. Duffie and K. J. Singleton. Simulated moments estimation of Markov models of asset prices. *Econometrica*, 61:929–952, 1993.

G. B. Durham and A. R. Gallant. Numerical techniques for maximum likelihood estimation of continuous time diffusion processes. *Journal of Business and Economic Statistics*, 20:297–338, 2002.

O. Elerian, S. Chib, and N. Shephard. Likelihood inference for discretely observed non-linear diffusions. *Econometrica*, 69:959–993, 2001.

B. Eraker. MCMC analysis of diffusion models with application to finance. *Journal of Business and Economic Statistics*, 19:177–191, 2001.

G. Evensen. Advanced Data Assimilation for Strongly Non-linear Dynamics. *Monthly Weather Review*, 125:1342–1354, 1997.

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

G. Evensen and P. J. van Leeuwen. An Ensemble Kalman Smoother for Non-linear Dynamics. *Monthly Weather Review*, 128:1852–1867, 1999.

G. L. Eyink and J. M. Restrepo. Most probable histories for non-linear dynamics: tracking climate transitions. *Journal of Statistical Physics*, 101:459–472, 2000.

G. L. Eyink, J. M. Restrepo, and F. J. Alexander. A mean field approximation in data assimilation for non-linear dynamics. *Physica D*, 194:347–368, 2004.

P. Fearnhead, O. Papaspiliopoulos, and G. Roberts. Particle filters for partially observed diffusions. *Journal of the Royal Statistical Society*, 70(B):755–777, 2008.

D. Florens-Zmirou. Approximate discrete time schemes for statistics of diffusion processes. *Statistics*, 20:547–557, 1989.

A. R. Gallant and G. Tauchen. Which moments to match? *Econometric Theory*, 12:657–681, 1996.

C. W. Gardiner. *Handbook of Stochastic Methods: for physics, chemistry and the natural sciences*. Springer Series in Synergetics, 3rd edition, 2003.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall, Texts in Statistical Science, 1995.

S. Geman and D. Geman. Stochastic relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–74, 1984.

A. Golightly and D. J. Wilkinson. Bayesian Sequential Inference for Non-linear Multivariate Diffusions. *Statistics and Computing*, 16:323–338, 2006.

G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

C. Gourieroux, A. Monfot, and E. Renault. Indirect inference. *Journal of Applied Econometrics*, 8:85–118, 1993.

J. H. Gove and D. Y. Hollinger. Application of a dual unscented Kalman filter for simultaneous state and parameter estimation in problems of surface-atmosphere exchange. *Journal of Geophysical Research*, 111(D08S07):0–0, 2006. doi: 10.1029/2005JD006021.

I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series and Products*. Academic Press, 7th edition, 2007.

J. A. Hansen and C. Penland. Efficient approximate techniques for integrating stochastic differential equations. *Monthly Weather Review, Notes and Correspondence*, 134:3006–3014, 2006.

J. N. Hansen and C. Penland. On stochastic parameter estimation using data assimilation. *Physica D*, 230:88–98, 2007.

L. P. Hansen. Large sample properties of Generalised Methods of Moments estimators. *Econometrica*, 50:1029–1054, 1982.

L. P. Hansen and J. A. Scheinkman. Back to the future: generating moment implications for continuous-time Markov processes. *Econometrica*, 63:767–804, 1995.

S. Haykin. *Neural Networks a Comprehensive Foundation*. Prentice-Hall Inc., second edition, 1999.

D. J. Higham. An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations. *Society for Industrial and Applied Mathematics*, 43:525 – 546, 2001.

J. Honerkamp. *Stochastic Dynamical Systems: Concepts, Numerical Methods, Data Analysis*. Wiley - VCH, 1993.

A. S. Hurn and K. A. Lindsay. Estimating the parameters of stochastic differential equations by Monte Carlo methods. *Mathematics and Computers in Simulation*, 43:495–501, 1997.

A. S. Hurn, K. A. Lindsay, and V. L. Martin. On the efficacy of simulated maximum likelihood for estimating the parameters of stochastic differential equations. *Journal of Time Series Analysis*, 24:45–63, 2003.

K. Ide, P. Courtier, M. Ghil, and A. C. Lorenc. Unified notation for data assimilation: operational, sequential and variational. *Journal of Meteorological Society, Japan*, 75(1B):181–189, 1997.

T. Jaakkola. *Advanced Mean Field Methods: Theory and Practise*, chapter Tutorial on Variational Approximation methods. The MIT Press, 2001.

M. Jacobsen. Discretely observed diffusions: classes of estimating functions and small Delta-optimality. *Scandinavian Journal of Statistics*, 28:123–149, 2001.

J. Jeisman. *Estimation of the parameters of Stochastic Differential Equations*. PhD thesis, Queensland University of Technology, December 2005.

B. Jensen and R. Poulsen. Transition densities of diffusion processes: numerical comparison of approximation techniques. *Journal of Derivatives*, 9:18–32, 2002.

S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A New Method for Non-linear Transformation of Means and Covariances in Filters and Estimators. *IEEE Transactions on Automated Control, Technical Notes and Correspondence*, 45(3):477–482, March 2000. Accepted for publication as technical note.

R. E. Kalman. A new approach to linear filter and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82 (Series D):35–45, 1960.

R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83 (Series D):95–108, 1961.

E. Kalnay. *Atmospheric Modelling, Data Assimilation and Predictability*. Cambridge University Press, 2003.

I. Karatzas and S. E. Shreve. *Brownian motion and stochastic calculus*. Springer, New York, 1991.

M. Kessler and M. Sorensen. Estimating functions based on eigenfunctions for discretely observed diffusion process. *Bernoulli*, 5:299–314, 1999.

G. Kitagawa. Non-Gaussian state space modelling of non-stationary time series. *Journal of the American Statistical Association, Theory and Methods*, 82:1032–1041, 1987.

G. A. Kivman. Sequential parameter estimation for stochastic systems. *Non-linear Processes in Geophysics*, 10:253–259, 2003.

M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskel, and D. Lang. Fast Particle Smoothing: If I Had a Million Particles. In *Proceedings of the 23'rd International Conference on Machine Learning (ICML)*, pages 481–488, 2006.

P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, Applications of Mathematics, 3rd edition, 1999.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annal of Mathematical Statistics*, 22:79–86, 1951.

V. Kurkova and K. Hlavackova. Approximation of continuous functions by RBF and KBF networks. In *ESANN, Proceedings*, pages 167–174, 1994.

H. J. Kushner. On the differential equations satisfied by conditional probability densities of markov processes, with applications. *SIAM Control*, A 2:106–119, 1962.

H. J. Kushner. Dynamical equations for optimal non-linear filtering. *Journal of Differential Equations*, 3:179–190, 1967a.

H. J. Kushner. Approximation to optimal non-linear filters. *IEEE Trans. Auto. Control*, 12:546–556, 1967b.

A. W. Lo. Maximum Likelihood Estimation of Generalized Ito Processes with Discretely Sampled Data. *Econometric Theory*, 4:231–247, 1988.

E. N. Lorenz. Deterministic non-periodic flow. *Journal of Atmospheric Science*, 20:130–141, 1963.

E. N. Lorenz. Predictability: A problem partly solved. In T. Palmer, editor, *Predictability*, volume 1, pages 1–18. ECMWF, 1996.

E. N. Lorenz. Designing chaotic models. *Journal of Atmospheric Science*, 62:1574–1587, 2005.

E. N. Lorenz and K. A. Emanuel. Optimal Sites for Supplementary Weather Observations: Simulations with a Small Model. *Journal of the Atmospheric Science*, 55:399–414, February 1998.

D. J. MacKay. Introduction to Gaussian processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, pages 84–92. Springer-Verlag, 1998.

P. S. Maybeck. *Stochastic models, estimation and control, (Volume 1)*. Academic Press, 1979.

R. N. Miller. Topics in data assimilation: Stochastic Processes. *Physica D*, 230:17–26, 2007.

R. N. Miller, M. Ghil, and F. Gauthiez. Advanced data assimilation in strongly non-linear dynamical systems. *Journal of the Atmospheric Sciences*, 51(8):1037–1056, April 1994.

R. N. Miller, E. F. Carter, and Jr. S. T. Blue. Data assimilation into non-linear stochastic models. *Tellus A*, 51:167–194, 1999.

T. Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research Ltd., Cambridge, UK, December 2005.

I. T. Nabney. *NETLAB: Algorithms for pattern recognition*. Advances in Patern Recognition. Springer, 2002.

I. M. Navon. Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography. *Dyn. Atmos. Ocean*, 27:55–79, 1997.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, September 1993.

R. M. Neal. *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics. Springer, 1996.

C. Nicolis and G. Nicolis. Stochastic aspects of climate transitions - additive fluctuations. *Tellus*, 33:225–234, 1981.

J. N. Nielsen, H. Madsen, and P. C. Young. Parameter estimation in stochastic differential equations: an overview. *Annual Reviews in Control*, 24:83–94, 2000.

B. Øksendal. *Stochastic Differential Equations. An introduction with applications*. Springer-Verlag, 5th edition, 2000.

D. Orrell. *Modelling non-linear dynamical systems: chaos, error and uncertainty*. PhD thesis, Oxford University, 2001.

D. Orrell. Model error and predictability over different timescales in the Lorenz 96 systems. *Journal of Atmospheric Science*, 60:2219–2228, 2003.

D. Orrell. *The future of everything - The science of prediction*. Thunder's Mouth Press, 2007.

D. Orrell, L. Smith, J. Barkmeijer, and T. N. Palmer. Model error in weather forecasting. *Nonlinear Processes in Geophysics*, 8:357–371, 2001.

M. Osborne. Gaussian Processes for Prediction. Technical Report PARG-07-01, Department of Engineering Science, University of Oxford, October 2007.

O. Papaspiliopoulos and G. Roberts. Retrospective MCMC methods for Dirichlet process hierarchical models. *Biometrika*, 95:169–186, 2008.

A. Papoulis. *Probability, Random Variables and Stochastic Processes*. Series in Electrical Engineering. McGraw-Hill, Inc. New York, 3rd edition, 1984.

E. Pardoux. Equations du filtrage non lineaire de la prediction et du lissage. *Stochastics*, 6: 193–231, 1982.

A. R. Pedersen. A new approach to maximum likelihood estimation for stochastic differential equations based on discrete observations. *Scandinavian Journal of Statistics*, 22:55–71, 1995.

C. Penland. A stochastic approach to non-linear dynamics. *American Meteorological Society (AMS)*, 84:921–925, 2003.

K. B. Petersen and M. S. Petersen. The matrix cookbook. Matrix Identites, September 2007.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT press, Cambridge, 2006.

H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA*, 3:1445–1450, 1965.

G. Roberts and O. Stramer. On inference for partially observed non-linear diffusion models using the Metropolis-Hastings algorithm. *Biometrika*, 88(3):603–621, 2001.

S. Sarkka. Unscented Rauch-Tung-Striebel smoother. *IEEE Trans. Auto. Control*, 53(3):845–849, April 2008.

Y. Sasaki. Some basic formalisms in numerical variational analysis. *Monthly Weather Review*, 98: 875–883, 1970.

G. Shafer. What is probability? In David C. Hoaglin and David S. Moore, editors, *Perspectives on Contemporary Statistics*, number 21, pages 93–105. Mathematical Association of America, 1992.

I. Shoji and T. Ozaki. Comparative study of estimation methods for continuous time stochastic processes. *Journal of Time Series Analysis*, 18:485–506, 1997.

H. Sorensen. Parametric inference for diffusion processes observed at discrete points in time: a survey. *International Statistics Review*, 72(3):337–354, 2004.

M. Sorensen. Prediction based estimating functions. *Econometrics*, 3:123–147, 2000.

R. L. Stratonovich. Conditional Markov Processes. *Theory of Probability and its Application*, 5: 156–178, 1960.

A. M. Stuart, J. Voss, and P. Wiberg. Conditional path sampling of SDEs and the Langevin MCMC method. *Communications in Mathematical Science*, 2:685–697, 2004.

A. Sutera. On stochastic perturbations and long term climate behaviour. *Quarterly Journal of the Royal Meteorological Society*, 107:137–152, 1980.

M. E. Tipping. Bayesian Inference: An Introduction to Principles and Practice in Machine Learning. In O. Bousquet, U. von Luxburg, and G. Ratsch, editors, *Advanced Lectures on Machine Learning*, pages 41–62. Springer, 2006.

Y. Tremolet. Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 132(621):2483–2504, 2006.

J. R. Tsai, P. C. Chung, and C. I. Chang. A sigmoidal radial basis function neural network for function approximation. In *IEEE International Conference on Neural Networks*, volume 1, pages 496–501, 1996.

G. E. Uhlenbeck and L. S. Ornstein. On the theory of Brownian motion. *Physical Review*, 36: 823–841, 1930.

J. K. Uhlmann. *Dynamic map buildings and localisations: New theoretical foundations*. PhD thesis, Department of Engineering, University of Oxford, 1995.

R. van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, OGI School of Science & Engineering, Oregon Health & Science University, 2004.

R. van der Merwe and E. A. Wan. The square root unscented Kalman filter for state and parameter estimation. In *IEEE International conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3461–3464, 2001.

P. J. van Leeuwen. An ensemble smoother with error statistics. *Monthly Weather Review*, 129: 709–728, 2001.

M. Verleysen and K. Hlavackova. An optimised RBF network for approximation of functions. In *ESANN, Proceedings*, pages 175–180, April 1994.

M. D. Vrettas, Y. Shen, and D. Cornford. Derivations of Variational Gaussian Process Approximation Framework. Technical Report NCRG/2008/002, Neural Computing Research Group (NCRG), Aston University, Birmingham, B4 7ET, UK, March 2008.

M. D. Vrettas, D. Cornford, and Y. Shen. A variational basis function approximation for diffusion processes. In *ESANN, Proceedings*, pages 497–502, April 2009.

M. D. Vrettas, D. Cornford, and M. Opper. Estimating parameters in stochastic systems: A variational bayesian approach. submitted, July 2010a.

M. D. Vrettas, D. Cornford, M. Opper, and Y. Shen. A new variational radial basis function approximation for inference in multivariate diffusions. *Neurocomputing*, 73:1186–1198, 2010b.

E. A. Wan and R. van der Merwe. The unscented Kalman filter for non-linear estimation. In *IEEE Symposium*, 2000.

E. A. Wan, R. van der Merwe, and A. T. Nelson. Dual estimation and the unscented transformation. In *Neural Information Processing Systems (NIPS)*, 2000.

C. K. Wikle and L. M. Berliner. A bayesian tutorial for data assimilation. *Physica D*, 230:1–16, 2007.

D. Zupanski. A general weak constraint applicable to operational 4D-VAR data assimilation systems. *Monthly Weather Review*, 125:2274–2292, 1996.

A Derivations of the VGPA framework

This Appendix derives the necessary equations for the formulation of the variational Gaussian process approximation to the posterior distribution over paths, for partially observed diffusion processes, as introduced in Chapter 4. The expressions here are presented in a generic format leaving the system specific derivations, for the systems studied in this thesis, to be given later (see Appendix D).

A.1 Basic setting

In order to fix ideas and make the derivations more clear the basic setting is introduced first on which the variational approximation framework is based on. Consider a finite set of d-dimensional noisy observations $\{\mathbf{y}_k\}_{k=1}^{K}$, that are generated by a D-dimensional latent process $\mathbf{x}_t$.

It is assumed that the time evolution of this D-dimensional stochastic process $\mathbf{x}_t$ is described by an Itô stochastic differential equation (SDE):

$$d\mathbf{x}_t = \mathbf{f}_{\boldsymbol{\theta}}(t,\mathbf{x}_t)\, dt + \boldsymbol{\Sigma}^{1/2} d\mathbf{w}_t, \qquad d\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, dt\mathbf{I}) \tag{A.1}$$

where $\mathbf{x}_t \in \Re^D$ is the (latent) state vector, $\mathbf{f}_{\boldsymbol{\theta}}(t,\mathbf{x}_t) \in \Re^D$ is (usually) a non-linear function, $\boldsymbol{\Sigma} = \text{diag}\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_D^2\}$ is the system noise covariance matrix and $\{\mathbf{w}_t\}_{t \in T}$ is the standard D dimensional *Wiener process*. A discrete version of (A.1) can be provided by the Euler-Maruyama representation of a SDE. Hence:

$$\delta \mathbf{x}_k \equiv \mathbf{x}_{k+1} - \mathbf{x}_k = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_k)\delta t + \sqrt{\delta t \boldsymbol{\Sigma}} \boldsymbol{\epsilon}_k \ , \tag{A.2}$$

where δt is a positive finite real number representing the time increment and $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0, I)$. As $\delta t \to 0$ this becomes equivalent to the continuous time version (A.1).

In a Bayesian framework, the posterior measure in the presence of independent and identically distributed (i.i.d.) observations is given by:

$$\frac{d p_{post}}{d p_{sde}} = \frac{1}{Z} \times \prod_{k=1}^{K} p(\mathbf{y}_k | \mathbf{x}_{t_k}) \ , \tag{A.3}$$

using the Radon-Nikodym notation, where K denotes the number of noisy observations and Z is the normalisation constant, or marginal likelihood, or evidence (i.e. $Z = p(\mathbf{y}_{1:K})$). As usual, the multivariate Gaussian likelihood is given by:

$$p(\mathbf{y}_k | \mathbf{x}_{t_k}) = \mathcal{N}(\mathbf{y}_k | h(\mathbf{x}_{t_k}), \mathbf{R}) \ , \tag{A.4}$$

where $h(\cdot) : \Re^D \mapsto \Re^d$ is a general non-linear transformation between the latent state vector $\mathbf{x}_{t_k}$ and the observation $\mathbf{y}_k$ and $\mathbf{R} \in \Re^{d \times d}$ is the noise covariance matrix related to the observables.

A more thorough study and presentation of stochastic differential equations, as well as different discretisation schemes, can be found in many text-books. Here are cited three of the most commonly used (Kloeden and Platen, 1999; Øksendal, 2000; Gardiner, 2003).

A.2 Approximate Inference

The variational free energy, is defined as follows:

$$\mathcal{F}(q(\mathbf{x}), \boldsymbol{\theta}, \boldsymbol{\Sigma}) = - \left\langle \ln \frac{p(\mathbf{y}, \mathbf{x} | \boldsymbol{\theta}, \boldsymbol{\Sigma})}{q(\mathbf{x} | \boldsymbol{\Sigma})} \right\rangle_{q_t} \tag{A.5}$$

where $p(\cdot)$ is the true posterior process of the system, $q(\cdot)$ is the one that is used as an approximation and time indices have been dropped for notational simplicity. Also $\mathbf{x} = \{\mathbf{x}_t, t_0 \leq t \leq t_f\}$ represents here the path of a continuous time D-dimensional stochastic process and $\langle \cdot \rangle_q$ indicates the expectation with respect to process $q(\cdot)$.

Alternatively, the variational free energy can be seen as the KL divergence between the approximate process $q(\mathbf{x})$ and the joint distribution of the latent states and the observations of the true system $p(\mathbf{y},\mathbf{x})$, as follows:

$$\mathcal{F}(q(\mathbf{x}),\boldsymbol{\theta},\boldsymbol{\Sigma}) = -\left\langle \ln \frac{p(\mathbf{y},\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\Sigma})}{q(\mathbf{x}|\boldsymbol{\Sigma})} \right\rangle_{q_t} \tag{A.6}$$

$$= -\int q(\mathbf{x}) \ln \frac{p(\mathbf{y},\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \tag{A.7}$$

$$= \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{y},\mathbf{x})} d\mathbf{x} \tag{A.8}$$

$$= \mathrm{KL}[q(\mathbf{x})\|p(\mathbf{y},\mathbf{x})] \ , \tag{A.9}$$

where the conditioning on the (hyper-) parameters $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ has been omitted for notational simplicity.

The free energy provides an upper bound to the negative marginal log-likelihood. Starting with the *product rule* of probabilities, this is:

$$p(\mathbf{x},\mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y}) \Rightarrow \tag{A.10}$$

$$p(\mathbf{y}) = \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x}|\mathbf{y})} \ , \tag{A.11}$$

after applying the natural logarithm on both sides of Equation (A.11), it yields:

$$\ln p(\mathbf{y}) = \ln \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x}|\mathbf{y})} \tag{A.12}$$

$$= \ln p(\mathbf{x},\mathbf{y}) - \ln p(\mathbf{x}|\mathbf{y}) \ , \tag{A.13}$$

then adding and subtracting the same quantity, by introducing a new distribution $q(\mathbf{x})$, results:

$$-\ln p(\mathbf{y}) = \ln p(\mathbf{x}|\mathbf{y}) - \ln p(\mathbf{x},\mathbf{y}) \tag{A.14}$$

$$= \ln p(\mathbf{x}|\mathbf{y}) - \ln q(\mathbf{x}) - \ln p(\mathbf{x},\mathbf{y}) + \ln q(\mathbf{x}) \tag{A.15}$$

$$= \ln \frac{p(\mathbf{x}|\mathbf{y})}{q(\mathbf{x})} - \ln \frac{p(\mathbf{x},\mathbf{y})}{q(\mathbf{x})} \ . \tag{A.16}$$

Multiplying both sides by $q(\mathbf{x})$ we have:

$$-q(\mathbf{x}) \ln p(\mathbf{y}) = q(\mathbf{x}) \ln \frac{p(\mathbf{x}|\mathbf{y})}{q(\mathbf{x})} - q(\mathbf{x}) \ln \frac{p(\mathbf{x},\mathbf{y})}{q(\mathbf{x})} \ , \tag{A.17}$$

and then integrating over $\mathbf{x}$ yields:

$$-\int q(\mathbf{x}) \ln p(\mathbf{y}) d\mathbf{x} = \int q(\mathbf{x}) \ln \frac{p(\mathbf{x}|\mathbf{y})}{q(\mathbf{x})} d\mathbf{x} - \int q(\mathbf{x}) \ln \frac{p(\mathbf{x},\mathbf{y})}{q(\mathbf{x})} d\mathbf{x} \Rightarrow \tag{A.18}$$

$$-\ln p(\mathbf{y}) = \mathrm{KL}[q(\mathbf{x})\|p(\mathbf{x},\mathbf{y})] - \mathrm{KL}[q(\mathbf{x})\|p(\mathbf{x}|\mathbf{y})] \ . \tag{A.19}$$

Since $p(\mathbf{y})$ has no dependency on $\mathbf{x}$ which leads to:

$$-\ln p(\mathbf{y}|\boldsymbol{\theta},\boldsymbol{\Sigma}) = \mathcal{F}(q(\mathbf{x}),\boldsymbol{\theta},\boldsymbol{\Sigma}) - \mathrm{KL}[q(\mathbf{x}|\boldsymbol{\Sigma})\|p(\mathbf{x}|\mathbf{y},\boldsymbol{\theta},\boldsymbol{\Sigma})] \tag{A.20}$$

$$\leq \mathcal{F}(q(\mathbf{x}),\boldsymbol{\theta},\boldsymbol{\Sigma}) \ , \tag{A.21}$$

because by definition KL ≥ 0. Note that the conditioning on the (hyper-) parameters $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ is added here for later clarity.

A.2.1 Optimal approximate posterior process

An approximate time-varying Gaussian process is defined, with the same diffusion coefficient ($\boldsymbol{\Sigma}$) as the process which is approximated. This process is governed by the following linear SDE:

$$d\mathbf{x}_t = \mathbf{g}_L(\mathbf{x}_t)\ dt + \boldsymbol{\Sigma}^{1/2} d\mathbf{w}_t, \qquad d\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, dt\mathbf{I}) \tag{A.22}$$

where the linear drift is defined as: $\mathbf{g}_L(\mathbf{x}_t) = -\mathbf{A}_t\mathbf{x}_t + \mathbf{b}_t$, with $\mathbf{A}_t \in \Re^{D\times D}$ and $\mathbf{b}_t \in \Re^D$. Note that both parameters $\mathbf{A}_t$ and $\mathbf{b}_t$, are time dependent functions to account for the non-stationarity induced by the observations. It is anticipated, that the Gaussian marginal at time 't' is defined as follows:

$$q(\mathbf{x}_t|\boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{x}_t|\mathbf{m}_t, \mathbf{S}_t)\ , \tag{A.23}$$

(henceforth q_t), where $\mathbf{m}_t \in \Re^D$ and $\mathbf{S}_t \in \Re^{D\times D}$, are respectively the marginal mean and marginal covariance at time 't'. The derivation of the free energy leads to the following result:

$$\mathcal{F}(q(\mathbf{x}), \boldsymbol{\theta}, \boldsymbol{\Sigma}) = \mathrm{KL}[q_0\|p_0] + \int_{t_0}^{t_f} E_{sde}(t)dt + \int_{t_0}^{t_f} E_{obs}(t) \sum_n \delta(t - t_n)dt \tag{A.24}$$

where $\delta(\cdot)$ is Dirac's delta function, $\mathrm{KL}[q_0\|p_0]$ is a shorthand notation for the KL at the initial state (i.e. $\mathrm{KL}[q(\mathbf{x}_0)\|p(\mathbf{x}_0)]$) and the energy functions are defined in equations (A.51) and (A.57) below:

Proof: From Equation (A.9) we have:

$$\begin{aligned}
\mathcal{F}(q(\mathbf{x}), \boldsymbol{\theta}, \boldsymbol{\Sigma}) &= \mathrm{KL}[q(\mathbf{x})\|p(\mathbf{y},\mathbf{x})] & \text{(A.25)}\\
&= \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{y},\mathbf{x})} d\mathbf{x} & \text{(A.26)}\\
&= \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})} d\mathbf{x} & \text{(A.27)}\\
&= \underbrace{\int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x}}_{\textbf{(I1)}} - \underbrace{\int q(\mathbf{x}) \ln p(\mathbf{y}|\mathbf{x}) d\mathbf{x}}_{\textbf{(I2)}} & \text{(A.28)}
\end{aligned}$$

I1: This integral is simply the KL divergence between the approximate prior process $q(\mathbf{x})$ and the true prior process $p(\mathbf{x})$ defined in (A.1). Alternatively, this integral can be written as:

$$\mathrm{KL}[q(\mathbf{x})\|p(\mathbf{x})] = \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x}\ . \tag{A.29}$$

However, to make the derivation more clear the above notation will change to the one that follows to emphasise the discretisation of the sample paths on the time interval (note a continuous time derivation is also possible).

$$\mathrm{KL}[q(\mathbf{x}_{0:N})\|p(\mathbf{x}_{0:N})] = \int\ldots\int q(\mathbf{x}_{0:N})\ln\frac{q(\mathbf{x}_{0:N})}{p(\mathbf{x}_{0:N})}d\mathbf{x}_{0:N} \tag{A.30}$$

$$= \int\ldots\int q(\mathbf{x}_{0:N})\ln\frac{q(\mathbf{x}_0)\prod_{j=0}^{N-1}q(\mathbf{x}_{j+1}|\mathbf{x}_j)}{p(\mathbf{x}_0)\prod_{j=0}^{N-1}p(\mathbf{x}_{j+1}|\mathbf{x}_j)}d\mathbf{x}_{0:N} \tag{A.31}$$

$$= \int\ldots\int q(\mathbf{x}_{0:N})\ln\frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)}d\mathbf{x}_{0:N} + \int\ldots\int q(\mathbf{x}_{0:N})\ln\prod_{j=0}^{N-1}\left[\frac{q(\mathbf{x}_{j+1}|\mathbf{x}_j)}{p(\mathbf{x}_{j+1}|\mathbf{x}_j)}\right]d\mathbf{x}_{0:N} \tag{A.32}$$

$$= \underbrace{\int q(\mathbf{x}_0)\ln\frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)}d\mathbf{x}_0}_{\mathrm{KL}[q_0\|p_0]} + \int\ldots\int q(\mathbf{x}_{0:N})\ln\prod_{j=0}^{N-1}\left[\frac{q(\mathbf{x}_{j+1}|\mathbf{x}_j)}{p(\mathbf{x}_{j+1}|\mathbf{x}_j)}\right]d\mathbf{x}_{0:N} \tag{A.33}$$

$$= \mathrm{KL}[q_0\|p_0] + \int\ldots\int q(\mathbf{x}_{0:N})\ln\prod_{j=0}^{N-1}\left[\frac{q(\mathbf{x}_{j+1}|\mathbf{x}_j)}{p(\mathbf{x}_{j+1}|\mathbf{x}_j)}\right]d\mathbf{x}_{0:N} \tag{A.34}$$

$$= \mathrm{KL}[q_0\|p_0] + \int\ldots\int q(\mathbf{x}_0)\prod_{i=0}^{N-1}q(\mathbf{x}_{i+1}|\mathbf{x}_i)\sum_{j=0}^{N-1}\ln\frac{q(\mathbf{x}_{j+1}|\mathbf{x}_j)}{p(\mathbf{x}_{j+1}|\mathbf{x}_j)}d\mathbf{x}_{0:N} \tag{A.35}$$

This result is due to the fact that both processes are Markovian. Hence, their marginal distributions can be factorised as a product of conditional distributions (i.e. the transition probabilities):

$$q(\mathbf{x}_{0:N}) = q(\mathbf{x}_0)\prod_{i=0}^{N-1}q(\mathbf{x}_{i+1}|\mathbf{x}_i)\ . \tag{A.36}$$

The same is true for $p(\mathbf{x}_{0:N})$. Continuing the derivation one obtains:

$$\mathrm{KL}[q\|p] = \mathrm{KL}[q_0\|p_0] + \sum_{j=0}^{N-1}\int\ldots\int\prod_{i=0}^{N-1}q(\mathbf{x}_{i+1}|\mathbf{x}_i)\ln\frac{q(\mathbf{x}_{j+1}|\mathbf{x}_j)}{p(\mathbf{x}_{j+1}|\mathbf{x}_j)}d\mathbf{x}_{1:N} \tag{A.37}$$

$$= \mathrm{KL}[q_0\|p_0] +$$

$$\sum_{j=0}^{N-1}\int\ldots\int\prod_{k=1}^{j}q(\mathbf{x}_k|\mathbf{x}_{k-1})q(\mathbf{x}_{j+1}|\mathbf{x}_j)\ln\frac{q(\mathbf{x}_{j+1}|\mathbf{x}_j)}{p(\mathbf{x}_{j+1}|\mathbf{x}_j)}\prod_{m=j+1}^{N-1}q(\mathbf{x}_{m+1}|\mathbf{x}_m)d\mathbf{x}_{1:N}\ . \tag{A.38}$$

At this point the following substitution takes place:

$$\int\ldots\int\prod_{k=1}^{j}q(\mathbf{x}_k|\mathbf{x}_{k-1})d\mathbf{x}_{1:j-1} = q(\mathbf{x}_j)\ , \tag{A.39}$$

since this is equal to the marginal distribution $q(\mathbf{x}_j)$. Therefore:

$$\mathrm{KL}[q\|p] = \mathrm{KL}[q_0\|p_0] + \sum_{j=1}^{N-1}\int\ldots\int q(\mathbf{x}_j)q(\mathbf{x}_{j+1}|\mathbf{x}_j)\ln\frac{q(\mathbf{x}_{j+1}|\mathbf{x}_j)}{p(\mathbf{x}_{j+1}|\mathbf{x}_j)}\prod_{m=j+1}^{N-1}q(\mathbf{x}_{m+1}|\mathbf{x}_m)d\mathbf{x}_{j:N} \tag{A.40}$$

A careful look on the right hand side, of the previous expression, after the $\left[\ln\frac{q(\mathbf{x}_{j+1}|\mathbf{x}_j)}{p(\mathbf{x}_{j+1}|\mathbf{x}_j)}\right]$, reveals a set of integrals that evaluate to one. That is:

$$\underbrace{\int q(\mathbf{x}_{j+2}|\mathbf{x}_{j+1})d\mathbf{x}_{j+2}}_{=1}\ \underbrace{\int q(\mathbf{x}_{j+3}|\mathbf{x}_{j+2})d\mathbf{x}_{j+3}}_{=1}\cdots\underbrace{\int q(\mathbf{x}_N|\mathbf{x}_{N-1})d\mathbf{x}_N}_{=1}\ .$$

So we are left with the following expression:

$$\mathrm{KL}[q\|p] = \mathrm{KL}[q_0\|p_0] + \sum_{j=1}^{N-1} \int q(\mathbf{x}_j) \underbrace{\int q(\mathbf{x}_{j+1}|\mathbf{x}_j) \ln \frac{q(\mathbf{x}_{j+1}|\mathbf{x}_j)}{p(\mathbf{x}_{j+1}|\mathbf{x}_j)} d\mathbf{x}_{j+1}}_{KL[q(\mathbf{x}_{j+1}|\mathbf{x}_j)||p_{sde}(\mathbf{x}_{j+1}|\mathbf{x}_j)]} d\mathbf{x}_j \tag{A.41}$$

$$= \mathrm{KL}[q_0\|p_0] + \sum_{j=1}^{N-1} \int q(\mathbf{x}_j) \mathrm{KL}[q(\mathbf{x}_{j+1}|\mathbf{x}_j)\|p_{sde}(\mathbf{x}_{j+1}|\mathbf{x}_j)] d\mathbf{x}_j \tag{A.42}$$

$$= \mathrm{KL}[q_0\|p_0] + \sum_{j=1}^{N-1} \Big\langle \mathrm{KL}[q(\mathbf{x}_{j+1}|\mathbf{x}_j)\|p_{sde}(\mathbf{x}_{j+1}|\mathbf{x}_j)] \Big\rangle_{q(\mathbf{x}_j)} . \tag{A.43}$$

The above KL divergence, provided that both processes p and q are Gaussians, is given by the following formula (Rasmussen and Williams, 2006, Mathematical Appendix):

$$\mathrm{KL}[q(\mathbf{x}_{j+1}|\mathbf{x}_j)\|p(\mathbf{x}_{j+1}|\mathbf{x}_j)] = \frac{1}{2} \ln |\boldsymbol{\Sigma}_p \boldsymbol{\Sigma}_q^{-1}| + \frac{1}{2} \mathrm{tr}\Big[\boldsymbol{\Sigma}_p^{-1}\Big((\mathbf{m}_p - \mathbf{m}_q)(\mathbf{m}_p - \mathbf{m}_q)^\top + \boldsymbol{\Sigma}_p - \boldsymbol{\Sigma}_q\Big)\Big]. \tag{A.44}$$

From equations (A.1) and (A.22) one can see a critical assumption; that both processes have the same system noise covariance $\boldsymbol{\Sigma}$. Hence the following substitution is made to the previous expression: $\boldsymbol{\Sigma}_p = \boldsymbol{\Sigma}_q = \boldsymbol{\Sigma}$.

$$\mathrm{KL}[q(\mathbf{x}_{j+1}|\mathbf{x}_j)\|p(\mathbf{x}_{j+1}|\mathbf{x}_j)] = \frac{1}{2} \ln |\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}| + \frac{1}{2} \mathrm{tr}\Big[\boldsymbol{\Sigma}^{-1}\Big((\mathbf{m}_p - \mathbf{m}_q)(\mathbf{m}_p - \mathbf{m}_q)^\top + \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\Big)\Big] \tag{A.45}$$

$$= \underbrace{\frac{1}{2} \ln |\mathbf{I}|}_{=0} + \frac{1}{2} \mathrm{tr}\Big[\boldsymbol{\Sigma}^{-1}\Big((\mathbf{m}_p - \mathbf{m}_q)(\mathbf{m}_p - \mathbf{m}_q)^\top\Big)\Big] \tag{A.46}$$

$$= \frac{1}{2} \mathrm{tr}\Big[\boldsymbol{\Sigma}^{-1}\Big((\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_{j+1}) - \mathbf{g}_L(\mathbf{x}_{j+1}))(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_{j+1}) - \mathbf{g}_L(\mathbf{x}_{j+1})^\top)\Big)\Big] \delta t \tag{A.47}$$

$$= \frac{1}{2} \Big[(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_{j+1}) - \mathbf{g}_L(\mathbf{x}_{j+1}))^\top \boldsymbol{\Sigma}^{-1} (\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_{j+1}) - \mathbf{g}_L(\mathbf{x}_{j+1}))\Big] \delta t \tag{A.48}$$

Therefore for the whole discretised path p_{sde} it holds:

$$\mathrm{KL}[q\|p] = \mathrm{KL}[q_0\|p_0] + \frac{1}{2} \sum_{k=1}^{N-1} \Big\langle (\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_k) - \mathbf{g}_L(\mathbf{x}_k))^\top \boldsymbol{\Sigma}^{-1} (\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_k) - \mathbf{g}_L(\mathbf{x}_k)) \Big\rangle_{q_k} \delta t . \tag{A.49}$$

And in the limit of $\delta t \rightarrow 0$:

$$\mathrm{KL}[q\|p_{sde}] = \mathrm{KL}[q_0\|p_0] + \frac{1}{2} \int_{t_0}^{t_f} \Big\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \boldsymbol{\Sigma}^{-1} (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t)) \Big\rangle_{q_t} dt . \tag{A.50}$$

The energy from the SDE is thus given by the following expression:

$$E_{sde}(t) = \frac{1}{2} \Big\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \boldsymbol{\Sigma}^{-1} (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t)) \Big\rangle_{q_t} \tag{A.51}$$

Then the computation of the log-likelihood follows, noting that this is now formulated in continuous time. This is done to simplify the computation of the integral I2, as shown below.

$$\ln p(\mathbf{y}_t|\mathbf{x}_t) = \ln\left(\mathcal{N}(\mathbf{y}_t|h(\mathbf{x}_t),\mathbf{R})\right) \tag{A.52}$$

$$= \ln\left((2\pi)^{-\frac{d}{2}}|\mathbf{R}|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\mathbf{y}_t - h(\mathbf{x}_t))^\top\mathbf{R}^{-1}(\mathbf{y}_t - h(\mathbf{x}_t))\right\}\right) \tag{A.53}$$

$$= -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{R}| - \frac{1}{2}(\mathbf{y}_t - h(\mathbf{x}_t))^\top\mathbf{R}^{-1}(\mathbf{y}_t - h(\mathbf{x}_t))\ . \tag{A.54}$$

I2: Finally, this integral becomes:

$$\int q(\mathbf{x}_t)\ln p(\mathbf{y}_t|\mathbf{x}_t)d\mathbf{x}_t = -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{R}| - \frac{1}{2}\int q(\mathbf{x}_t)\left((\mathbf{y}_t - h(\mathbf{x}_t))^\top\mathbf{R}^{-1}(\mathbf{y}_t - h(\mathbf{x}_t))\right)d\mathbf{x}_t \tag{A.55}$$

$$= -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{R}| - \frac{1}{2}\left\langle(\mathbf{y}_t - h(\mathbf{x}_t))^\top\mathbf{R}^{-1}(\mathbf{y}_t - h(\mathbf{x}_t))\right\rangle_{q_t}\ . \tag{A.56}$$

Thus the energy from the observations, at time ’t’, is the given by:

$$E_{obs}(t) = \frac{1}{2}\left\langle(\mathbf{y}_t - h(\mathbf{x}_t))^\top\mathbf{R}^{-1}(\mathbf{y}_t - h(\mathbf{x}_t))\right\rangle_{q_t} + \frac{d}{2}\ln(2\pi) + \frac{1}{2}\ln|\mathbf{R}|\ , \tag{A.57}$$

where $\mathbf{y} = \{\mathbf{y}_t\ , t_0 \le t \le t_f\} \in \Re^d$ is a continuous-time observable process. The discrete time nature of the actual observations adds the delta function in equation (A.24).

A.2.2 Smoothing algorithm

The time evolution of the Gaussian measure Eq. (A.23) can be described by a set of ordinary differential equations. These follow from Equation (A.22), and given in Eq. (A.58) and (A.59).

$$\dot{\mathbf{m}}_t = -\mathbf{A}_t\mathbf{m}_t + \mathbf{b}_t \tag{A.58}$$

$$\dot{\mathbf{S}}_t = -\mathbf{A}_t\mathbf{S}_t - \mathbf{S}_t\mathbf{A}_t^\top + \mathbf{\Sigma} \tag{A.59}$$

where $\dot{\mathbf{m}}_t$ and $\dot{\mathbf{S}}_t$ are shorthand notations for $\frac{d\mathbf{m}_t}{dt}$ and $\frac{d\mathbf{S}_t}{dt}$, respectively.

ODEs of the means (with respect to time t) :

$$d\mathbf{m}_t = \langle\mathbf{x}_t + d\mathbf{x}_t\rangle - \langle\mathbf{x}_t\rangle \tag{A.60}$$

$$= \langle\mathbf{x}_t\rangle + \langle d\mathbf{x}_t\rangle - \langle\mathbf{x}_t\rangle \tag{A.61}$$

$$= \langle d\mathbf{x}_t\rangle \tag{A.62}$$

$$= \left\langle\mathbf{g}_L(\mathbf{x}_t)dt + \mathbf{\Sigma}^{1/2}d\mathbf{w}_t\right\rangle \tag{A.63}$$

$$= \langle\mathbf{g}_L(\mathbf{x}_t)\rangle dt + \mathbf{\Sigma}^{1/2}\langle d\mathbf{w}_t\rangle \tag{A.64}$$

(continues to A.65)

$$= \langle -\mathbf{A}_t\mathbf{x}_t + \mathbf{b}_t\rangle\, dt \tag{A.65}$$

$$= -\mathbf{A}_t \langle \mathbf{x}_t\rangle\, dt + \mathbf{b}_t dt \tag{A.66}$$

$$= -\mathbf{A}_t\mathbf{m}_t dt + \mathbf{b}_t dt \tag{A.67}$$

where $d\mathbf{x}_t$ has been replaced with $(\mathbf{g}_L(\mathbf{x}_t)dt + \mathbf{\Sigma}^{1/2}d\mathbf{w}_t)$, from Eq. (A.22), and $\langle d\mathbf{w}_t\rangle = 0$.

ODEs of the variances (with respect to time t) :

$$d\mathbf{S}_t = \left\langle (\mathbf{x}_t - \mathbf{m}_t + d\mathbf{x}_t - d\mathbf{m}_t)(\mathbf{x}_t - \mathbf{m}_t + d\mathbf{x}_t - d\mathbf{m}_t)^\top \right\rangle - \left\langle (\mathbf{x}_t - \mathbf{m}_t)(\mathbf{x}_t - \mathbf{m}_t)^\top \right\rangle \tag{A.68}$$

$$= \left\langle (\mathbf{x}_t - \mathbf{m}_t + d\mathbf{x}_t - d\mathbf{m}_t)(\mathbf{x}_t^\top - \mathbf{m}_t^\top + d\mathbf{x}_t^\top - d\mathbf{m}_t^\top) \right\rangle - \mathbf{S}_t \tag{A.69}$$

$$\begin{aligned} &= \left\langle \mathbf{x}_t\mathbf{x}_t^\top - \mathbf{x}_t\mathbf{m}_t^\top + \mathbf{x}_t d\mathbf{x}_t^\top - \mathbf{x}_t d\mathbf{m}_t^\top \right\rangle + \\ &\left\langle -\mathbf{m}_t\mathbf{x}_t^\top + \mathbf{m}_t\mathbf{m}_t^\top - \mathbf{m}_t d\mathbf{x}_t^\top + \mathbf{m}_t d\mathbf{m}_t^\top \right\rangle + \\ &\left\langle d\mathbf{x}_t\mathbf{x}_t^\top - d\mathbf{x}_t\mathbf{m}_t^\top + d\mathbf{x}_t d\mathbf{x}_t^\top - d\mathbf{x}_t d\mathbf{m}_t^\top \right\rangle + \\ &\left\langle -d\mathbf{m}_t\mathbf{x}_t^\top + d\mathbf{m}_t\mathbf{m}_t^\top - d\mathbf{m}_t d\mathbf{x}_t^\top + d\mathbf{m}_t d\mathbf{m}_t^\top \right\rangle - \mathbf{S}_t \end{aligned} \tag{A.70}$$

$$\begin{aligned} &= \left\langle \mathbf{x}_t\mathbf{x}_t^\top \right\rangle - \left\langle \mathbf{x}_t\mathbf{m}_t^\top \right\rangle + \left\langle \mathbf{x}_t d\mathbf{x}_t^\top \right\rangle - \left\langle \mathbf{x}_t d\mathbf{m}_t^\top \right\rangle + \\ &\left\langle -\mathbf{m}_t\mathbf{x}_t^\top \right\rangle + \left\langle \mathbf{m}_t\mathbf{m}_t^\top \right\rangle - \left\langle \mathbf{m}_t d\mathbf{x}_t^\top \right\rangle + \left\langle \mathbf{m}_t d\mathbf{m}_t^\top \right\rangle + \\ &\left\langle d\mathbf{x}_t\mathbf{x}_t^\top \right\rangle - \left\langle d\mathbf{x}_t\mathbf{m}_t^\top \right\rangle + \left\langle d\mathbf{x}_t d\mathbf{x}_t^\top \right\rangle - \left\langle d\mathbf{x}_t d\mathbf{m}_t^\top \right\rangle + \\ &\left\langle -d\mathbf{m}_t\mathbf{x}_t^\top \right\rangle + \left\langle d\mathbf{m}_t\mathbf{m}_t^\top \right\rangle - \left\langle d\mathbf{m}_t d\mathbf{x}_t^\top \right\rangle + \left\langle d\mathbf{m}_t d\mathbf{m}_t^\top \right\rangle - \mathbf{S}_t \end{aligned} \tag{A.71}$$

$$\begin{aligned} &= \mathbf{m}_t\mathbf{m}_t^\top + \mathbf{S}_t - \mathbf{m}_t\mathbf{m}_t^\top - \mathbf{m}_t\mathbf{m}_t^\top\mathbf{A}_t^\top dt - \mathbf{S}_t\mathbf{A}_t^\top dt + \mathbf{m}_t\mathbf{b}_t^\top dt + \\ &\mathbf{m}_t\mathbf{m}_t^\top\mathbf{A}_t^\top dt - \mathbf{m}_t\mathbf{b}_t^\top dt - \mathbf{m}_t\mathbf{m}_t^\top + \mathbf{m}_t\mathbf{m}_t^\top + \mathbf{m}_t\mathbf{m}_t^\top\mathbf{A}_t^\top dt - \\ &\mathbf{m}_t\mathbf{b}_t^\top dt - \mathbf{m}_t\mathbf{m}_t^\top\mathbf{A}_t^\top dt + \mathbf{m}_t\mathbf{b}_t^\top dt - \mathbf{A}_t\mathbf{m}_t\mathbf{m}_t^\top dt - \\ &\mathbf{A}_t\mathbf{S}_t dt + \mathbf{b}_t\mathbf{m}_t^\top dt + \mathbf{A}_t\mathbf{m}_t\mathbf{m}_t^\top dt - \mathbf{b}_t\mathbf{m}_t^\top dt + \mathbf{\Sigma} dt + \\ &\mathbf{A}_t\mathbf{m}_t\mathbf{m}_t^\top dt - \mathbf{b}_t\mathbf{m}_t^\top dt - \mathbf{A}_t\mathbf{m}_t\mathbf{m}_t^\top dt + \mathbf{b}_t\mathbf{m}_t^\top dt + O(dt^2) \end{aligned} \tag{A.72}$$

$$= -\mathbf{A}_t\mathbf{S}_t dt - \mathbf{S}_t\mathbf{A}_t^\top dt + \mathbf{\Sigma} dt + O(dt^2)\;, \tag{A.73}$$

Note that in Eq. (A.59) have been neglected terms beyond first order in dt. For the above derivations the following expectations (with respect to the approximate process q_t) have been used:

$$\left\langle \mathbf{x}_t \mathbf{x}_t^\top \right\rangle = \mathbf{m}_t \mathbf{m}_t^\top + \mathbf{S}_t \tag{A.74}$$

$$\left\langle \mathbf{x}_t \mathbf{m}_t^\top \right\rangle = \mathbf{m}_t \mathbf{m}_t^\top \tag{A.75}$$

$$\left\langle \mathbf{x}_t d\mathbf{m}_t^\top \right\rangle = \left\langle \mathbf{x}_t (-\mathbf{A}_t \mathbf{m}_t dt + \mathbf{b}_t dt)^\top \right\rangle \tag{A.76}$$

$$= \left\langle \mathbf{x}_t (-\mathbf{m}_t^\top \mathbf{A}_t^\top + \mathbf{b}_t^\top) dt \right\rangle \tag{A.77}$$

$$= \left\langle -\mathbf{x}_t \mathbf{m}_t^\top \mathbf{A}_t^\top \right\rangle dt + \left\langle \mathbf{x}_t \mathbf{b}_t^\top \right\rangle dt \tag{A.78}$$

$$= -\mathbf{m}_t \mathbf{m}_t^\top \mathbf{A}_t^\top dt + \mathbf{m}_t \mathbf{b}_t^\top dt \tag{A.79}$$

$$\left\langle \mathbf{m}_t \mathbf{x}_t^\top \right\rangle = \mathbf{m}_t \mathbf{m}_t^\top \tag{A.80}$$

$$\left\langle \mathbf{m}_t \mathbf{m}_t^\top \right\rangle = \mathbf{m}_t \mathbf{m}_t^\top \tag{A.81}$$

$$\left\langle \mathbf{m}_t d\mathbf{x}_t^\top \right\rangle = \mathbf{m}_t \left\langle d\mathbf{x}_t^\top \right\rangle \tag{A.82}$$

$$= \mathbf{m}_t (-\mathbf{A}_t \mathbf{m}_t dt + \mathbf{b}_t dt)^\top \tag{A.83}$$

$$= \mathbf{m}_t (-\mathbf{m}_t^\top \mathbf{A}_t^\top dt + \mathbf{b}_t^\top dt) \tag{A.84}$$

$$= -\mathbf{m}_t \mathbf{m}_t^\top \mathbf{A}_t^\top dt + \mathbf{m}_t \mathbf{b}_t^\top dt \tag{A.85}$$

$$\left\langle \mathbf{m}_t d\mathbf{m}_t^\top \right\rangle = \mathbf{m}_t \left\langle (-\mathbf{A}_t \mathbf{m}_t + \mathbf{b}_t)^\top \right\rangle dt \tag{A.86}$$

$$= \mathbf{m}_t \left\langle -\mathbf{m}_t^\top \mathbf{A}_t^\top + \mathbf{b}_t^\top \right\rangle dt \tag{A.87}$$

$$= -\mathbf{m}_t \mathbf{m}_t^\top \mathbf{A}_t^\top dt + \mathbf{m}_t \mathbf{b}_t^\top dt \tag{A.88}$$

$$\left\langle d\mathbf{x}_t d\mathbf{x}_t^\top \right\rangle = \left\langle (\mathbf{g}_L(\mathbf{x}_t) dt + \mathbf{\Sigma}^{1/2} d\mathbf{w}_t)(\mathbf{g}_L(\mathbf{x}_t) dt + \mathbf{\Sigma}^{1/2} d\mathbf{w}_t)^\top \right\rangle \tag{A.89}$$

$$= \left\langle (\mathbf{g}_L(\mathbf{x}_t) dt + \mathbf{\Sigma}^{1/2} d\mathbf{w}_t)(\mathbf{g}_L(\mathbf{x}_t)^\top dt + d\mathbf{w}_t^\top \mathbf{\Sigma}^{1/2}) \right\rangle \tag{A.90}$$

$$\begin{aligned} &= \underbrace{\left\langle \mathbf{g}_L(\mathbf{x}_t) \mathbf{g}_L(\mathbf{x}_t)^\top \right\rangle (dt^2)}_{O(dt^2)} + \underbrace{\left\langle \mathbf{g}_L(\mathbf{x}_t) dt d\mathbf{w}_t^\top \mathbf{\Sigma}^{1/2} \right\rangle}_{=0} \\ &\quad + \underbrace{\left\langle \mathbf{\Sigma}^{1/2} d\mathbf{w}_t \mathbf{g}_L(\mathbf{x}_t)^\top dt \right\rangle}_{=0} + \left\langle \mathbf{\Sigma}^{1/2} d\mathbf{w}_t d\mathbf{w}_t^\top \mathbf{\Sigma}^{1/2} \right\rangle \end{aligned} \tag{A.91}$$

$$= \mathbf{\Sigma}^{1/2} \underbrace{\left\langle d\mathbf{w}_t d\mathbf{w}_t^\top \right\rangle}_{=\,\mathrm{dt}\mathbf{I}} \mathbf{\Sigma}^{1/2} + O(dt^2) \tag{A.92}$$

$$= \mathbf{\Sigma}^{1/2} dt \mathbf{I} \mathbf{\Sigma}^{1/2} + O(dt^2) \tag{A.93}$$

$$= dt \mathbf{\Sigma} + O(dt^2) \tag{A.94}$$

$$\left\langle d\mathbf{x}_t d\mathbf{m}_t^\top \right\rangle = \left\langle (\mathbf{g}_L(\mathbf{x}_t) dt + \mathbf{\Sigma}^{1/2} d\mathbf{w}_t)(-\mathbf{A}_t \mathbf{m}_t dt + \mathbf{b}_t dt)^\top \right\rangle \tag{A.95}$$

$$= 0 + O(dt^2) \tag{A.96}$$

$$\left\langle d\mathbf{m}_t d\mathbf{x}_t^\top \right\rangle = \left\langle (-\mathbf{A}_t \mathbf{m}_t dt + \mathbf{b}_t dt)(\mathbf{g}_L(\mathbf{x}_t) dt + \mathbf{\Sigma}^{1/2} d\mathbf{w}_t)^\top \right\rangle \tag{A.97}$$

$$= 0 + O(dt^2) \tag{A.98}$$

$$\begin{aligned}
\left\langle d\mathbf{m}_t d\mathbf{m}_t^\top \right\rangle &= \left\langle (-\mathbf{A}_t\mathbf{m}_t dt + \mathbf{b}_t dt)(-\mathbf{A}_t\mathbf{m}_t dt + \mathbf{b}_t dt)^\top \right\rangle \qquad &\text{(A.99)}\\
&= O(dt^2) &\text{(A.100)}\\
\left\langle d\mathbf{m}_t \mathbf{m}_t^\top \right\rangle &= \left\langle (-\mathbf{A}_t\mathbf{m}_t dt + \mathbf{b}_t dt)\mathbf{m}_t^\top \right\rangle &\text{(A.101)}\\
&= -\mathbf{A}_t\mathbf{m}_t\mathbf{m}_t^\top dt + \mathbf{b}_t\mathbf{m}_t^\top dt &\text{(A.102)}\\
\left\langle d\mathbf{m}_t \mathbf{x}_t^\top \right\rangle &= \left\langle (-\mathbf{A}_t\mathbf{m}_t dt + \mathbf{b}_t dt)\mathbf{x}_t^\top \right\rangle &\text{(A.103)}\\
&= -\mathbf{A}_t\mathbf{m}_t\mathbf{m}_t^\top dt + \mathbf{b}_t\mathbf{m}_t^\top dt &\text{(A.104)}\\
\left\langle d\mathbf{x}_t \mathbf{m}_t^\top \right\rangle &= \langle d\mathbf{x}_t \rangle \mathbf{m}_t^\top &\text{(A.105)}\\
&= -\mathbf{A}_t\mathbf{m}_t\mathbf{m}_t^\top dt + \mathbf{m}_t\mathbf{m}_t^\top dt &\text{(A.106)}\\
\left\langle d\mathbf{x}_t \mathbf{x}_t^\top \right\rangle &= \left\langle (\mathbf{g}_L(\mathbf{x}_t)dt + \boldsymbol{\Sigma}^{1/2} d\mathbf{w}_t)\mathbf{x}_t^\top \right\rangle &\text{(A.107)}\\
&= \left\langle \mathbf{g}_L(\mathbf{x}_t)\mathbf{x}_t^\top dt + \boldsymbol{\Sigma}^{1/2} d\mathbf{w}_t\mathbf{x}_t^\top \right\rangle &\text{(A.108)}\\
&= \left\langle \mathbf{g}_L(\mathbf{x}_t)\mathbf{x}_t^\top \right\rangle dt + \underbrace{\boldsymbol{\Sigma}^{1/2} \left\langle d\mathbf{w}_t\mathbf{x}_t^\top \right\rangle}_{=0} &\text{(A.109)}\\
&= \left\langle (-\mathbf{A}_t\mathbf{x}_t + \mathbf{b}_t)\mathbf{x}_t^\top \right\rangle dt &\text{(A.110)}\\
&= -\mathbf{A}_t \left\langle \mathbf{x}_t\mathbf{x}_t^\top \right\rangle dt + \mathbf{b}_t \left\langle \mathbf{x}_t^\top \right\rangle dt &\text{(A.111)}\\
&= -\mathbf{A}_t\mathbf{m}_t\mathbf{m}_t^\top dt - \mathbf{A}_t\mathbf{S}_t dt + \mathbf{b}_t\mathbf{m}_t^\top dt &\text{(A.112)}\\
\left\langle \mathbf{x}_t d\mathbf{x}_t^\top \right\rangle &= \left\langle \mathbf{x}_t(\mathbf{g}_L(\mathbf{x}_t)dt + \boldsymbol{\Sigma}^{1/2} d\mathbf{w}_t)^\top \right\rangle &\text{(A.113)}\\
&= \left\langle \mathbf{x}_t\mathbf{g}_L(\mathbf{x}_t)^\top dt + \mathbf{x}_t d\mathbf{w}_t^\top \boldsymbol{\Sigma}^{1/2} \right\rangle &\text{(A.114)}\\
&= \left\langle \mathbf{x}_t\mathbf{g}_L(\mathbf{x}_t)^\top \right\rangle dt + \underbrace{\left\langle \mathbf{x}_t d\mathbf{w}_t^\top \right\rangle \boldsymbol{\Sigma}^{1/2}}_{=0} &\text{(A.115)}\\
&= \left\langle \mathbf{x}_t(-\mathbf{A}_t\mathbf{x}_t + \mathbf{b}_t)^\top \right\rangle dt &\text{(A.116)}\\
&= -\left\langle \mathbf{x}_t\mathbf{x}_t^\top \right\rangle \mathbf{A}_t^\top dt + \langle \mathbf{x}_t \rangle \mathbf{b}_t^\top dt &\text{(A.117)}\\
&= -\mathbf{m}_t\mathbf{m}_t^\top\mathbf{A}_t^\top dt - \mathbf{S}_t\mathbf{A}_t^\top dt + \mathbf{m}_t\mathbf{b}_t^\top dt &\text{(A.118)}
\end{aligned}$$

Lagrangian cost function

In order to ensure that constraints (A.58) and (A.59), are satisfied, the following $\mathcal{L}$agrangian is formulated:

$$\mathcal{L} = \mathcal{F}(q, \boldsymbol{\theta}, \boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t + \mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t)dt - \int_{t_0}^{t_f} \mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t + \mathbf{A}_t\mathbf{S}_t + \mathbf{S}_t\mathbf{A}_t^\top - \boldsymbol{\Sigma})\}dt\ , \quad \text{(A.119)}$$

where $\boldsymbol{\lambda}_t \in \Re^D$ and $\boldsymbol{\Psi}_t \in \Re^{D\times D}$ are time dependent Lagrange multipliers, with $\boldsymbol{\Psi}_t$ being symmetric.

Taking the functional derivative of (A.119) w.r.t. $\mathbf{A}_t$ yields:

$$\begin{aligned}
\nabla_{\mathbf{A}_t}\mathcal{L} &= \nabla_{\mathbf{A}_t}\Bigg(\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f}\mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t + \mathbf{A}_t\mathbf{S}_t + \mathbf{S}_t\mathbf{A}_t^\top - \boldsymbol{\Sigma})\}dt \\
&\qquad - \int_{t_0}^{t_f}\boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t + \mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t)dt\Bigg) && \text{(A.120)} \\
&= \nabla_{\mathbf{A}_t}\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \nabla_{\mathbf{A}_t}\int_{t_0}^{t_f}\mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t + \mathbf{A}_t\mathbf{S}_t + \mathbf{S}_t\mathbf{A}_t^\top - \boldsymbol{\Sigma})\}dt \\
&\qquad - \nabla_{\mathbf{A}_t}\int_{t_0}^{t_f}\boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t + \mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t)dt && \text{(A.121)} \\
&= \nabla_{\mathbf{A}_t}\left(\int_{t_0}^{t_f}E_{sde}(t)dt + \int_{t_0}^{t_f}E_{obs}(t)\sum_n \delta(t-t_n)dt + \mathrm{KL}[q_0\|p_0]\right) \\
&\qquad - 2\nabla_{\mathbf{A}_t}\int_{t_0}^{t_f}\mathrm{tr}\{\boldsymbol{\Psi}_t\mathbf{A}_t\mathbf{S}_t\}dt - \nabla_{\mathbf{A}_t}\int_{t_0}^{t_f}\boldsymbol{\lambda}_t^\top\mathbf{A}_t\mathbf{m}_t dt && \text{(A.122)} \\
&= \nabla_{\mathbf{A}_t}E_{sde}(t) - 2\boldsymbol{\Psi}_t\mathbf{S}_t - \boldsymbol{\lambda}_t\mathbf{m}_t^\top && \text{(A.123)}
\end{aligned}$$

where facts that $\boldsymbol{\Psi}_t$ and $\mathbf{S}_t$ are symmetric has been used.

In a similar way, the functional derivative of (A.119) w.r.t. $\mathbf{b}_t$ is:

$$\begin{aligned}
\nabla_{\mathbf{b}_t}\mathcal{L} &= \nabla_{\mathbf{b}_t}\Bigg(\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f}\mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t + \mathbf{A}_t\mathbf{S}_t + \mathbf{S}_t\mathbf{A}_t^\top - \boldsymbol{\Sigma})\}dt \\
&\qquad - \int_{t_0}^{t_f}\boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t + \mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t)dt\Bigg) && \text{(A.124)} \\
&= \nabla_{\mathbf{b}_t}\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \nabla_{\mathbf{b}_t}\int_{t_0}^{t_f}\mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t + \mathbf{A}_t\mathbf{S}_t + \mathbf{S}_t\mathbf{A}_t^\top - \boldsymbol{\Sigma})\}dt \\
&\qquad - \nabla_{\mathbf{b}_t}\int_{t_0}^{t_f}\boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t + \mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t)dt && \text{(A.125)} \\
&= \nabla_{\mathbf{b}_t}\left(\int_{t_0}^{t_f}E_{sde}(t)dt + \int_{t_0}^{t_f}E_{obs}(t)\sum_n \delta(t-t_n)dt + \mathrm{KL}[q_0\|p_0]\right) \\
&\qquad + \nabla_{\mathbf{b}_t}\int_{t_0}^{t_f}\boldsymbol{\lambda}_t^\top\mathbf{b}_t dt && \text{(A.126)} \\
&= \nabla_{\mathbf{b}_t}E_{sde}(t) + \boldsymbol{\lambda}_t && \text{(A.127)}
\end{aligned}$$

At this point one can derive the functional derivatives of the energy E_{sde}, with respect to the variational functions $\mathbf{A}_t$ and $\mathbf{b}_t$. First Equation (A.51) is differentiated w.r.t. $\mathbf{b}_t$:

$$\begin{aligned}
\nabla_{\mathbf{b}_t}E_{sde}(t) &= \nabla_{\mathbf{b}_t}\left(\frac{1}{2}\left\langle(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top\boldsymbol{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))\right\rangle_{q_t}\right) && \text{(A.128)} \\
&= \frac{1}{2}\left\langle\nabla_{\mathbf{b}_t}\left[(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top\boldsymbol{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))\right]\right\rangle_{q_t} && \text{(A.129)} \\
&= \frac{1}{2}\left\langle\nabla_{\mathbf{b}_t}\left[\left((\mathbf{f}(\mathbf{x}_t) + \mathbf{A}_t\mathbf{x}_t) - \mathbf{b}_t\right)^\top\boldsymbol{\Sigma}^{-1}\left((\mathbf{f}(\mathbf{x}_t) + \mathbf{A}_t\mathbf{x}_t) - \mathbf{b}_t\right)\right]\right\rangle_{q_t} && \text{(A.130)} \\
&= -\frac{1}{2}2\boldsymbol{\Sigma}^{-1}\left(\langle\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t)\rangle_{q_t}\right) && \text{(A.131)} \\
&= -\boldsymbol{\Sigma}^{-1}\left(\langle\mathbf{f}(\mathbf{x}_t)\rangle_{q_t} + \mathbf{A}_t\langle\mathbf{x}_t\rangle_{q_t} - \mathbf{b}_t\right) && \text{(A.132)}
\end{aligned}$$

Moreover, from Equations (A.127) and (A.132) we have:

$$\nabla_{\mathbf{b}_t} E_{sde}(t) = -\boldsymbol{\lambda}_t \tag{A.133}$$

$$\nabla_{\mathbf{b}_t} E_{sde}(t) = -\boldsymbol{\Sigma}^{-1}\left(\langle \mathbf{f}(\mathbf{x}_t)\rangle_{q_t} + \mathbf{A}_t \langle \mathbf{x}_t\rangle_{q_t} - \mathbf{b}_t\right) , \tag{A.134}$$

from the above equations it reads:

$$\boldsymbol{\lambda}_t = \boldsymbol{\Sigma}^{-1}\left(\langle \mathbf{f}(\mathbf{x}_t)\rangle_{q_t} + \mathbf{A}_t \langle \mathbf{x}_t\rangle_{q_t} - \mathbf{b}_t\right) \tag{A.135}$$

and by re-arranging the terms in the above equations we get:

$$\mathbf{b}_t = \langle \mathbf{f}(\mathbf{x}_t)\rangle_{q_t} + \mathbf{A}_t \mathbf{m}_t - \boldsymbol{\Sigma}\boldsymbol{\lambda}_t , \tag{A.136}$$

which is the update variational function of $\mathbf{b}_t$.

Following the same procedure, the differentiation of Eq. (A.51) w.r.t. $\mathbf{A}_t$ is:

$$\nabla_{\mathbf{A}_t} E_{sde}(t) = \nabla_{\mathbf{A}_t}\left(\frac{1}{2}\left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \boldsymbol{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))\right\rangle_{q_t}\right) \tag{A.137}$$

$$= \frac{1}{2}\left\langle \nabla_{\mathbf{A}_t}\left[\Big((\mathbf{f}(\mathbf{x}_t) - \mathbf{b}_t) - (-\mathbf{A}_t\mathbf{x}_t)\Big)^\top \boldsymbol{\Sigma}^{-1}\Big((\mathbf{f}(\mathbf{x}_t) - \mathbf{b}_t) - (-\mathbf{A}_t\mathbf{x}_t)\Big)\right]\right\rangle_{q_t} \tag{A.138}$$

$$= \frac{1}{2}2\boldsymbol{\Sigma}^{-1}\left\langle (\mathbf{f}(\mathbf{x}_t) + \mathbf{A}_t\mathbf{x}_t - \mathbf{b}_t)\mathbf{x}_t^\top\right\rangle_{q_t} \tag{A.139}$$

$$= \boldsymbol{\Sigma}^{-1}\left\langle \mathbf{f}(\mathbf{x}_t)\mathbf{x}_t^\top + \mathbf{A}_t\mathbf{x}_t\mathbf{x}_t^\top - \mathbf{b}_t\mathbf{x}_t^\top\right\rangle_{q_t} \tag{A.140}$$

$$= \boldsymbol{\Sigma}^{-1}\left(\left\langle \mathbf{f}(\mathbf{x}_t)\mathbf{x}_t^\top\right\rangle_{q_t} + \mathbf{A}_t\left\langle \mathbf{x}_t\mathbf{x}_t^\top\right\rangle_{q_t} - \mathbf{b}_t\left\langle \mathbf{x}_t^\top\right\rangle_{q_t}\right) \tag{A.141}$$

$$= \boldsymbol{\Sigma}^{-1}\left(\left\langle \mathbf{f}(\mathbf{x}_t)\mathbf{x}_t^\top\right\rangle_{q_t} + \mathbf{A}_t(\mathbf{m}_t\mathbf{m}_t^\top + \mathbf{S}_t) - \mathbf{b}_t\mathbf{m}_t^\top\right) \tag{A.142}$$

$$= \boldsymbol{\Sigma}^{-1}\left(\left\langle \mathbf{f}(\mathbf{x}_t)\mathbf{x}_t^\top\right\rangle_{q_t} + \mathbf{A}_t(\mathbf{m}_t\mathbf{m}_t^\top + \mathbf{S}_t) - \mathbf{b}_t\mathbf{m}_t^\top + \left\langle \mathbf{f}(\mathbf{x}_t)\mathbf{m}_t^\top\right\rangle_{q_t} - \left\langle \mathbf{f}(\mathbf{x}_t)\mathbf{m}_t^\top\right\rangle_{q_t}\right) \tag{A.143}$$

$$= \boldsymbol{\Sigma}^{-1}\left(\left\langle \mathbf{f}(\mathbf{x}_t)\mathbf{x}_t^\top\right\rangle_{q_t} - \left\langle \mathbf{f}(\mathbf{x}_t)\mathbf{m}_t^\top\right\rangle_{q_t} + \mathbf{A}_t\mathbf{S}_t\right) - \boldsymbol{\Sigma}^{-1}\left(-\langle \mathbf{f}(\mathbf{x}_t)\rangle_{q_t} - \mathbf{A}_t\mathbf{m}_t + \mathbf{b}_t\right)\mathbf{m}_t^\top \tag{A.144}$$

$$= \boldsymbol{\Sigma}^{-1}\left(\left\langle \mathbf{f}(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{m}_t)^\top\right\rangle_{q_t} + \mathbf{A}_t\mathbf{S}_t\right) - \nabla_{\mathbf{b}_t} E_{sde}(t)\mathbf{m}_t^\top \tag{A.145}$$

$$= \boldsymbol{\Sigma}^{-1}\left(\langle \nabla_{\mathbf{x}_t}\mathbf{f}(\mathbf{x}_t)\rangle_{q_t}\mathbf{S}_t + \mathbf{A}_t\mathbf{S}_t\right) - \nabla_{\mathbf{b}_t} E_{sde}(t)\mathbf{m}_t^\top \tag{A.146}$$

$$= \boldsymbol{\Sigma}^{-1}\left(\langle \nabla_{\mathbf{x}_t}\mathbf{f}(\mathbf{x}_t)\rangle_{q_t} + \mathbf{A}_t\right)\mathbf{S}_t - \nabla_{\mathbf{b}_t} E_{sde}(t)\mathbf{m}_t^\top \tag{A.147}$$

where we have made use of the Equation (A.136) and the following identity:

$$\langle \nabla_{\mathbf{x}_t}\mathbf{f}(\mathbf{x}_t)\rangle_{q_t} = \left\langle \mathbf{f}(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{m}_t)^\top\right\rangle_{q_t}\mathbf{S}_t^{-1} \tag{A.148}$$

Proof of identity given by Eq.(A.148) :

$$\langle \nabla_{\mathbf{x}_t} \mathbf{f}(\mathbf{x}_t) \rangle_{q_t} = \int_{-\infty}^{+\infty} \nabla_{\mathbf{x}_t} \mathbf{f}(\mathbf{x}_t) q(\mathbf{x}_t) d\mathbf{x}_t \tag{A.149}$$

$$= \int_{-\infty}^{+\infty} \left[\nabla_{\mathbf{x}_t} \left(\mathbf{f}(\mathbf{x}_t) q(\mathbf{x}_t) \right) - \mathbf{f}(\mathbf{x}_t) \nabla_{\mathbf{x}_t} q(\mathbf{x}_t) \right] d\mathbf{x}_t \tag{A.150}$$

$$= \underbrace{\int_{-\infty}^{+\infty} \nabla_{\mathbf{x}_t} \left[\mathbf{f}(\mathbf{x}_t) q(\mathbf{x}_t) \right] d\mathbf{x}_t}_{=0} + \int_{-\infty}^{+\infty} \mathbf{f}(\mathbf{x}_t) q(\mathbf{x}_t) \mathbf{S}_t^{-1} (\mathbf{x}_t - \mathbf{m}_t) d\mathbf{x}_t \tag{A.151}$$

$$= \int_{-\infty}^{+\infty} \mathbf{f}(\mathbf{x}_t) (\mathbf{x}_t - \mathbf{m}_t)^\top \mathbf{S}_t^{-1} q(\mathbf{x}_t) d\mathbf{x}_t \tag{A.152}$$

$$= \left\langle \mathbf{f}(\mathbf{x}_t) (\mathbf{x}_t - \mathbf{m}_t)^\top \right\rangle_{q_t} \mathbf{S}_t^{-1} \tag{A.153}$$

Note however, that in order for the first integral in Equation $(A.151)$, to be zero is is assumed that the unknown function $\mathbf{f}(\mathbf{x}_t)$, “moves” slower then the Gaussian approximation process $q(\mathbf{x}_t)$, as $\mathbf{x}_t \to \infty$.

The functional derivative of (A.119) w.r.t. $\mathbf{m}_t$ is given by:

$$\nabla_{\mathbf{m}_t} \mathcal{L} = \nabla_{\mathbf{m}_t} \left(\mathcal{F}_{\mathbf{\Sigma}}(q, \boldsymbol{\theta}) - \int_{t_0}^{t_f} \operatorname{tr}\{ \mathbf{\Psi}_t (\dot{\mathbf{S}}_t + 2\mathbf{A}_t \mathbf{S}_t - \mathbf{\Sigma}) \} dt \right.$$
$$\left. - \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top (\dot{\mathbf{m}}_t + \mathbf{A}_t \mathbf{m}_t - \mathbf{b}_t) dt \right) \tag{A.154}$$

$$= \nabla_{\mathbf{m}_t} \mathcal{F}(q, \boldsymbol{\theta}, \mathbf{\Sigma}) - \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} \operatorname{tr}\{ \mathbf{\Psi}_t (\dot{\mathbf{S}}_t + 2\mathbf{A}_t \mathbf{S}_t - \mathbf{\Sigma}) \} dt$$
$$- \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top (\dot{\mathbf{m}}_t + \mathbf{A}_t \mathbf{m}_t - \mathbf{b}_t) dt \tag{A.155}$$

$$= \nabla_{\mathbf{m}_t} \left(\int_{t_0}^{t_f} E_{sde}(t) dt + \int_{t_0}^{t_f} E_{obs}(t) \sum_n \delta(t - t_n) dt + \mathrm{KL}[q_0 \| p_0] \right)$$
$$- \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top \dot{\mathbf{m}}_t + \boldsymbol{\lambda}_t^\top \mathbf{A}_t \mathbf{m}_t - \boldsymbol{\lambda}_t^\top \mathbf{b}_t dt \tag{A.156}$$

$$= \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} E_{sde}(t) dt - \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top \dot{\mathbf{m}}_t dt - \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top \mathbf{A}_t \mathbf{m}_t dt \tag{A.157}$$

$$= \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} E_{sde}(t) dt + \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} \dot{\boldsymbol{\lambda}}_t^\top \mathbf{m}_t dt - \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top \mathbf{A}_t \mathbf{m}_t dt \tag{A.158}$$

Setting this expression equal to zero ($\nabla_{\mathbf{m}_t} \mathcal{L} = 0$) and then rearranging:

$$\nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} E_{sde}(t) dt + \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} \dot{\boldsymbol{\lambda}}_t^\top \mathbf{m}_t dt - \nabla_{\mathbf{m}_t} \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top \mathbf{A}_t \mathbf{m}_t dt = 0 \tag{A.159}$$

$$\nabla_{\mathbf{m}_t} E_{sde}(t) + \dot{\boldsymbol{\lambda}}_t - \mathbf{A}_t^\top \boldsymbol{\lambda}_t = 0 \ , \tag{A.160}$$

leads to an ODE that describes the time evolution of the Lagrange multiplier $\boldsymbol{\lambda}_t$:

$$\dot{\boldsymbol{\lambda}}_t = -\nabla_{\mathbf{m}_t} E_{sde}(t) + \mathbf{A}_t^\top \boldsymbol{\lambda}_t \ , \tag{A.161}$$

where we have used the fact (from product rule for differentiation) that:

$$\frac{d}{dt} (\boldsymbol{\lambda}_t^\top \mathbf{m}_t) = \frac{d\boldsymbol{\lambda}_t^\top}{dt} \mathbf{m}_t + \boldsymbol{\lambda}_t^\top \frac{d\mathbf{m}_t}{dt} \tag{A.162}$$

and also the assumption that at the final time, t_f, there are no consistency constraints, that is: $\boldsymbol{\lambda}_{t_f} = \boldsymbol{\Psi}_{t_f} = 0$.

Working the same way as above and taking the functional derivative of (A.119) w.r.t. $\mathbf{S}_t$ results:

$$\begin{aligned}
\nabla_{\mathbf{S}_t}\mathcal{L} = \nabla_{\mathbf{S}_t}\Bigg(&\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t + 2\mathbf{A}_t\mathbf{S}_t - \boldsymbol{\Sigma})\}dt \\
&- \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t + \mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t)dt\Bigg) &\text{(A.163)}\\
= {}& \nabla_{\mathbf{S}_t}\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} \mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t + 2\mathbf{A}_t\mathbf{S}_t - \boldsymbol{\Sigma})\}dt &\text{(A.164)}\\
= {}& \nabla_{\mathbf{S}_t}\left(\int_{t_0}^{t_f} E_{sde}(t)dt + \int_{t_0}^{t_f} E_{obs}(t)\sum_n \delta(t-t_n)dt + \mathrm{KL}[q_0\|p_0]\right)\\
&- \nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} \mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t + 2\mathbf{A}_t\mathbf{S}_t\}dt &\text{(A.165)}\\
= {}& \nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} E_{sde}(t)dt - \nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} \mathrm{tr}\{\boldsymbol{\Psi}_t\dot{\mathbf{S}}_t\}dt - 2\nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} \mathrm{tr}\{\boldsymbol{\Psi}_t\mathbf{A}_t\mathbf{S}_t\}dt &\text{(A.166)}\\
= {}& \nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} E_{sde}(t)dt + \nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} \mathrm{tr}\{\dot{\boldsymbol{\Psi}}_t\mathbf{S}_t\}dt - 2\nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} \mathrm{tr}\{\boldsymbol{\Psi}_t\mathbf{A}_t\mathbf{S}_t\}dt &\text{(A.167)}
\end{aligned}$$

Setting this expression equal to zero ($\nabla_{\mathbf{S}_t}\mathcal{L} = 0$) and then rearranging:

$$0 = \nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} E_{sde}(t)dt + \nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} \mathrm{tr}\{\dot{\boldsymbol{\Psi}}_t\mathbf{S}_t\}dt - 2\nabla_{\mathbf{S}_t}\int_{t_0}^{t_f} \mathrm{tr}\{\boldsymbol{\Psi}_t\mathbf{A}_t\mathbf{S}_t\}dt \tag{A.168}$$

$$0 = \nabla_{\mathbf{S}_t}E_{sde}(t) + \dot{\boldsymbol{\Psi}}_t - 2\boldsymbol{\Psi}_t\mathbf{A}_t\ , \tag{A.169}$$

leads to an ODE that describes the time evolution of the Lagrange multiplier $\boldsymbol{\Psi}_t$:

$$\dot{\boldsymbol{\Psi}}_t = -\nabla_{\mathbf{S}_t}E_{sde}(t) + 2\boldsymbol{\Psi}_t\mathbf{A}_t\ , \tag{A.170}$$

where the fact has been used (from properties of trace differentiation) that:

$$\begin{aligned}
\frac{d}{dt}\mathrm{tr}\{\boldsymbol{\Psi}_t\mathbf{S}_t\} &= \mathrm{tr}\{\frac{d}{dt}(\boldsymbol{\Psi}_t\mathbf{S}_t)\} &\text{(A.171)}\\
&= \mathrm{tr}\{\frac{d\boldsymbol{\Psi}_t}{dt}\mathbf{S}_t + \boldsymbol{\Psi}_t\frac{d\mathbf{S}_t}{dt}\} &\text{(A.172)}\\
&= \mathrm{tr}\{\frac{d\boldsymbol{\Psi}_t}{dt}\mathbf{S}_t\} + \mathrm{tr}\{\boldsymbol{\Psi}_t\frac{d\mathbf{S}_t}{dt}\}\ . &\text{(A.173)}
\end{aligned}$$

Along with the set of ordinary differential equations (A.161) and (A.170), which describe the time evolution of the Lagrange multipliers, whenever there is an observation the following ***jump-conditions*** apply.

First is considered the $\boldsymbol{\lambda}_t$ jump-condition, which is given by the following expression:

$$\boldsymbol{\lambda}(t_n^+) = \boldsymbol{\lambda}(t_n^-) - \nabla_{\mathbf{m}_t}E_{obs}(t_n)\ , \tag{A.174}$$

where the superscripts t_n^- and t_n^+ indicate times just before and after the observation time. Then the functional derivative of $E_{obs}(t_n)$ w.r.t. $\mathbf{m}_t$ is calculated, which plays the role of the jump amplitude:

$$\nabla_{\mathbf{m}_t} E_{obs}(t) = \nabla_{\mathbf{m}_t} \left(\frac{1}{2} \left\langle (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t)^\top \mathbf{R}^{-1} (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t) \right\rangle_{q_t} + \frac{d}{2} \ln(2\pi) + \frac{1}{2} \ln |\mathbf{R}| \right) \tag{A.175}$$

$$= \frac{1}{2} \nabla_{\mathbf{m}_t} \left(\left\langle (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t)^\top \mathbf{R}^{-1} (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t) \right\rangle_{q_t} \right) \tag{A.176}$$

$$= \frac{1}{2} \nabla_{\mathbf{m}_t} \left(\left\langle \mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{y}_t - \mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{x}_t - \mathbf{x}_t^\top \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{y}_t + \mathbf{x}_t^\top \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{x}_t \right\rangle_{q_t} \right) \tag{A.177}$$

$$= \frac{1}{2} \nabla_{\mathbf{m}_t} \left(\mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{y}_t - \mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{m}_t - \mathbf{m}_t^\top \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{y}_t + \operatorname{tr} \left\{ \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{S}_t \right\} + \mathbf{m}_t^\top \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{m}_t \right) \tag{A.178}$$

$$= \frac{1}{2} \nabla_{\mathbf{m}_t} \left(\mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{y}_t - 2\mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{m}_t + \operatorname{tr} \left\{ \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{S}_t \right\} + \mathbf{m}_t^\top \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{m}_t \right) \tag{A.179}$$

$$= \frac{1}{2} \left(-2\mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{H} + 2\mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{m}_t \right) \tag{A.180}$$

$$= -\mathbf{H}^\top \mathbf{R}^{-1} (\mathbf{y}_t - \mathbf{H}\mathbf{m}_t) \tag{A.181}$$

Finally we have:

$$\boldsymbol{\lambda}(t_n^+) = \boldsymbol{\lambda}(t_n^-) + \mathbf{H}^\top \mathbf{R}^{-1} (\mathbf{Y}_{t_n} - \mathbf{H}\mathbf{m}_{t_n}) \tag{A.182}$$

Then we consider the $\boldsymbol{\Psi}_t$ jump-condition which is given by the following expression:

$$\boldsymbol{\Psi}(t_n^+) = \boldsymbol{\Psi}(t_n^-) - \nabla_{\mathbf{S}_t} E_{obs}(t_n) \,, \tag{A.183}$$

Again the functional derivative of $E_{obs}(t_n)$ w.r.t. $\mathbf{S}_t$, plays the role of the jump-amplitude.

$$\nabla_{\mathbf{S}_t} E_{obs}(t) = \nabla_{\mathbf{S}_t} \left(\frac{1}{2} \left\langle (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t)^\top \mathbf{R}^{-1} (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t) \right\rangle_{q_t} + \frac{d}{2} \ln(2\pi) + \frac{1}{2} \ln |\mathbf{R}| \right) \tag{A.184}$$

$$= \frac{1}{2} \nabla_{\mathbf{S}_t} \left(\left\langle (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t)^\top \mathbf{R}^{-1} (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t) \right\rangle_{q_t} \right) \tag{A.185}$$

$$= \frac{1}{2} \nabla_{\mathbf{S}_t} \left(\left\langle \mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{y}_t - \mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{x}_t - \mathbf{x}_t^\top \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{y}_t + \mathbf{x}_t^\top \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{x}_t \right\rangle_{q_t} \right) \tag{A.186}$$

$$= \frac{1}{2} \nabla_{\mathbf{S}_t} \left(\mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{y}_t - \mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{m}_t - \mathbf{m}_t^\top \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{y}_t + \operatorname{tr} \left\{ \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{S}_t \right\} + \mathbf{m}_t^\top \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{m}_t \right) \tag{A.187}$$

$$= \frac{1}{2} \nabla_{\mathbf{S}_t} \left(\mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{y}_t - 2\mathbf{y}_t^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{m}_t + \operatorname{tr} \left\{ \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{S}_t \right\} + \mathbf{m}_t^\top \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}\mathbf{m}_t \right) \tag{A.188}$$

$$= \frac{1}{2} \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H} \,. \tag{A.189}$$

The final expression becomes:

$$\boldsymbol{\Psi}(t_n^+) = \boldsymbol{\Psi}(t_n^-) - \frac{1}{2} \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H} \,. \tag{A.190}$$

A.3 Parameter Estimation

Before computing the necessary gradients for estimating the parameters the Lagrangian equation (A.119), needs to be integrated by parts in order to make the boundary conditions explicit. That leads to the following expression:

$$\begin{aligned}
\mathcal{L} &= \mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \mathrm{tr}\Big\{\boldsymbol{\Psi}_t\Big(\dot{\mathbf{S}}_t + 2\mathbf{A}_t\mathbf{S}_t - \boldsymbol{\Sigma}\Big)\Big\}dt - \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top\Big(\dot{\mathbf{m}}_t + \mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t\Big)dt && \text{(A.191)}\\
&= \mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \mathrm{tr}\Big\{\boldsymbol{\Psi}_t\dot{\mathbf{S}}_t\Big\} + \mathrm{tr}\Big\{2\boldsymbol{\Psi}_t\mathbf{A}_t\mathbf{S}_t\Big\} - \mathrm{tr}\Big\{\boldsymbol{\Psi}_t\boldsymbol{\Sigma}\Big\}dt\\
&\quad - \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top\dot{\mathbf{m}}_t + \boldsymbol{\lambda}_t^\top\mathbf{A}_t\mathbf{m}_t - \boldsymbol{\lambda}_t^\top\mathbf{b}_t dt && \text{(A.192)}\\
&= \mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \frac{d}{dt}\mathrm{tr}\Big\{\boldsymbol{\Psi}_t\mathbf{S}_t\Big\} - \mathrm{tr}\Big\{\dot{\boldsymbol{\Psi}}_t\mathbf{S}_t\Big\} + \mathrm{tr}\Big\{2\boldsymbol{\Psi}_t\mathbf{A}_t\mathbf{S}_t\Big\} - \mathrm{tr}\Big\{\boldsymbol{\Psi}_t\boldsymbol{\Sigma}\Big\}dt\\
&\quad - \int_{t_0}^{t_f} \frac{d}{dt}(\boldsymbol{\lambda}_t^\top\mathbf{m}_t) - \dot{\boldsymbol{\lambda}}_t^\top\mathbf{m}_t + \boldsymbol{\lambda}_t^\top\mathbf{A}_t\mathbf{m}_t - \boldsymbol{\lambda}_t^\top\mathbf{b}_t dt && \text{(A.193)}\\
&= \mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \mathrm{tr}\Big\{\boldsymbol{\Psi}_t\Big(2\mathbf{A}_t\mathbf{S}_t - \boldsymbol{\Sigma}\Big) - \dot{\boldsymbol{\Psi}}_t\mathbf{S}_t\Big\}dt - \int_{t_0}^{t_f}\Big\{\boldsymbol{\lambda}_t^\top\Big(\mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t\Big) - \dot{\boldsymbol{\lambda}}_t^\top\mathbf{m}_t\Big\}dt\\
&\quad - \int_{t_0}^{t_f} \frac{d}{dt}\mathrm{tr}\Big\{\boldsymbol{\Psi}_t\mathbf{S}_t\Big\} + \frac{d}{dt}(\boldsymbol{\lambda}_t^\top\mathbf{m}_t)dt && \text{(A.194)}\\
&= \mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \mathrm{tr}\Big\{\boldsymbol{\Psi}_t\Big(2\mathbf{A}_t\mathbf{S}_t - \boldsymbol{\Sigma}\Big) - \dot{\boldsymbol{\Psi}}_t\mathbf{S}_t\Big\}dt - \int_{t_0}^{t_f}\Big\{\boldsymbol{\lambda}_t^\top\Big(\mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t\Big) - \dot{\boldsymbol{\lambda}}_t^\top\mathbf{m}_t\Big\}dt\\
&\quad - \int_{t_0}^{t_f} \frac{d}{dt}\Big(\mathrm{tr}\Big\{\boldsymbol{\Psi}_t\mathbf{S}_t\Big\} + (\boldsymbol{\lambda}_t^\top\mathbf{m}_t)\Big)dt && \text{(A.195)}\\
&= \mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \mathrm{tr}\Big\{\boldsymbol{\Psi}_t\Big(2\mathbf{A}_t\mathbf{S}_t - \boldsymbol{\Sigma}\Big) - \dot{\boldsymbol{\Psi}}_t\mathbf{S}_t\Big\}dt - \int_{t_0}^{t_f}\Big\{\boldsymbol{\lambda}_t^\top\Big(\mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t\Big) - \dot{\boldsymbol{\lambda}}_t^\top\mathbf{m}_t\Big\}dt\\
&\quad - \underbrace{\boldsymbol{\lambda}_{t_f}^\top\mathbf{m}_{t_f}}_{=0} + \boldsymbol{\lambda}_{t_0}^\top\mathbf{m}_{t_0} - \underbrace{\mathrm{tr}\Big\{\boldsymbol{\Psi}_{t_f}\mathbf{S}_{t_f}\Big\}}_{=0} + \mathrm{tr}\Big\{\boldsymbol{\Psi}_{t_0}\mathbf{S}_{t_0}\Big\} && \text{(A.196)}
\end{aligned}$$

this derives from the fact that at the final (algorithm) time, when the cost function has been minimised, the consistency constraints should be fulfilled. That means that both Lagrange multipliers are equal to zero.

A.3.1 Initial State

The initial approximate posterior process $q(\mathbf{x}_0)$ is equal to $\mathcal{N}(\mathbf{x}_0|\mathbf{m}_0,\mathbf{S}_0)$, where the initial true posterior process $p(\mathbf{x}_0)$ is chosen to be an isotropic Gaussian (i.e. $\mathcal{N}(\mathbf{x}_0|\boldsymbol{\mu}_0,\tau_0\mathbf{I})$).

Taking the derivative of (A.196) with respect to $\mathbf{m}_0$ leads to the following expression:

$$\begin{aligned}
\nabla_{\mathbf{m}_0}\mathcal{L} = \nabla_{\mathbf{m}_0}\Bigg(&\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f}\mathrm{tr}\Bigg\{\boldsymbol{\Psi}_t\Bigg(2\mathbf{A}_t\mathbf{S}_t-\boldsymbol{\Sigma}\Bigg)-\dot{\boldsymbol{\Psi}}_t\mathbf{S}_t\Bigg\}dt \\
&-\int_{t_0}^{t_f}\Bigg\{\boldsymbol{\lambda}_t^\top\Bigg(\mathbf{A}_t\mathbf{m}_t-\mathbf{b}_t\Bigg)-\dot{\boldsymbol{\lambda}}_t^\top\mathbf{m}_t\Bigg\}dt+\boldsymbol{\lambda}_0^\top\mathbf{m}_0+\mathrm{tr}\Bigg\{\boldsymbol{\Psi}_0\mathbf{S}_0\Bigg\}\Bigg) && \text{(A.197)}\\
&= \nabla_{\mathbf{m}_0}\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma})+\nabla_{\mathbf{m}_0}(\boldsymbol{\lambda}_0^\top\mathbf{m}_0) && \text{(A.198)}\\
&= \nabla_{\mathbf{m}_0}\mathrm{KL}[q(\mathbf{X}_0)||p(\mathbf{X}_0)]+\boldsymbol{\lambda}_0 && \text{(A.199)}\\
&= \boldsymbol{\lambda}_0+\frac{1}{2}\nabla_{\mathbf{m}_0}\Bigg(\ln|\tau_0\mathbf{I}\cdot\mathbf{S}_0^{-1}|+\mathrm{tr}\Bigg\{(\tau_0\mathbf{I})^{-1}\Big[(\mathbf{m}_0-\boldsymbol{\mu}_0)(\mathbf{m}_0-\boldsymbol{\mu}_0)^\top+\mathbf{S}_0-\tau_0\mathbf{I}\Big]\Bigg\}\Bigg) \\
& && \text{(A.200)}\\
&= \boldsymbol{\lambda}_0+\frac{1}{2}\nabla_{\mathbf{m}_0}\Bigg(\mathrm{tr}\Bigg\{(\tau_0\mathbf{I})^{-1}\Big[(\mathbf{m}_0-\boldsymbol{\mu}_0)(\mathbf{m}_0-\boldsymbol{\mu}_0)^\top\Big]\Bigg\}\Bigg) && \text{(A.201)}\\
&= \boldsymbol{\lambda}_0+\frac{1}{2}\mathrm{tr}\Bigg\{\nabla_{\mathbf{m}_0}\Bigg((\tau_0\mathbf{I})^{-1}\Big[(\mathbf{m}_0-\boldsymbol{\mu}_0)(\mathbf{m}_0-\boldsymbol{\mu}_0)^\top\Big]\Bigg)\Bigg\} && \text{(A.202)}\\
&= \boldsymbol{\lambda}_0+\frac{1}{2}\mathrm{tr}\Bigg\{\nabla_{\mathbf{m}_0}\Bigg(\tau_0^{-1}(\mathbf{m}_0-\boldsymbol{\mu}_0)(\mathbf{m}_0-\boldsymbol{\mu}_0)^\top\Bigg)\Bigg\} && \text{(A.203)}\\
&= \boldsymbol{\lambda}_0+\frac{1}{2}\mathrm{tr}\Bigg\{\tau_0^{-1}2(\mathbf{m}_0-\boldsymbol{\mu}_0)\Bigg\} && \text{(A.204)}\\
&= \boldsymbol{\lambda}_0+\tau_0^{-1}(\mathbf{m}_0-\boldsymbol{\mu}_0) && \text{(A.205)}
\end{aligned}$$

Taking the derivative of (A.196) with respect to $\mathbf{S}_0$ leads to the following expression:

$$\begin{aligned}
\nabla_{\mathbf{S}_0}\mathcal{L} = \nabla_{\mathbf{S}_0}\Bigg(&\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f}\mathrm{tr}\Bigg\{\boldsymbol{\Psi}_t\Bigg(2\mathbf{A}_t\mathbf{S}_t-\boldsymbol{\Sigma}\Bigg)-\dot{\boldsymbol{\Psi}}_t\mathbf{S}_t\Bigg\}dt \\
&-\int_{t_0}^{t_f}\Bigg\{\boldsymbol{\lambda}_t^\top\Bigg(\mathbf{A}_t\mathbf{m}_t-\mathbf{b}_t\Bigg)-\dot{\boldsymbol{\lambda}}_t^\top\mathbf{m}_t\Bigg\}dt+\boldsymbol{\lambda}_0^\top\mathbf{m}_0+\mathrm{tr}\Bigg\{\boldsymbol{\Psi}_0\mathbf{S}_0\Bigg\}\Bigg) && \text{(A.206)}\\
&= \nabla_{\mathbf{S}_0}\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma})+\nabla_{\mathbf{S}_0}\mathrm{tr}\Bigg\{\boldsymbol{\Psi}_0\mathbf{S}_0\Bigg\} && \text{(A.207)}\\
&= \nabla_{\mathbf{S}_0}\mathrm{KL}[q(\mathbf{X}_0)||p(\mathbf{X}_0)]+\boldsymbol{\Psi}_0 && \text{(A.208)}\\
&= \boldsymbol{\Psi}_0+\frac{1}{2}\nabla_{\mathbf{S}_0}\Bigg(\ln|\tau_0\mathbf{I}\cdot\mathbf{S}_0^{-1}|+\mathrm{tr}\Big\{(\tau_0\mathbf{I})^{-1}\Big[(\mathbf{m}_0-\boldsymbol{\mu}_0)(\mathbf{m}_0-\boldsymbol{\mu}_0)^\top+\mathbf{S}_0-\tau_0\mathbf{I}\Big]\Big\}\Bigg) \\
& && \text{(A.209)}\\
&= \boldsymbol{\Psi}_0+\frac{1}{2}\nabla_{\mathbf{S}_0}\ln|\tau_0\mathbf{I}\cdot\mathbf{S}_0^{-1}|+\frac{1}{2}\nabla_{\mathbf{S}_0}\mathrm{tr}\left\{(\tau_0\mathbf{I})^{-1}\mathbf{S}_0\right\} && \text{(A.210)}\\
&= \boldsymbol{\Psi}_0-\frac{1}{2}\mathbf{S}_0^{-1}+\frac{1}{2}(\tau_0\mathbf{I})^{-1} && \text{(A.211)}\\
&= \boldsymbol{\Psi}_0+\frac{1}{2}\Bigg(\tau_0^{-1}\mathbf{I}-\mathbf{S}_0^{-1}\Bigg) && \text{(A.212)}
\end{aligned}$$

A.3.2 Drift Parameter

The gradients that are associated with the drift parameters $\boldsymbol{\theta}$ depend only on the energy that comes from the SDE term in the posterior process. Hence:

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}}\mathcal{L} = \nabla_{\boldsymbol{\theta}}\Bigg(& \mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t_0}^{t_f} \mathrm{tr}\Big\{ \boldsymbol{\Psi}_t \Big(2\mathbf{A}_t\mathbf{S}_t - \boldsymbol{\Sigma} \Big) - \dot{\boldsymbol{\Psi}}_t\mathbf{S}_t \Big\} dt \\
& - \int_{t_0}^{t_f} \Big\{ \boldsymbol{\lambda}_t^\top \Big(\mathbf{A}_t\mathbf{m}_t - \mathbf{b}_t \Big) - \dot{\boldsymbol{\lambda}}_t^\top \mathbf{m}_t \Big\} dt + \boldsymbol{\lambda}_0^\top \mathbf{m}_0 + \mathrm{tr}\Big\{ \boldsymbol{\Psi}_0\mathbf{S}_0 \Big\} \Bigg) \qquad \text{(A.213)}
\end{aligned}
$$

$$
\begin{aligned}
&= \nabla_{\boldsymbol{\theta}}\mathcal{F}(q,\boldsymbol{\theta},\boldsymbol{\Sigma}) && \text{(A.214)}\\
&= \nabla_{\boldsymbol{\theta}}\Bigg(\int_{t_0}^{t_f} E_{sde}(t)dt + \int_{t_0}^{t_f} E_{obs}(t)\sum_n \delta(t-t_n)dt + \mathrm{KL}[q(\mathbf{X}_0)\|p(\mathbf{X}_0)] \Bigg) && \text{(A.215)}\\
&= \nabla_{\boldsymbol{\theta}} \int_{t_0}^{t_f} E_{sde}(t)dt && \text{(A.216)}\\
&= \int_{t_0}^{t_f} \nabla_{\boldsymbol{\theta}}E_{sde}(t)dt\ . && \text{(A.217)}
\end{aligned}
$$

To compute the above integral (A.217) one must first compute the derivative of $E_{sde}(t)$ w.r.t. $\boldsymbol{\theta}$ as follows:

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}}E_{sde}(t) &= \nabla_{\boldsymbol{\theta}}\left(\frac{1}{2}\Big\langle (\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{g})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{g}) \Big\rangle_{q_t} \right) && \text{(A.218)}\\
&= \frac{1}{2}\Big\langle \nabla_{\boldsymbol{\theta}}\Big[(\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{g})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{g}) \Big] \Big\rangle_{q_t} && \text{(A.219)}\\
&= \frac{1}{2}\Bigg\langle \nabla_{\boldsymbol{\theta}}\Big(\mathbf{f}_{\boldsymbol{\theta}}^\top \boldsymbol{\Sigma}^{-1}\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{f}_{\boldsymbol{\theta}}^\top \boldsymbol{\Sigma}^{-1}\mathbf{g} - \mathbf{g}^\top \boldsymbol{\Sigma}^{-1}\mathbf{f}_{\boldsymbol{\theta}} + \mathbf{g}^\top \boldsymbol{\Sigma}^{-1}\mathbf{g} \Big) \Bigg\rangle_{q_t} && \text{(A.220)}\\
&= \frac{1}{2}\Big\langle \nabla_{\boldsymbol{\theta}}(\mathbf{f}_{\boldsymbol{\theta}}^\top \boldsymbol{\Sigma}^{-1}\mathbf{f}_{\boldsymbol{\theta}}) - \nabla_{\boldsymbol{\theta}}(\mathbf{f}_{\boldsymbol{\theta}}^\top \boldsymbol{\Sigma}^{-1}\mathbf{g}) - \nabla_{\boldsymbol{\theta}}(\mathbf{g}^\top \boldsymbol{\Sigma}^{-1}\mathbf{f}_{\boldsymbol{\theta}}) \Big\rangle_{q_t} && \text{(A.221)}\\
&= \frac{1}{2}\Big\langle (\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}^\top)\boldsymbol{\Sigma}^{-1}\mathbf{f}_{\boldsymbol{\theta}} + \mathbf{f}_{\boldsymbol{\theta}}^\top \boldsymbol{\Sigma}^{-1}(\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}) - \mathbf{g}^\top \boldsymbol{\Sigma}^{-1}(\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}) - \mathbf{g}^\top \boldsymbol{\Sigma}^{-1}(\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}) \Big\rangle_{q_t} && \text{(A.222)}\\
&= \frac{1}{2}\Big\langle 2\mathbf{f}_{\boldsymbol{\theta}}^\top \boldsymbol{\Sigma}^{-1}(\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}) - 2\mathbf{g}^\top \boldsymbol{\Sigma}^{-1}(\nabla_{\boldsymbol{\theta}}\mathbf{f}_{\boldsymbol{\theta}}) \Big\rangle_{q_t} && \text{(A.223)}\\
&= \Bigg\langle \Big(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t) \Big)^\top \boldsymbol{\Sigma}^{-1} \Big(\nabla_{\boldsymbol{\theta}}\mathbf{f}(\mathbf{x}_t) \Big) \Bigg\rangle_{q_t}\ , && \text{(A.224)}
\end{aligned}
$$

where we have used the shorthand notations $\mathbf{f}_{\boldsymbol{\theta}}$ for $\mathbf{f}(\mathbf{x}_t)$ and $\mathbf{g}$ instead of $\mathbf{g}_L(\mathbf{x}_t)$.

A.3.3 System Noise Covariance Parameter

The estimation of the system noise is of great importance because the system noise along with the drift parameter determines the dynamics of the system. The gradient of (A.196) with respect to the system noise covariance $\mathbf{\Sigma}$ is given by:

$$\begin{aligned}\nabla_{\mathbf{\Sigma}}\mathcal{L} &= \nabla_{\mathbf{\Sigma}}\Bigg(\mathcal{F}(q,\boldsymbol{\theta},\mathbf{\Sigma}) - \int_{t_0}^{t_f}\mathrm{tr}\Big\{\mathbf{\Psi}_t\Big(2\mathbf{A}_t\mathbf{S}_t-\mathbf{\Sigma}\Big)-\dot{\mathbf{\Psi}}_t\mathbf{S}_t\Big\}dt \\ &\qquad - \int_{t_0}^{t_f}\Big\{\boldsymbol{\lambda}_t^{\top}\Big(\mathbf{A}_t\mathbf{m}_t-\mathbf{b}_t\Big)-\dot{\boldsymbol{\lambda}}_t^{\top}\mathbf{m}_t\Big\}dt+\boldsymbol{\lambda}_0^{\top}\mathbf{m}_0+\mathrm{tr}\Big\{\mathbf{\Psi}_0\mathbf{S}_0\Big\}\Bigg) \qquad \text{(A.225)} \\ &= \nabla_{\mathbf{\Sigma}}\mathcal{F}(q,\boldsymbol{\theta},\mathbf{\Sigma})+\nabla_{\mathbf{\Sigma}}\int_{t_0}^{t_f}\mathrm{tr}\Big\{\mathbf{\Psi}_t\mathbf{\Sigma}\Big\}dt \qquad \text{(A.226)} \\ &= \int_{t_0}^{t_f}\nabla_{\mathbf{\Sigma}}E_{sde}(t)dt+\int_{t_0}^{t_f}\nabla_{\mathbf{\Sigma}}\mathrm{tr}\Big\{\mathbf{\Psi}_t\mathbf{\Sigma}\Big\}dt \qquad \text{(A.227)} \\ &= \int_{t_0}^{t_f}\nabla_{\mathbf{\Sigma}}E_{sde}(t)dt+\int_{t_0}^{t_f}\mathbf{\Psi}_t dt \qquad \text{(A.228)}\end{aligned}$$

and the gradient of E_{sde} with respect to $\mathbf{\Sigma}$ is given by:

$$\begin{aligned}\nabla_{\mathbf{\Sigma}}E_{sde}(t) &= \nabla_{\mathbf{\Sigma}}\left[\frac{1}{2}\Big\langle(\mathbf{f}_{\boldsymbol{\theta}}-\mathbf{g})^{\top}\mathbf{\Sigma}^{-1}(\mathbf{f}_{\boldsymbol{\theta}}-\mathbf{g})\Big\rangle_{q_t}\right] \qquad \text{(A.229)} \\ &= \frac{1}{2}\Big\langle\nabla_{\mathbf{\Sigma}}\Big[(\mathbf{f}_{\boldsymbol{\theta}}-\mathbf{g})^{\top}\mathbf{\Sigma}^{-1}(\mathbf{f}_{\boldsymbol{\theta}}-\mathbf{g})\Big]\Big\rangle_{q_t} \qquad \text{(A.230)} \\ &= -\frac{1}{2}\Big\langle\mathbf{\Sigma}^{-\top}(\mathbf{f}_{\boldsymbol{\theta}}-\mathbf{g})(\mathbf{f}_{\boldsymbol{\theta}}-\mathbf{g})^{\top}\mathbf{\Sigma}^{-\top}\Big\rangle_{q_t} \qquad \text{(A.231)} \\ &= -\frac{1}{2}\mathbf{\Sigma}^{-1}\Big\langle(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))^{\top}\Big\rangle_{q_t}\mathbf{\Sigma}^{-1}\,, \qquad \text{(A.232)}\end{aligned}$$

because matrix $\mathbf{\Sigma}$ is symmetric.

A.3.4 Observation Noise Covariance Parameter

Although estimation of the noise related to the observable values is not addressed in the thesis, the estimation of the noise covariance parameters is a straightforward extension and in some sense completes the variational framework. The gradient of (A.196) with respect to the observation noise covariance $\mathbf{R}$ is given by:

$$\begin{aligned}\nabla_{\mathbf{R}}\mathcal{L} &= \nabla_{\mathbf{R}}\Bigg(\mathcal{F}(q,\boldsymbol{\theta},\mathbf{\Sigma}) - \int_{t_0}^{t_f}\mathrm{tr}\Big\{\mathbf{\Psi}_t\Big(2\mathbf{A}_t\mathbf{S}_t-\mathbf{\Sigma}\Big)-\dot{\mathbf{\Psi}}_t\mathbf{S}_t\Big\}dt \\ &\qquad - \int_{t_0}^{t_f}\Big\{\boldsymbol{\lambda}_t^{\top}\Big(\mathbf{A}_t\mathbf{m}_t-\mathbf{b}_t\Big)-\dot{\boldsymbol{\lambda}}_t^{\top}\mathbf{m}_t\Big\}dt+\boldsymbol{\lambda}_0^{\top}\mathbf{m}_0+\mathrm{tr}\Big\{\mathbf{\Psi}_0\mathbf{S}_0\Big\}\Bigg) \qquad \text{(A.233)} \\ &= \nabla_{\mathbf{R}}\mathcal{F}(q,\boldsymbol{\theta},\mathbf{\Sigma}) \qquad \text{(A.234)} \\ &= \nabla_{\mathbf{R}}\Bigg(\int_{t_0}^{t_f}E_{sde}(t)dt+\int_{t_0}^{t_f}E_{obs}(t)\sum_{n}\delta(t-t_n)dt+\mathrm{KL}[q(\mathbf{X}_0)\|p(\mathbf{X}_0)]\Bigg) \qquad \text{(A.235)} \\ &= \nabla_{\mathbf{R}}\int_{t_0}^{t_f}E_{obs}(t)\sum_{n}\delta(t-t_n)dt\;. \qquad \text{(A.236)}\end{aligned}$$

Therefore to compute the above gradient, one has to compute first the gradient of E_{obs} w.r.t. $\mathbf{R}$.

$$\nabla_{\mathbf{R}} E_{obs}(t) = \nabla_{\mathbf{R}} \left(\frac{1}{2} \left\langle (\mathbf{y}_t - h(\mathbf{x}_t))^\top \mathbf{R}^{-1} (\mathbf{y}_t - h(\mathbf{x}_t)) \right\rangle_{q_t} + \frac{d}{2} \ln(2\pi) + \frac{1}{2} \ln|\mathbf{R}| \right) \tag{A.237}$$

$$= \frac{1}{2} \nabla_{\mathbf{R}} \left\langle (\mathbf{y}_t - h(\mathbf{x}_t))^\top \mathbf{R}^{-1} (\mathbf{y}_t - h(\mathbf{x}_t)) \right\rangle_{q_t} + \frac{1}{2} \nabla_{\mathbf{R}} \ln|\mathbf{R}| \tag{A.238}$$

$$= -\frac{1}{2} \left\langle \mathbf{R}^{-1} (\mathbf{y}_t - h(\mathbf{x}_t))(\mathbf{y}_t - h(\mathbf{x}_t))^\top \mathbf{R}^{-1} \right\rangle_{q_t} + \frac{1}{2} \mathbf{R}^{-1} \tag{A.239}$$

$$= -\frac{1}{2} \mathbf{R}^{-1} \left\langle (\mathbf{y}_t - h(\mathbf{x}_t))(\mathbf{y}_t - h(\mathbf{x}_t))^\top \right\rangle_{q_t} \mathbf{R}^{-1} + \frac{1}{2} \mathbf{R}^{-1} \tag{A.240}$$

$$= \frac{1}{2} \mathbf{R}^{-1} \left(\mathbf{I} - \left\langle (\mathbf{y}_t - h(\mathbf{x}_t))(\mathbf{y}_t - h(\mathbf{x}_t))^\top \right\rangle_{q_t} \mathbf{R}^{-1} \right) . \tag{A.241}$$

A.4 Summary

To sum up, Appendix A presents a complete derivation the original VGPA framework. The mathematical expressions cover the full multivariate case, although for the univariate cases more simplifications apply. The specific expressions of the aforementioned equations, for the systems tested here, are presented in Appendix D. In deriving the above equations many useful matrix identities were found in *The Matrix Cookbook* (Petersen and Petersen, 2007).

B Computing the new gradients of the RBF extension

Chapter 5 introduced a new RBF re-parametrisation of the initial variational problem ending up with two sets of weights, one for the linear variational parameter $\{\boldsymbol{A}_i\}_{i=0}^{L_{Ab}}$ and one for the offset parameter $\{\boldsymbol{b}_i\}_{i=0}^{L_{Ab}}$, whose optimal values need to be inferred by means of a gradient based optimisation algorithm (SCG).

As usual these algorithms need the gradients of the objective (cost) function, with respect to the parameters that are optimised. In this case the necessary gradients that need to be computed are those of the approximate $\mathcal{L}$agrangian function (see Eq. (5.5)), with respect to these matrices/vectors, as shown in Eq. (5.8). Following the derivations of the initial (VGPA) algorithm (Appendix A), the desired expressions are derived in the following two sections.

Once all the necessary derivatives have been computed they are packed all together and a batch optimisation of the cost function leads to the optimal approximate posterior process. One obvious difference, comparing to the original VGPA framework, is that all the previous partial derivatives have to be computed separately, for each basis function. In a serial implementation, as the one presented in the current work, this has a negative effect on the total computational performance of the new approach. However, there is no fundamental reason to prevent a parallel implementation in computing these derivatives, to speed up the overall performance.

B.1 Gradient of the approximate $\mathcal{L}$agrangian with respect to $\mathbf{A}$ weights.

To compute the required gradient $\nabla_{\mathbf{A}}\tilde{\mathcal{L}}$, with $\mathbf{A} \equiv \{\mathbf{A}_i\}_0^{L_{Ab}}$, one must first compute the partial derivatives of $\tilde{\mathcal{L}}$ with respect to $\mathbf{A}_i \;\forall\, i \in \{0,\, 1,\, 2,\, \ldots,\, L_{Ab}\}$. This is done as follows:

$$\begin{aligned}
\frac{\partial\tilde{\mathcal{L}}}{\partial\mathbf{A}_i} = \frac{\partial}{\partial\mathbf{A}_i}\Bigg(& \tilde{\mathcal{F}}(q_t,\theta,\mathbf{\Sigma}) \\
& - \int_{t_0}^{t_f} \Big\{ \boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t + \tilde{\mathbf{A}}_t\mathbf{m}_t - \tilde{\mathbf{b}}_t) + \mathrm{tr}\{\mathbf{\Psi}_t(\dot{\mathbf{S}}_t + \tilde{\mathbf{A}}_t\mathbf{S}_t + \mathbf{S}_t\tilde{\mathbf{A}}_t^\top - \mathbf{\Sigma})\}\Big\}\, dt \Bigg) \qquad \text{(B.1)}\\
= \frac{\partial}{\partial\mathbf{A}_i} & \tilde{\mathcal{F}}(q_t,\theta,\mathbf{\Sigma}) \\
- \frac{\partial}{\partial\mathbf{A}_i} & \int_{t_0}^{t_f} \Big\{ \boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t + \tilde{\mathbf{A}}_t\mathbf{m}_t - \tilde{\mathbf{b}}_t) + \mathrm{tr}\{\mathbf{\Psi}_t(\dot{\mathbf{S}}_t + \tilde{\mathbf{A}}_t\mathbf{S}_t + \mathbf{S}_t\tilde{\mathbf{A}}_t^\top - \mathbf{\Sigma})\}\Big\}\, dt \qquad \text{(B.2)}\\
= \frac{\partial}{\partial\mathbf{A}_i} & \tilde{\mathcal{F}}(q_t,\theta,\mathbf{\Sigma}) - \frac{\partial}{\partial\mathbf{A}_i}\int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top\tilde{\mathbf{A}}_t\mathbf{m}_t\; dt - \frac{\partial}{\partial\mathbf{A}_i}\int_{t_0}^{t_f} \mathrm{tr}\{\mathbf{\Psi}_t\tilde{\mathbf{A}}_t\mathbf{S}_t\} + \mathrm{tr}\{\mathbf{\Psi}_t\mathbf{S}_t\tilde{\mathbf{A}}_t^\top\}\; dt \qquad \text{(B.3)}\\
= \frac{\partial}{\partial\mathbf{A}_i} & \tilde{\mathcal{F}}(q_t,\theta,\mathbf{\Sigma}) - \int_{t_0}^{t_f} \boldsymbol{\lambda}_t\mathbf{m}_t^\top\phi_i(t)\; dt - 2\int_{t_0}^{t_f} \mathbf{\Psi}_t\mathbf{S}_t\phi_i(t)\; dt\;, \qquad \text{(B.4)}
\end{aligned}$$

where $\tilde{\mathbf{A}}_t$ has been substituted with $\sum_{i=0}^{L_A}\boldsymbol{A}_i \times \phi_i(t)$, according to Equation (5.4).

B.2 Gradient of the approximate $\mathcal{L}$agrangian with respect to $\mathbf{b}$ weights.

In a similar manner, the required gradient $\nabla_{\mathbf{b}}\tilde{\mathcal{L}}$, with $\mathbf{b} \equiv \{\mathbf{b}_i\}_0^{L_{Ab}}$, is computed after the partial derivatives of $\tilde{\mathcal{L}}$ with respect to $\mathbf{b}_i \;\forall\, i \in \{0,\, 1,\, 2,\, \ldots,\, L_{Ab}\}$, have been calculated.

$$\begin{aligned}
\frac{\partial\tilde{\mathcal{L}}}{\partial\mathbf{b}_i} = \frac{\partial}{\partial\mathbf{b}_i}\Bigg(& \tilde{\mathcal{F}}(q_t,\theta,\mathbf{\Sigma}) \\
& - \int_{t_0}^{t_f} \Big\{ \boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t + \tilde{\mathbf{A}}_t\mathbf{m}_t - \tilde{\mathbf{b}}_t) + \mathrm{tr}\{\mathbf{\Psi}_t(\dot{\mathbf{S}}_t + \tilde{\mathbf{A}}_t\mathbf{S}_t + \mathbf{S}_t\tilde{\mathbf{A}}_t^\top - \mathbf{\Sigma})\}\Big\}\, dt \Bigg) \qquad \text{(B.5)}\\
= \frac{\partial}{\partial\mathbf{b}_i} & \tilde{\mathcal{F}}(q_t,\theta,\mathbf{\Sigma}) + \frac{\partial}{\partial\mathbf{b}_i}\int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top\tilde{\mathbf{b}}_t\; dt \qquad \text{(B.6)}\\
= \frac{\partial}{\partial\mathbf{b}_i} & \tilde{\mathcal{F}}(q_t,\theta,\mathbf{\Sigma}) + \int_{t_0}^{t_f} \boldsymbol{\lambda}_t^\top\phi_i(t)\; dt\;, \qquad \text{(B.7)}
\end{aligned}$$

where $\tilde{\mathbf{b}}_t = \sum_{i=0}^{L_b}\boldsymbol{b}_i \times \phi_i(t)$. The partial derivatives of the approximate value of the free energy $\tilde{\mathcal{F}}(q_t,\theta,\mathbf{\Sigma})$, with respect to the weights $\mathbf{A}_i$ and $\mathbf{b}_i$ are computed in a similar way.

C Computing the new gradients of the LP extension

The new parametrisation of the initial variational problem in terms of local polynomials, as presented in Chapter 6, concluded with two sets of coefficients, one for the linear variational parameter $\{\boldsymbol{A}^j\}_{j=0}^J$ and one for the offset parameter $\{\boldsymbol{b}^j\}_{j=0}^J$, whose optimal values need to be inferred by means of a gradient optimisation algorithm. In a similar fashion as Appendix B, the necessary gradients that need to be estimated are those of the approximate $\mathcal{L}$agrangian function (see Eq. 6.3), with respect to these matrices/vectors.

Note that the gradients have to be computed within each sub-interval separately. In the present implementation this is coded serially for simplicity, which has an increased computational cost to the total performance of the LP algorithm. However, a parallel implementation is encouraged to improve the speed of computations.

C.1 Gradient of the approximate $\mathcal{L}$agrangian with respect to $\mathbf{A}$ coefficients.

To compute the required gradient $\nabla_{\mathbf{A}}\tilde{\mathcal{L}}$, with $\mathbf{A} \equiv \{\boldsymbol{A}^j\}_{j=0}^J$, one must first compute the partial derivatives of $\tilde{\mathcal{L}}$ with respect to $\boldsymbol{A}^j \; \forall \; j \in \{0, 1, 2, \ldots, J\}$. This is done as follows:

$$
\begin{aligned}
\frac{\partial \tilde{\mathcal{L}}^j}{\partial \mathbf{A}^j} &= \frac{\partial}{\partial \mathbf{A}^j}\Bigg(\tilde{\mathcal{F}}^j(q(\mathbf{x}_t),\boldsymbol{\theta},\boldsymbol{\Sigma}) \\
&\quad - \int_{t\in T^j}\Big\{\boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t+\tilde{\mathbf{A}}_t^j\mathbf{m}_t-\tilde{\mathbf{b}}_t^j)+\mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t+\tilde{\mathbf{A}}_t^j\mathbf{S}_t+\mathbf{S}_t\tilde{\mathbf{A}}_t^{j\top}-\boldsymbol{\Sigma})\}\Big\}\,dt\Bigg) \qquad \text{(C.1)} \\
&= \frac{\partial}{\partial \mathbf{A}^j}\tilde{\mathcal{F}}^j(q(\mathbf{x}_t),\boldsymbol{\theta},\boldsymbol{\Sigma}) - \frac{\partial}{\partial \mathbf{A}^j}\int_{t\in T^j}\boldsymbol{\lambda}_t^\top\tilde{\mathbf{A}}_t^j\mathbf{m}_t\,dt \\
&\quad - \frac{\partial}{\partial \mathbf{A}^j}\int_{t\in T^j}\mathrm{tr}\{\boldsymbol{\Psi}_t\tilde{\mathbf{A}}_t^j\mathbf{S}_t\}+\mathrm{tr}\{\boldsymbol{\Psi}_t\mathbf{S}_t\tilde{\mathbf{A}}_t^{j\top}\}\,dt \qquad \text{(C.2)} \\
&= \frac{\partial}{\partial \mathbf{A}^j}\tilde{\mathcal{F}}^j(q(\mathbf{x}_t),\boldsymbol{\theta},\boldsymbol{\Sigma}) - \int_{t\in T^j}\boldsymbol{\lambda}_t\mathbf{m}_t^\top\boldsymbol{p}^j(t)dt - 2\int_{t\in T^j}\boldsymbol{\Psi}_t\mathbf{S}_t\boldsymbol{p}^j(t)dt\ . \qquad \text{(C.3)}
\end{aligned}
$$

where $\tilde{\mathbf{A}}_t^j$ has been replaced with $\boldsymbol{A}^j \times \boldsymbol{p}^j(t)$, according to Equation (6.4).

C.2 Gradient of the approximate $\mathcal{L}$agrangian with respect to b coefficients.

To compute the required gradient $\nabla_{\mathbf{b}}\tilde{\mathcal{L}}$, with $\mathbf{b} \equiv \{\boldsymbol{b}^j\}_{j=0}^J$, one must first compute the partial derivatives of $\tilde{\mathcal{L}}$ with respect to $\boldsymbol{b}^j\ \forall\ j\ \in \{0,\ 1,\ 2,\ \ldots,\ J\}$:

$$
\begin{aligned}
\frac{\partial \tilde{\mathcal{L}}^j}{\partial \mathbf{b}^j} &= \frac{\partial}{\partial \mathbf{b}^j}\Bigg(\tilde{\mathcal{F}}^j(q(\mathbf{x}_t),\boldsymbol{\theta},\boldsymbol{\Sigma}) \\
&\quad - \int_{t\in T^j}\Big\{\boldsymbol{\lambda}_t^\top(\dot{\mathbf{m}}_t+\tilde{\mathbf{A}}_t^j\mathbf{m}_t-\tilde{\mathbf{b}}_t^j)+\mathrm{tr}\{\boldsymbol{\Psi}_t(\dot{\mathbf{S}}_t+\tilde{\mathbf{A}}_t^j\mathbf{S}_t+\mathbf{S}_t\tilde{\mathbf{A}}_t^{j\top}-\boldsymbol{\Sigma})\}\Big\}\,dt\Bigg) \qquad \text{(C.4)} \\
&= \frac{\partial}{\partial \mathbf{b}^j}\tilde{\mathcal{F}}^j(q(\mathbf{x}_t),\boldsymbol{\theta},\boldsymbol{\Sigma}) + \frac{\partial}{\partial \mathbf{b}^j}\int_{t\in T^j}\boldsymbol{\lambda}_t^\top\tilde{\mathbf{b}}_t^j\,dt \qquad \text{(C.5)} \\
&= \frac{\partial}{\partial \mathbf{b}^j}\tilde{\mathcal{F}}^j(q(\mathbf{x}_t),\boldsymbol{\theta},\boldsymbol{\Sigma}) + \int_{t\in T^j}\boldsymbol{\lambda}_t^\top\boldsymbol{p}^j(t)\,dt\ \text{,where } \tilde{\mathbf{b}}_t^j = \boldsymbol{B}^j\times\boldsymbol{p}^j(t). \qquad \text{(C.6)}
\end{aligned}
$$

The partial derivatives of the approximate value of the free energy $\tilde{\mathcal{F}}^j(q(\mathbf{x}_t),\boldsymbol{\theta},\boldsymbol{\Sigma})$, with respect to the coefficients $\mathbf{A}^j$ and $\mathbf{b}^j$, are computed in a similar way.

D Analytic expressions of the systems studied

This Appendix provides detailed analytic derivations of the energy terms and related gradients for the univariate systems (OU and DW), as well as the three dimensional Lorenz '63 (L3D). Once the complete analytic expression of the energy related to the SDE is derived (E_{sde}), the gradients of this quantity will be developed. The optimal initial values (for the gradient optimisation procedure) of the linear and offset parameters $\mathbf{A}(0)$ and $\mathbf{b}(0)$ will be given. In addition, when the analytic expressions are not available an alternative method to obtain the necessary approximate expressions is provided.

D.1 Ornstein - Uhlenbeck (OU) system equations

The first system derived analytically is the one dimensional *Ornstein-Uhlenbeck* (OU) process. The stochastic differential equation that describes the time evolution of this linear process, as introduced in Chapter 3, is given by:

$$dx_t = -\theta(x_t - \mu)dt + \sigma^2 dw_t \tag{D.1}$$

where $\theta > 0$, is the mean reversion rate and μ is the mean value, which is often set to zero. Hence the drift function of this one dimensional SDE is:

$$\mathbf{f}(x_t) = -\theta(x_t - \mu) \tag{D.2}$$

Before proceeding with the expression of the energy term, two important averages are computed in advance that will help the following derivations. The first is the averaged drift function with respect to the approximate Gaussian distribution q_t[1]. This is given by:

$$\begin{aligned} \langle \mathbf{f}(x_t) \rangle_{q_t} &= \langle -\theta(x_t - \mu) \rangle_{q_t} && \text{(D.3)} \\ &= -\theta \langle x_t \rangle_{q_t} + \theta\mu && \text{(D.4)} \\ &= \theta(\mu - m_t) && \text{(D.5)} \end{aligned}$$

The second expression is the averaged gradient of the drift function with respect to x_t, which is derived as follows:

$$\begin{aligned} \langle \nabla_{x_t} \mathbf{f}(x_t) \rangle_{q_t} &= \langle \nabla_{x_t}(-\theta(x_t - \mu)) \rangle_{q_t} && \text{(D.6)} \\ &= \langle -\theta \nabla_{x_t} x_t \rangle_{q_t} && \text{(D.7)} \\ &= \langle -\theta \rangle_{q_t} && \text{(D.8)} \\ &= -\theta && \text{(D.9)} \end{aligned}$$

Energy from the SDE

The first expression is the energy term associated with the stochastic differential equation (D.1). In what follows the initial expression will correspond to the general multivariate case as shown in Chapter 4 and derived in more detail in Appendix A. Later on the general expressions are substituted by the model equations of the system studied.

[1] This is a shorthand notation for $\mathcal{N}(x_t|m_t, s_t)$.

$$
\begin{aligned}
E_{sde}(t) &= \frac{1}{2}\left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \boldsymbol{\Sigma}^{-1} (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t)) \right\rangle_{q_t} && \text{(D.10)}\\
&= \frac{1}{2}\boldsymbol{\Sigma}^{-1} \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \right\rangle_{q_t} && \text{(D.11)}\\
&= \frac{1}{2}\sigma^{-2}\left\langle \Big(-\theta(x_t-\mu) - (-a_t x_t + b_t)\Big)^2 \right\rangle_{q_t} && \text{(D.12)}\\
&= \frac{1}{2}\sigma^{-2}\left\langle \Big(-\theta x_t + \theta\mu + a_t x_t - b_t\Big)^2 \right\rangle_{q_t} && \text{(D.13)}\\
&= \frac{1}{2}\sigma^{-2}\Big\langle \theta^2 x_t^2 - \theta^2\mu x_t - \theta a_t x_t^2 + \theta b_t x_t - \theta^2\mu x_t + \theta^2\mu^2 + \theta\mu a_t x_t - \theta\mu b_t \\
&\quad - \theta a_t x_t^2 + \theta\mu a_t x_t + a_t^2 x_t^2 - a_t b_t x_t + \theta b_t x_t - \theta\mu b_t - a_t b_t x_t + b_t^2 \Big\rangle_{q_t} && \text{(D.14)}\\
&= \frac{1}{2}\sigma^{-2}\Big(\left\langle x_t^2(\theta^2 - 2\theta a_t + a_t^2)\right\rangle_{q_t} + 2\left\langle x_t(\theta b_t - \theta^2\mu + \theta\mu a_t - a_t b_t)\right\rangle_{q_t} \\
&\quad + \theta^2\mu^2 - 2\theta\mu b_t + b_t^2\Big) && \text{(D.15)}\\
&= \frac{1}{2}\sigma^{-2}\Big(\left\langle x_t^2\right\rangle_{q_t}(\theta - a_t)^2 + 2\left\langle x_t\right\rangle_{q_t}(\theta b_t - \theta^2\mu + \theta\mu a_t - a_t b_t) \\
&\quad + \theta^2\mu^2 - 2\theta\mu b_t + b_t^2\Big) && \text{(D.16)}
\end{aligned}
$$

Step (D.11) is possible because the system is univariate and the noise coefficient does not depend on the state vector $\mathbf{x}_t$. Finally, $\langle x_t\rangle_{q_t}$ and $\langle x_t^2\rangle_{q_t}$ are Gaussian moments; their expressions can be found in Appendix F.

Gradients with respect to the marginal means and variances

The next section involves the derivation of the gradients of E_{sde} with respect to the first two marginal moments. First is shown the derivative with respect to the marginal mean m_t and in a similar way the derivative with respect to the marginal variance s_t follows.

Gradient of E_{sde} w.r.t. m_t:

$$
\begin{aligned}
\nabla_{m_t} E_{sde}(t) &= \nabla_{m_t}\Bigg[\frac{1}{2}\sigma^{-2}\Big(\left\langle x_t^2\right\rangle_{q_t}(\theta - a_t)^2 + 2\left\langle x_t\right\rangle_{q_t}(\theta b_t - \theta^2\mu + \theta\mu a_t - a_t b_t) + \theta^2\mu^2 \\
&\quad - 2\theta\mu b_t + b_t^2\Big)\Bigg] && \text{(D.17)}\\
&= \frac{1}{2}\sigma^{-2}\Big(\nabla_{m_t}\left\langle x_t^2\right\rangle_{q_t}(\theta - a_t)^2 + 2\nabla_{m_t}\left\langle x_t\right\rangle_{q_t}(\theta b_t - \theta^2\mu + \theta\mu a_t - a_t b_t)\Big) && \text{(D.18)}\\
&= \frac{1}{2}\sigma^{-2}\Big(2m_t(\theta - a_t)^2 + 2(\theta b_t - \theta^2\mu + \theta\mu a_t - a_t b_t)\Big) && \text{(D.19)}\\
&= \sigma^{-2}\Big(m_t(\theta - a_t)^2 + \theta(b_t - \theta\mu + \mu a_t) - a_t b_t\Big) && \text{(D.20)}
\end{aligned}
$$

Gradient of E_{sde} w.r.t. s_t:

$$\nabla_{s_t} E_{sde}(t) = \nabla_{s_t}\Bigg[\frac{1}{2}\sigma^{-2}\Bigg(\left\langle x_t^2\right\rangle_{q_t}(\theta - a_t)^2 + 2\left\langle x_t\right\rangle_{q_t}(\theta b_t - \theta^2\mu + \theta\mu a_t - a_t b_t) + \theta^2\mu^2 - 2\theta\mu b_t + b_t^2\Bigg)\Bigg] \tag{D.21}$$

$$= \frac{1}{2}\sigma^{-2}\nabla_{s_t}\left\langle x_t^2\right\rangle_{q_t}(\theta - a_t)^2 \tag{D.22}$$

$$= \frac{1}{2}\sigma^{-2}(\theta - a_t)^2 \tag{D.23}$$

where the derivatives of the Gaussian moments with respect to m_t and s_t (i.e. $\nabla_{m_t}\left\langle x_t\right\rangle_{q_t}$, $\nabla_{m_t}\left\langle x_t^2\right\rangle_{q_t}$ and $\nabla_{s_t}\left\langle x_t^2\right\rangle_{q_t}$), are provided in Appendix F.

Gradients with respect to the variational parameters

Appendix A, showed how the derivatives of the general expression of the E_{sde} with respect the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$ can be derived. Following that example, here we compute the same expressions for the OU system.

Gradient of E_{sde} w.r.t. the offset parameter b_t:

$$\nabla_{b_t} E_{sde}(t) = -\sigma^{-2}\Big(\left\langle \mathbf{f}(x_t)\right\rangle_{q_t} + a_t\left\langle x_t\right\rangle_{q_t} - b_t\Big) \tag{D.24}$$

$$= -\sigma^{-2}\Big(\theta(\mu - m_t) + a_t\left\langle x_t\right\rangle_{q_t} - b_t\Big) \tag{D.25}$$

using Equation (D.5).

Gradient of E_{sde} w.r.t. the linear parameter a_t:

$$\nabla_{a_t} E_{sde}(t) = \sigma^{-2}\Big(\left\langle \nabla_{x_t}\mathbf{f}(x_t)\right\rangle_{q_t} + a_t\Big)s_t - \nabla_{b_t}E_{sde}(t)m_t$$

$$= \sigma^{-2}(-\theta + a_t)s_t - \nabla_{b_t}E_{sde}(t)m_t \tag{D.26}$$

using Equation (D.9).

Gradients with respect to the (hyper-) parameters

For the estimation of the (hyper-) parameters, in a gradient based optimisation routine, as suggested in Chapter 4, the gradients of the energy term have to be estimated. For the OU system are given as follows:

Gradient of E_{sde} w.r.t. the drift parameters $\boldsymbol{\theta}$: The general expression of the OU drift function includes two parameters $\boldsymbol{\theta} = [\theta, \mu]^\top$. The gradient of E_{sde} with respect to this vector is:

$$\nabla_{\boldsymbol{\theta}} E_{sde} = \left[\frac{\partial E_{sde}}{\partial\theta}, \frac{\partial E_{sde}}{\partial\mu}\right]^\top$$

The partial derivatives are computed separately as follows:

$$\begin{aligned}\frac{\partial E_{sde}}{\partial\theta} &= \frac{\partial}{\partial\theta}\left[\frac{1}{2}\sigma^{-2}\left(\left\langle x_t^2\right\rangle_{q_t}(\theta-a_t)^2+2\left\langle x_t\right\rangle_{q_t}(\theta b_t-\theta^2\mu+\theta\mu a_t-a_t b_t)+\theta^2\mu^2\right.\right.\\ &\quad\left.\left.-2\theta\mu b_t+b_t^2\right)\right] &\text{(D.27)}\\ &= \frac{1}{2}\sigma^{-2}\left(\left\langle x_t^2\right\rangle_{q_t}\frac{\partial}{\partial\theta}(\theta-a_t)^2+2\left\langle x_t\right\rangle_{q_t}\frac{\partial}{\partial\theta}(\theta b_t-\theta^2\mu+\theta\mu a_t-a_t b_t)+\frac{\partial}{\partial\theta}\theta^2\mu^2\right.\\ &\quad\left.-2\frac{\partial}{\partial\theta}\theta\mu b_t\right) &\text{(D.28)}\\ &= \frac{1}{2}\sigma^{-2}\left(2\left\langle x_t^2\right\rangle_{q_t}(\theta-a_t)+2\left\langle x_t\right\rangle_{q_t}(b_t-2\theta\mu+\mu a_t)+2\theta\mu^2-2\mu b_t\right) &\text{(D.29)}\\ &= \sigma^{-2}\left(\left\langle x_t^2\right\rangle_{q_t}(\theta-a_t)+\left\langle x_t\right\rangle_{q_t}(b_t-2\theta\mu+\mu a_t)+\theta\mu^2-\mu b_t\right) &\text{(D.30)}\end{aligned}$$

$$\begin{aligned}\frac{\partial E_{sde}}{\partial\mu} &= \frac{\partial}{\partial\mu}\left[\frac{1}{2}\sigma^{-2}\left(\left\langle x_t^2\right\rangle_{q_t}(\theta-a_t)^2+2\left\langle x_t\right\rangle_{q_t}(\theta b_t-\theta^2\mu+\theta\mu a_t-a_t b_t)+\theta^2\mu^2\right.\right.\\ &\quad\left.\left.-2\theta\mu b_t+b_t^2\right)\right] &\text{(D.31)}\\ &= \frac{1}{2}\sigma^{-2}\left(2\left\langle x_t\right\rangle_{q_t}\frac{\partial}{\partial\mu}(\theta b_t-\theta^2\mu+\theta\mu a_t-a_t b_t)+\frac{\partial}{\partial\mu}\theta^2\mu^2-2\frac{\partial}{\partial\mu}\theta\mu b_t\right) &\text{(D.32)}\\ &= \frac{1}{2}\sigma^{-2}\left(2\left\langle x_t\right\rangle_{q_t}(\theta a_t-\theta^2)+2\theta^2\mu-2\theta b_t\right) &\text{(D.33)}\\ &= \theta\sigma^{-2}\left(m_t(a_t-\theta)+\theta\mu-b_t\right) &\text{(D.34)}\end{aligned}$$

Gradient of E_{sde} w.r.t. the noise parameter σ^2: Following a similar derivation to above the derivative of the energy with respect to the system noise coefficient is given by:

$$\begin{aligned}\frac{\partial E_{sde}}{\partial\sigma^2} &= \frac{\partial}{\partial\sigma^2}\left[\frac{1}{2}\sigma^{-2}\left(\left\langle x_t^2\right\rangle_{q_t}(\theta-a_t)^2+2\left\langle x_t\right\rangle_{q_t}(\theta b_t-\theta^2\mu+\theta\mu a_t-a_t b_t)+\theta^2\mu^2\right.\right.\\ &\quad\left.\left.-2\theta\mu b_t+b_t^2\right)\right] &\text{(D.35)}\\ &= -\frac{1}{2}\sigma^{-4}\left(\left\langle x_t^2\right\rangle_{q_t}(\theta-a_t)^2+2\left\langle x_t\right\rangle_{q_t}(\theta b_t-\theta^2\mu+\theta\mu a_t-a_t b_t)+\theta^2\mu^2\right.\\ &\quad\left.-2\theta\mu b_t+b_t^2\right) &\text{(D.36)}\end{aligned}$$

Optimal initial values of the variational parameters

In order to initialize the optimisation routine (smoothing algorithm), as suggested in Chapter 4,an initial guess has to be made for the all the *discretized* variational parameters $\mathbf{a}(k)$ and $\mathbf{b}(k)$. As shown in Archambeau et al. (2008), these can be expressed as functions of the Lagrange multipliers $\boldsymbol{\psi}(k)$, $\boldsymbol{\lambda}(k)$ as well as the means $\mathbf{m}(k)$ as shown below:

$$\mathbf{a}(k) = -\left\langle \nabla_{\mathbf{x}_t} \mathbf{f}(\mathbf{x}_t) \right\rangle_{q_t} + 2\sigma^2 \boldsymbol{\psi}(k) \tag{D.37}$$

$$\mathbf{b}(k) = \left\langle \mathbf{f}(\mathbf{x}_t) \right\rangle_{q_t} + \mathbf{a}(k) * \mathbf{m}(k) - \sigma^2 \boldsymbol{\lambda}(k) \tag{D.38}$$

where $\mathbf{a}(k)$, $\mathbf{b}(k)$, $\mathbf{m}(k)$, $\boldsymbol{\psi}(k)$ and $\boldsymbol{\lambda}(k)$ are now vectors containing all the discrete time variables a_t, b_t, m_t, ψ_t and λ_t, k is the index of the optimisation loop (i.e. k indicates algorithmic time rather than discretisation time) and the symbol '$*$' indicates element-wise multiplication between two vectors of the same size. Using Equations (D.5) and (D.9) these expressions can be further expanded. The linear parameter becomes:

$$\begin{aligned} \mathbf{a}(k) &= -\left\langle \nabla_{\mathbf{x}_t} \mathbf{f}(\mathbf{x}_t) \right\rangle_{q_t} + 2\sigma^2 \boldsymbol{\psi}(k) && \text{(D.39)} \\ &= \theta + 2\sigma^2 \boldsymbol{\psi}(k) && \text{(D.40)} \end{aligned}$$

and the bias parameter:

$$\begin{aligned} \mathbf{b}(k) &= \left\langle \mathbf{f}(\mathbf{x}_t) \right\rangle_{q_t} + \mathbf{a}(k) * \mathbf{m}(k) - \sigma^2 \boldsymbol{\lambda}(k) && \text{(D.41)} \\ &= \theta(\mu - \mathbf{m}(k)) + (\theta + 2\sigma^2 \boldsymbol{\psi}(k)) * \mathbf{m}(k) - \sigma^2 \boldsymbol{\lambda}(k) && \text{(D.42)} \\ &= \theta\mu - \theta\mathbf{m}(k) + \theta\mathbf{m}(k) + 2\sigma^2 \boldsymbol{\psi}(k) * \mathbf{m}(k) - \sigma^2 \boldsymbol{\lambda}(k) && \text{(D.43)} \\ &= \theta\mu + \sigma^2 (2\boldsymbol{\psi}(k) * \mathbf{m}(k) - \boldsymbol{\lambda}(k)) && \text{(D.44)} \end{aligned}$$

However for $k = 0$ (i.e. the beginning of the optimisation process), the Lagrange multipliers are set to zero (i.e. $\boldsymbol{\psi}(0) = 0$ and $\boldsymbol{\lambda}(0) = 0$). Hence the above expressions for the initial iteration of the algorithm are simplified to:

$$\mathbf{a}(0) = \theta \quad \text{and} \quad \mathbf{b}(0) = \theta\mu \tag{D.45}$$

D.2 Double Well (DW) system equations

The second system for which the expressions of the variational framework are derived analytically, is the univariate *double well* (DW). This non-linear stochastic process is governed by the following SDE:

$$dx_t = 4x_t(\theta - x_t^2)dt + \sigma^2\, dw_t \tag{D.46}$$

with drift parameter $\theta > 0$, indicating the system's stable states.

The analytic derivations of this system is follows a similar approach to that presented for the OU process. First the averaged drift function is computed with respect to the Gaussian distribution

q_t, as shown in Equation (D.49), followed by the averaged gradient of the drift with respect to x_t (Eq. D.53).

$$\begin{aligned} \langle \mathbf{f}(x_t) \rangle_{q_t} &= \left\langle 4x_t(\theta - x_t^2) \right\rangle_{q_t} && \text{(D.47)} \\ &= \left\langle 4\theta x_t - 4x_t^3 \right\rangle_{q_t} && \text{(D.48)} \\ &= 4\theta \langle x_t \rangle_{q_t} - 4 \left\langle x_t^3 \right\rangle_{q_t} && \text{(D.49)} \end{aligned}$$

$$\begin{aligned} \langle \nabla_{x_t} \mathbf{f}(x_t) \rangle_{q_t} &= \left\langle \nabla_{x_t}(4x_t(\theta - x_t^2)) \right\rangle_{q_t} && \text{(D.50)} \\ &= \left\langle \nabla_{x_t}(4\theta X_t - 4X_t^3) \right\rangle_{q_t} && \text{(D.51)} \\ &= \left\langle 4\theta - 12x_t^2 \right\rangle_{q_t} && \text{(D.52)} \\ &= 4\theta - 12 \left\langle x_t^2 \right\rangle_{q_t} && \text{(D.53)} \end{aligned}$$

Energy from the SDE

The energy term related to the stochastic differential equation (D.46), is:

$$\begin{aligned} E_{sde}(t) &= \frac{1}{2} \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1} (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t)) \right\rangle_{q_t} && \text{(D.54)} \\ &= \frac{1}{2} \mathbf{\Sigma}^{-1} \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \right\rangle_{q_t} && \text{(D.55)} \\ &= \frac{1}{2} \sigma^{-2} \left\langle \left(4x_t(\theta - x_t^2) - (-a_t x_t + b_t) \right)^2 \right\rangle_{q_t} && \text{(D.56)} \\ &= \frac{1}{2} \sigma^{-2} \left\langle \left(4\theta x_t - 4x_t^3 + a_t x_t - b_t \right)^2 \right\rangle_{q_t} && \text{(D.57)} \\ &= \frac{1}{2} \sigma^{-2} \left\langle \left((4\theta + a_t) x_t - 4x_t^3 - b_t \right)^2 \right\rangle_{q_t} && \text{(D.58)} \\ &= \frac{1}{2} \sigma^{-2} \left\langle \left(c_t x_t - 4x_t^3 - b_t \right)^2 \right\rangle_{q_t} && \text{(D.59)} \\ &= \frac{1}{2} \sigma^{-2} \left\langle c_t^2 x_t^2 - 4c_t x_t^4 - b_t c_t x_t - 4c_t x_t^4 + 16x_t^6 + 4b_t x_t^3 - b_t c_t x_t + 4b_t x_t^3 + b_t^2 \right\rangle_{q_t} && \text{(D.60)} \\ &= \frac{1}{2} \sigma^{-2} \left\langle c_t^2 x_t^2 - 8c_t x_t^4 - 2b_t c_t x_t + 16x_t^6 + 8b_t x_t^3 + b_t^2 \right\rangle_{q_t} && \text{(D.61)} \\ &= \frac{1}{2} \sigma^{-2} \left(c_t^2 \left\langle x_t^2 \right\rangle_{q_t} - 8c_t \left\langle x_t^4 \right\rangle_{q_t} - 2b_t c_t \langle x_t \rangle_{q_t} + 16 \left\langle x_t^6 \right\rangle_{q_t} + 8b_t \left\langle x_t^3 \right\rangle_{q_t} + b_t^2 \right) && \text{(D.62)} \end{aligned}$$

where in step (D.59) is introduced, for simplicity of the presentation, the variable $c_t = (4\theta + a_t)$ and all the higher order Gaussian moments $\langle x_t \rangle_{q_t}$, $\left\langle x_t^2 \right\rangle_{q_t}$, $\left\langle x_t^3 \right\rangle_{q_t}$, $\left\langle x_t^4 \right\rangle_{q_t}$ and $\left\langle x_t^6 \right\rangle_{q_t}$, are given in Appendix F.

Gradients with respect to the marginal means and variances

This section presents the derivations of the gradients of E_{sde} with respect to the marginal means and variances. The derivative with respect to the marginal mean m_t is computed first, followed by the derivative with respect to the marginal variance s_t.

Gradient of E_{sde} w.r.t. m_t:

$$\nabla_{m_t} E_{sde}(t) = \nabla_{m_t}\left[\frac{1}{2}\sigma^{-2}\left(c_t^2\left\langle x_t^2\right\rangle_{q_t} - 8c_t\left\langle x_t^4\right\rangle_{q_t} - 2b_tc_t\left\langle x_t\right\rangle_{q_t} + 16\left\langle x_t^6\right\rangle_{q_t} + 8b_t\left\langle x_t^3\right\rangle_{q_t} + b_t^2\right)\right] \tag{D.63}$$

$$= \frac{1}{2}\sigma^{-2}\left(c_t^2\nabla_{m_t}\left\langle x_t^2\right\rangle_{q_t} - 8c_t\nabla_{m_t}\left\langle x_t^4\right\rangle_{q_t} - 2b_tc_t + 16\nabla_{m_t}\left\langle x_t^6\right\rangle_{q_t} + 8b_t\nabla_{m_t}\left\langle x_t^3\right\rangle_{q_t}\right) \tag{D.64}$$

Gradient of E_{sde} w.r.t. s_t:

$$\nabla_{s_t} E_{sde}(t) = \nabla_{s_t}\left[\frac{1}{2}\sigma^{-2}\left(c_t^2\left\langle x_t^2\right\rangle_{q_t} - 8c_t\left\langle x_t^4\right\rangle_{q_t} - 2b_tc_t\left\langle x_t\right\rangle_{q_t} + 16\left\langle x_t^6\right\rangle_{q_t} + 8b_t\left\langle x_t^3\right\rangle_{q_t} + b_t^2\right)\right] \tag{D.65}$$

$$= \frac{1}{2}\sigma^{-2}\left(c_t^2\nabla_{s_t}\left\langle x_t^2\right\rangle_{q_t} - 8c_t\nabla_{s_t}\left\langle x_t^4\right\rangle_{q_t} + 16\nabla_{s_t}\left\langle x_t^6\right\rangle_{q_t} + 8b_t\nabla_{s_t}\left\langle x_t^3\right\rangle_{q_t}\right) \tag{D.66}$$

with all the derivatives of the higher order Gaussian moments with respect to m_t and s_t, such as $\nabla_{m_t}\left\langle x_t^2\right\rangle_{q_t}$, $\nabla_{m_t}\left\langle x_t^3\right\rangle_{q_t}$, $\nabla_{m_t}\left\langle x_t^4\right\rangle_{q_t}$, $\nabla_{m_t}\left\langle x_t^6\right\rangle_{q_t}$, $\nabla_{s_t}\left\langle x_t^2\right\rangle_{q_t}$, $\nabla_{s_t}\left\langle x_t^3\right\rangle_{q_t}$, $\nabla_{s_t}\left\langle x_t^4\right\rangle_{q_t}$ and $\nabla_{s_t}\left\langle x_t^6\right\rangle_{q_t}$, again provided in Appendix F.

Gradients with respect to the variational parameters

Following a similar methodology, as shown for the OU system, the gradients of E_{sde} w.r.t. the variational parameters a_t and b_t for the DW system, are derived using Equations (D.49) and (D.53).

Gradient of E_{sde} w.r.t. the offset parameter b_t:

$$\nabla_{b_t} E_{sde}(t) = -\sigma^{-2}\left(\left\langle \mathbf{f}(x_t)\right\rangle_{q_t} + a_t\left\langle x_t\right\rangle_{q_t} - b_t\right) \tag{D.67}$$

$$= -\sigma^{-2}\left(4\theta\left\langle x_t\right\rangle_{q_t} - 4\left\langle x_t^3\right\rangle_{q_t} + a_t\left\langle x_t\right\rangle_{q_t} - b_t\right) \tag{D.68}$$

$$= -\sigma^{-2}\left((4\theta + a_t)\left\langle x_t\right\rangle_{q_t} - 4\left\langle x_t^3\right\rangle_{q_t} - b_t\right) \tag{D.69}$$

Gradient of E_{sde} w.r.t. the linear coefficient a_t:

$$\nabla_{a_t} E_{sde}(t) = \sigma^{-2}\left(\left\langle \nabla_{x_t}\mathbf{f}(x_t)\right\rangle_{q_t} + a_t\right)s_t - \nabla_{b_t}E_{sde}(t)m_t \tag{D.70}$$

$$= \sigma^{-2}\left(4\theta - 12\left\langle x_t^2\right\rangle_{q_t} + a_t\right)s_t - \nabla_{b_t}E_{sde}(t)m_t \tag{D.71}$$

Gradients with respect to the (hyper-) parameters

The estimation of the (hyper-) parameters, for the DW system, includes the partial derivatives of E_{sde} with respect to θ and σ^2. These are given by:

Gradient of E_{sde} w.r.t. drift parameter θ: As shown earlier the drift function of the DW process (see Eq. D.46), has only one parameter. The classical approach to derive the necessary partial derivative of E_{sde} is simply to differentiate Equation (D.62) w.r.t. θ parameter. However a simpler approach is to use the general expression of this gradient as shown in Appendix A and then make the appropriate substitution for the model that is studied. That leads to the following expression:

$$\nabla_{\boldsymbol{\theta}} E_{sde}(t) = \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \boldsymbol{\Sigma}^{-1} (\nabla_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}_t)) \right\rangle_{q_t} \tag{D.72}$$

$$= \boldsymbol{\Sigma}^{-1} \left\langle \nabla_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}_t) (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \right\rangle_{q_t} \tag{D.73}$$

$$= \sigma^{-2} \left\langle 4x_t(4\theta x_t - 4x_t^3 + a_t x_t - b_t) \right\rangle_{q_t} \tag{D.74}$$

$$= \sigma^{-2} \left\langle 16\theta x_t^2 - 16x_t^4 + 4a_t x_t^2 - 4b_t x_t \right\rangle_{q_t} \tag{D.75}$$

$$= 4\sigma^{-2} \left\langle (4\theta + a_t) x_t^2 - 4x_t^4 - b_t x_t \right\rangle_{q_t} \tag{D.76}$$

$$= 4\sigma^{-2} \left((4\theta + a_t) \left\langle x_t^2 \right\rangle_{q_t} - 4 \left\langle x_t^4 \right\rangle_{q_t} - b_t \left\langle x_t \right\rangle_{q_t} \right) \tag{D.77}$$

where we have make use of the fact that the derivative of the DW drift function with respect to the drift parameter is given by:

$$\nabla_\theta \mathbf{f}(x_t) = \nabla_\theta \left(4x_t(\theta - x_t^2) \right) \tag{D.78}$$

$$= \nabla_\theta \left(4\theta x_t - 4x_t^3 \right) \tag{D.79}$$

$$= 4x_t \tag{D.80}$$

Gradient of E_{sde} w.r.t. the system noise parameter σ^2: Although the computation of the partial derivative of E_{sde} w.r.t. σ^2 is straightforward, here is followed a similar approach as above (using the general expressions of the gradient) to demonstrate the ease of computing system specific expressions from the general multivariate variational framework.

$$\nabla_{\boldsymbol{\Sigma}} E_{sde}(t) = -\frac{1}{2} \boldsymbol{\Sigma}^{-1} \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \right\rangle_{q_t} \boldsymbol{\Sigma}^{-1} \tag{D.81}$$

$$= -\frac{1}{2} \sigma^{-2} \left\langle \left((4\theta + a_t) x_t - 4x_t^3 - b_t \right)^2 \right\rangle_{q_t} \sigma^{-2} \tag{D.82}$$

$$= -\frac{1}{2} \sigma^{-4} \left\langle c_t^2 x_t^2 - 4c_t x_t^4 - b_t c_t x_t - 4c_t x_t^4 + 16x_t^6 + 4b_t x_t^3 - b_t c_t x_t + 4b_t x_t^3 + b_t^2 \right\rangle_{q_t} \tag{D.83}$$

$$= -\frac{1}{2} \sigma^{-4} \left\langle c_t^2 x_t^2 - 8c_t x_t^4 - 2b_t c_t x_t + 16x_t^6 + 8b_t x_t^3 + b_t^2 \right\rangle_{q_t} \tag{D.84}$$

$$= -\frac{1}{2} \sigma^{-4} \left(c_t^2 \left\langle x_t^2 \right\rangle_{q_t} - 8c_t \left\langle x_t^4 \right\rangle_{q_t} - 2b_t c_t \left\langle x_t \right\rangle_{q_t} + 16 \left\langle x_t^6 \right\rangle_{q_t} + 8b_t \left\langle x_t^3 \right\rangle_{q_t} + b_t^2 \right) \tag{D.85}$$

where the variable $c_t = 4\theta + a_t$, is used to simplify the expressions.

Optimal initial values of the variational parameters

In a similar fashion to the OU system, the initial "*guesses*" for the all the *discretized* variational parameters $\mathbf{a}(k)$ and $\mathbf{b}(k)$ are given by Equations D.38, after having substituted the Equations (D.49) and (D.53) for the DW system.

Initial linear parameter:

$$\mathbf{a}(k) = -\left\langle \nabla_{\mathbf{x}_t} \mathbf{f}(\mathbf{x}_t) \right\rangle_{q_t} + 2\sigma^2 \boldsymbol{\psi}(k) \quad \text{(D.86)}$$

$$= -4\theta + 12\left\langle x_t^2 \right\rangle_{q_t} + 2\sigma^2 \boldsymbol{\psi}(k) \quad \text{(D.87)}$$

$$= -4\theta + 12\left(\mathbf{m}(k)^2 + \mathbf{s}(k)\right) + 2\sigma^2 \boldsymbol{\psi}(k) \quad \text{(D.88)}$$

and since for $k = 0$ the Lagrange multiplier $\boldsymbol{\psi}(0) = 0$, the expression becomes:

$$\mathbf{a}(0) = -4\theta + 12\left(\mathbf{m}(0)^2 + \mathbf{s}(0)\right) \quad \text{(D.89)}$$

Initial bias parameter:

$$\mathbf{b}(k) = \left\langle \mathbf{f}(\mathbf{x}_t) \right\rangle_{q_t} + \mathbf{a}(k) * \mathbf{m}(k) - \sigma^2 \boldsymbol{\lambda}(k) \quad \text{(D.90)}$$

$$= 4\theta \left\langle x_t \right\rangle_{q_t} - 4\left\langle x_t^3 \right\rangle_{q_t} + \left(-4\theta + 12\left(\mathbf{m}(k)^2 + \mathbf{s}(k)\right) + 2\sigma^2 \boldsymbol{\psi}(k)\right) * \mathbf{m}(k) - \sigma^2 \boldsymbol{\lambda}(k) \quad \text{(D.91)}$$

$$= 4\theta\mathbf{m}(k) - 4\left(\mathbf{m}(k)^3 + 3\mathbf{m}(k) * \mathbf{s}(k)\right) - 4\theta\mathbf{m}(k) + 12\mathbf{m}(k)^3 + 12\mathbf{m}(k) * \mathbf{s}(k)$$

$$+ 2\sigma^2 \boldsymbol{\psi}(k) * \mathbf{m}(k) - \sigma^2 \boldsymbol{\lambda}(k) \quad \text{(D.92)}$$

$$= 4\theta\mathbf{m}(k) - 4\mathbf{m}(k)^3 - 12\mathbf{m}(k) * \mathbf{s}(k) - 4\theta\mathbf{m}(k) + 12\mathbf{m}(k)^3 + 12\mathbf{m}(k) * \mathbf{s}(k)$$

$$+ 2\sigma^2 \boldsymbol{\psi}(k) * \mathbf{m}(k) - \sigma^2 \boldsymbol{\lambda}(k) \quad \text{(D.93)}$$

$$= 8\mathbf{m}(k)^3 + 2\sigma^2 \boldsymbol{\psi}(k) * \mathbf{m}(k) - \sigma^2 \boldsymbol{\lambda}(k) \quad \text{(D.94)}$$

$$\text{(D.95)}$$

where the higher order expectations $\left\langle x_t^2 \right\rangle_{q_t}$ and $\left\langle x_t^3 \right\rangle_{q_t}$, have been expanded to make the derivations more complete. Also the symbol "*" operates element-wise multiplication, between two vectors of the same size. Since for $k = 0$ the Lagrange multipliers $\boldsymbol{\psi}(0) = 0$ and $\boldsymbol{\lambda}(0)$, the expression becomes:

$$\mathbf{b}(0) = 8\mathbf{m}(0)^3 \quad \text{(D.96)}$$

Here an obvious obstacle arises since to initialize all the discrete parameters $\mathbf{a}(0)$ and $\mathbf{b}(0)$, one needs an initial guess for all the marginal means and variances (i.e. $\mathbf{m}(0)$ and $\mathbf{s}(0)$). To overcome this problem a simple solution is to use a Gaussian process regression "smoother" (Rasmussen and Williams, 2006), on the observations, to get an initial estimate of the marginal means and variances and then use these values to initialize the variational parameters.

D.3 Lorenz ’63 (L3D) system equations

The final system for which the required equations were computed analytically is the stochastic version of the three dimensional chaotic Lorenz ’63 (L3D). Because these derivations demand many computations an effort is made to emphasize more the way the equations are derived, rather than the detailed individual steps. Initially, the systems’ equations will be provided in detail and subsequently they will be decomposed to several parts that will help the readability of the presentation.

The time evolution of the L3D system is described by the following stochastic differential equation:

$$d\mathbf{x}_t = \begin{bmatrix} \sigma(y_t - x_t) \\ \rho x_t - y_t - x_t z_t \\ x_t y_t - \beta z_t \end{bmatrix} dt + \mathbf{\Sigma}^{1/2}\, d\mathbf{w}_t \;, \tag{D.97}$$

where $\mathbf{x}_t = [x_t\; y_t\; z_t]^\top \in \Re^3$ in the state vector representing all three dimensions, $\boldsymbol{\theta} = [\sigma\; \rho\; \beta]^\top \in \Re^3$, is the drift parameter vector, $\mathbf{\Sigma} \in \Re^{3\times 3}$ is a diagonal covariance matrix and $\mathbf{w}_t \in \Re^3$.

Energy from the SDE

Initially we recall the expression that gives the energy term related to the SDE (see Appendix A), that is:

$$E_{sde}(t) = \frac{1}{2}\left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1} (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t)) \right\rangle_{q_t} \tag{D.98}$$

The drift function $\mathbf{f}(\mathbf{x}_t)$ of this system is given by:

$$\mathbf{f}(\mathbf{x}_t) = \begin{bmatrix} \sigma(y_t - x_t) \\ \rho x_t - y_t - x_t z_t \\ x_t y_t - \beta z_t \end{bmatrix} \;, \tag{D.99}$$

while the linear approximation $\mathbf{g}_L(\mathbf{x}_t)$ is given by:

$$\mathbf{g}_L(\mathbf{x}_t) = -\mathbf{A}_t \mathbf{x}_t + \mathbf{b}_t \tag{D.100}$$

$$= -\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}_t \times \begin{bmatrix} x \\ y \\ z \end{bmatrix}_t + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}_t \tag{D.101}$$

$$= -\begin{bmatrix} A_{11}x + A_{12}y + A_{13}z - b_1 \\ A_{21}x + A_{22}y + A_{23}z - b_2 \\ A_{31}x + A_{32}y + A_{33}z - b_3 \end{bmatrix}_t \;, \tag{D.102}$$

where the (continuous) dependency on time 't' is denoted by the subscript on the matrix and vectors. Combining Equations (D.99) and (D.102) we get the necessary vector $\mathbf{v}_t = (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))$, as shown in Eq.(D.104).

$$\mathbf{v}_t = \begin{bmatrix} \sigma(y-x) \\ \rho x - y - xz \\ xy - \beta z \end{bmatrix}_t + \begin{bmatrix} A_{11}x + A_{12}y + A_{13}z - b_1 \\ A_{21}x + A_{22}y + A_{23}z - b_2 \\ A_{31}x + A_{32}y + A_{33}z - b_3 \end{bmatrix}_t \tag{D.103}$$

$$= \begin{bmatrix} \sigma y - \sigma x + A_{11}x + A_{12}y + A_{13}z - b_1 \\ \rho x - y - xz + A_{21}x + A_{22}y + A_{23}z - b_2 \\ xy - \beta z + A_{31}x + A_{32}y + A_{33}z - b_3 \end{bmatrix}_t \tag{D.104}$$

At this point the initial energy expression Eq.(D.98) becomes:

$$E_{sde}(t) = \frac{1}{2}\left\langle \mathbf{v}_t^\top \mathbf{\Sigma}^{-1} \mathbf{v}_t \right\rangle_{q_t} \tag{D.105}$$

$$= \frac{1}{2}\left\langle \mathbf{v}_t^\top \begin{bmatrix} \Sigma_x & 0 & 0 \\ 0 & \Sigma_y & 0 \\ 0 & 0 & \Sigma_z \end{bmatrix}^{-1} \mathbf{v}_t \right\rangle_{q_t} \tag{D.106}$$

$$= \frac{1}{2}\left[\Sigma_x^{-1}\,\Sigma_y^{-1}\,\Sigma_z^{-1}\right]\left\langle \mathbf{v}_t^2 \right\rangle_{q_t} \tag{D.107}$$

$$= \frac{1}{2}\left[\Sigma_x^{-1}\,\Sigma_y^{-1}\,\Sigma_z^{-1}\right] \times \left\langle \begin{bmatrix} (\sigma y - \sigma x + A_{11}x + A_{12}y + A_{13}z - b_1)^2 \\ (\rho x - y - xz + A_{21}x + A_{22}y + A_{23}z - b_2)^2 \\ (xy - \beta z + A_{31}x + A_{32}y + A_{33}z - b_3)^2 \end{bmatrix}_t \right\rangle_{q_t} \tag{D.108}$$

$$= \frac{1}{2}\left[\Sigma_x^{-1}\,\Sigma_y^{-1}\,\Sigma_z^{-1}\right] \times \begin{bmatrix} \left\langle(\sigma y - \sigma x + A_{11}x + A_{12}y + A_{13}z - b_1)^2\right\rangle_{q_t} \\ \left\langle(\rho x - y - xz + A_{21}x + A_{22}y + A_{23}z - b_2)^2\right\rangle_{q_t} \\ \left\langle(xy - \beta z + A_{31}x + A_{32}y + A_{33}z - b_3)^2\right\rangle_{q_t} \end{bmatrix}_t, \tag{D.109}$$

where Σ_x, Σ_y and Σ_z represent the noise on each separate dimension of the system and the square of the vector $\mathbf{v}_t$, as appears in step (D.107), operates element-wise square, as shown later in step (D.108).

The next step is to expand the squares in the above expression and compute the required expectations. This requires some straightforward but tedious calculations, which result in some rather long mathematical expressions. Here such presentation is avoided and the resulting expression of the E_{sde} is given more compactly by:

$$\begin{aligned} E_{sde}(t) = \frac{1}{2}\Big(& \Sigma_x^{-1}\left\langle(\sigma y - \sigma x + A_{11}x + A_{12}y + A_{13}z - b_1)^2\right\rangle_{q_t} \\ & + \Sigma_y^{-1}\left\langle(\rho x - y - xz + A_{21}x + A_{22}y + A_{23}z - b_2)^2\right\rangle_{q_t} \\ & + \Sigma_z^{-1}\left\langle(xy - \beta z + A_{31}x + A_{32}y + A_{33}z - b_3)^2\right\rangle_{q_t}\Big), \end{aligned} \tag{D.110}$$

where the dependency on time 't' has been omitted of notational convenience.

Gradients with respect to the marginal means and variances

Once the complete expression of E_{sde} has been expanded then the gradients with respect to $\mathbf{m}_t$ and $\mathbf{S}_t$ can be computed.

Gradient of E_{sde} w.r.t. the marginal mean vector $\mathbf{m}_t$: Since the marginal mean vector $\mathbf{m}_t$ consists of three variables (i.e. $\mathbf{m}_t = [m_x\; m_y\; m_z]_t^\top$), in order to compute $\nabla_{\mathbf{m}_t} E_{sde}(t)$, one has to compute the partial derivatives of $E_{sde}(t)$ with respect to m_{xt}, m_{y_t} and m_{z_t} as shown below.

$$\nabla_{\mathbf{m}_t} E_{sde}(t) = \begin{bmatrix} \frac{\partial E_{sde}}{\partial m_x} \\ \frac{\partial E_{sde}}{\partial m_y} \\ \frac{\partial E_{sde}}{\partial m_z} \end{bmatrix}_t , \tag{D.111}$$

where the sub-script 't' indicates time. The partial derivatives are computed as follows:

- Partial derivative of $E_{sde}(t)$ with respect to m_{xt}:

$$\begin{aligned}\frac{\partial E_{sde}(t)}{\partial m_{xt}} &= \frac{1}{2}\frac{\partial}{\partial m_{xt}}\Big(\Sigma_x^{-1}\left\langle(\sigma y - \sigma x + A_{11}x + A_{12}y + A_{13}z - b_1)^2\right\rangle_{q_t} \\ &+ \Sigma_y^{-1}\left\langle(\rho x - y - xz + A_{21}x + A_{22}y + A_{23}z - b_2)^2\right\rangle_{q_t} \\ &+ \Sigma_z^{-1}\left\langle(xy - \beta z + A_{31}x + A_{32}y + A_{33}z - b_3)^2\right\rangle_{q_t}\Big) \qquad (D.112) \\ &= \frac{1}{2}\Big(\Sigma_x^{-1}\frac{\partial}{\partial m_{xt}}\left\langle(\sigma y - \sigma x + A_{11}x + A_{12}y + A_{13}z - b_1)^2\right\rangle_{q_t} \\ &+ \Sigma_y^{-1}\frac{\partial}{\partial m_{xt}}\left\langle(\rho x - y - xz + A_{21}x + A_{22}y + A_{23}z - b_2)^2\right\rangle_{q_t} \\ &+ \Sigma_z^{-1}\frac{\partial}{\partial m_{xt}}\left\langle(xy - \beta z + A_{31}x + A_{32}y + A_{33}z - b_3)^2\right\rangle_{q_t}\Big) \qquad (D.113)\end{aligned}$$

- Partial derivative of $E_{sde}(t)$ with respect to m_{y_t}:

$$\begin{aligned}\frac{\partial E_{sde}(t)}{\partial m_{y_t}} &= \frac{1}{2}\frac{\partial}{\partial m_{y_t}}\Big(\Sigma_x^{-1}\left\langle(\sigma y - \sigma x + A_{11}x + A_{12}y + A_{13}z - b_1)^2\right\rangle_{q_t} \\ &+ \Sigma_y^{-1}\left\langle(\rho x - y - xz + A_{21}x + A_{22}y + A_{23}z - b_2)^2\right\rangle_{q_t} \\ &+ \Sigma_z^{-1}\left\langle(xy - \beta z + A_{31}x + A_{32}y + A_{33}z - b_3)^2\right\rangle_{q_t}\Big) \qquad (D.114) \\ &= \frac{1}{2}\Big(\Sigma_x^{-1}\frac{\partial}{\partial m_{y_t}}\left\langle(\sigma y - \sigma x + A_{11}x + A_{12}y + A_{13}z - b_1)^2\right\rangle_{q_t} \\ &+ \Sigma_y^{-1}\frac{\partial}{\partial m_{y_t}}\left\langle(\rho x - y - xz + A_{21}x + A_{22}y + A_{23}z - b_2)^2\right\rangle_{q_t} \\ &+ \Sigma_z^{-1}\frac{\partial}{\partial m_{y_t}}\left\langle(xy - \beta z + A_{31}x + A_{32}y + A_{33}z - b_3)^2\right\rangle_{q_t}\Big) \qquad (D.115)\end{aligned}$$

- Partial derivative of $E_{sde}(t)$ with respect to m_{z_t}:

$$\begin{aligned}\frac{\partial E_{sde}(t)}{\partial m_{z_t}} &= \frac{1}{2}\frac{\partial}{\partial m_{z_t}}\Big(\Sigma_x^{-1}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Big) \qquad \text{(D.116)} \\ &=\frac{1}{2}\Big(\Sigma_x^{-1}\frac{\partial}{\partial m_{z_t}}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\frac{\partial}{\partial m_{z_t}}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\frac{\partial}{\partial m_{z_t}}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Big) \qquad \text{(D.117)}\end{aligned}$$

Gradient of E_{sde} w.r.t. the covariance matrix $\mathbf{S}_t$: In a similar manner the computation of $\nabla_{\mathbf{S}_t}E_{sde}(t)$, requires the partial derivatives of $E_{sde}(t)$ with respect to S_{xx}, S_{xy}, S_{xz}, S_{yy}, S_{yz} and S_{zz}. The marginal covariance matrix $\mathbf{S}_t$, can be schematically represented as:

$$\mathbf{S}_t=\begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz}\end{bmatrix}_t, \qquad \text{(D.118)}$$

where $S_{xx}\in\Re$ represents the variance on the x dimension, $S_{xy}\in\Re$ is the co-variance between the x and y and sub-script 't' indicates time. Here the fact that matrix $\mathbf{S}_t$ is symmetric reduces the number of partial derivatives from nine to six (i.e. $\frac{\partial E_{sde}(t)}{\partial S_{xy}}=\frac{\partial E_{sde}(t)}{\partial S_{yx}}$). The gradient is given by:

$$\nabla_{\mathbf{S}_t}E_{sde}(t)=\begin{bmatrix} \frac{\partial E_{sde}}{\partial S_{xx}} & \frac{\partial E_{sde}}{\partial S_{xy}} & \frac{\partial E_{sde}}{\partial S_{xz}} \\ \frac{\partial E_{sde}}{\partial S_{yx}} & \frac{\partial E_{sde}}{\partial S_{yy}} & \frac{\partial E_{sde}}{\partial S_{yz}} \\ \frac{\partial E_{sde}}{\partial S_{zx}} & \frac{\partial E_{sde}}{\partial S_{zy}} & \frac{\partial E_{sde}}{\partial S_{zz}}\end{bmatrix}_t \qquad \text{(D.119)}$$

- Partial derivative of $E_{sde}(t)$ with respect to S_{xx}:

$$\begin{aligned}\frac{\partial E_{sde}(t)}{\partial S_{xx}} &= \frac{1}{2}\frac{\partial}{\partial S_{xx}}\Big(\Sigma_x^{-1}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Big) \qquad \text{(D.120)} \\ &=\frac{1}{2}\Big(\Sigma_x^{-1}\frac{\partial}{\partial S_{xx}}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\frac{\partial}{\partial S_{xx}}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\frac{\partial}{\partial S_{xx}}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Big) \qquad \text{(D.121)}\end{aligned}$$

- Partial derivative of $E_{sde}(t)$ with respect to S_{xy}:

$$\begin{aligned}\frac{\partial E_{sde}(t)}{\partial S_{xy}} &= \frac{1}{2}\frac{\partial}{\partial S_{xy}}\Bigg(\Sigma_x^{-1}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Bigg) \qquad (D.122) \\ &= \frac{1}{2}\Bigg(\Sigma_x^{-1}\frac{\partial}{\partial S_{xy}}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\frac{\partial}{\partial S_{xy}}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\frac{\partial}{\partial S_{xy}}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Bigg) \qquad (D.123)\end{aligned}$$

- Partial derivative of $E_{sde}(t)$ with respect to S_{xz}:

$$\begin{aligned}\frac{\partial E_{sde}(t)}{\partial S_{xz}} &= \frac{1}{2}\frac{\partial}{\partial S_{xz}}\Bigg(\Sigma_x^{-1}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Bigg) \qquad (D.124) \\ &= \frac{1}{2}\Bigg(\Sigma_x^{-1}\frac{\partial}{\partial S_{xz}}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\frac{\partial}{\partial S_{xz}}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\frac{\partial}{\partial S_{xz}}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Bigg) \qquad (D.125)\end{aligned}$$

- Partial derivative of $E_{sde}(t)$ with respect to S_{yy}:

$$\begin{aligned}\frac{\partial E_{sde}(t)}{\partial S_{yy}} &= \frac{1}{2}\frac{\partial}{\partial S_{yy}}\Bigg(\Sigma_x^{-1}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Bigg) \qquad (D.126) \\ &= \frac{1}{2}\Bigg(\Sigma_x^{-1}\frac{\partial}{\partial S_{yy}}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\frac{\partial}{\partial S_{yy}}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\frac{\partial}{\partial S_{yy}}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Bigg) \qquad (D.127)\end{aligned}$$

- Partial derivative of $E_{sde}(t)$ with respect to S_{yz}:

$$\begin{aligned}\frac{\partial E_{sde}(t)}{\partial S_{yz}} &= \frac{1}{2}\frac{\partial}{\partial S_{yz}}\Big(\Sigma_x^{-1}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Big) \qquad \text{(D.128)}\\ &= \frac{1}{2}\Big(\Sigma_x^{-1}\frac{\partial}{\partial S_{yz}}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\frac{\partial}{\partial S_{yz}}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\frac{\partial}{\partial S_{yz}}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Big) \qquad \text{(D.129)}\end{aligned}$$

- Partial derivative of $E_{sde}(t)$ with respect to S_{zz}:

$$\begin{aligned}\frac{\partial E_{sde}(t)}{\partial S_{zz}} &= \frac{1}{2}\frac{\partial}{\partial S_{zz}}\Big(\Sigma_x^{-1}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Big) \qquad \text{(D.130)}\\ &= \frac{1}{2}\Big(\Sigma_x^{-1}\frac{\partial}{\partial S_{zz}}\left\langle(\sigma y-\sigma x+A_{11}x+A_{12}y+A_{13}z-b_1)^2\right\rangle_{q_t} \\ &+\Sigma_y^{-1}\frac{\partial}{\partial S_{zz}}\left\langle(\rho x-y-xz+A_{21}x+A_{22}y+A_{23}z-b_2)^2\right\rangle_{q_t} \\ &+\Sigma_z^{-1}\frac{\partial}{\partial S_{zz}}\left\langle(xy-\beta z+A_{31}x+A_{32}y+A_{33}z-b_3)^2\right\rangle_{q_t}\Big) \qquad \text{(D.131)}\end{aligned}$$

Gradients with respect to the variational parameters

The general expressions of the gradients of E_{sde} with respect to $\mathbf{A}_t$ and $\mathbf{b}_t$, are given in Appendix A as follows:

$$\nabla_{\mathbf{b}_t}E_{sde}(t) = -\boldsymbol{\Sigma}^{-1}\left(\langle\mathbf{f}(\mathbf{x}_t)\rangle_{q_t}+\mathbf{A}_t\langle\mathbf{x}_t\rangle_{q_t}-\mathbf{b}_t\right) \qquad \text{(D.132)}$$

$$\nabla_{\mathbf{A}_t}E_{sde}(t) = \boldsymbol{\Sigma}^{-1}\left(\langle\nabla_{\mathbf{x}_t}\mathbf{f}(\mathbf{x}_t)\rangle_{q_t}+\mathbf{A}_t\right)\mathbf{S}_t-\nabla_{\mathbf{b}_t}E_{sde}(t)\mathbf{m}_t^\top \qquad \text{(D.133)}$$

Equation D.132, requires the expectation of the drift function. This is given as follows:

$$\langle \mathbf{f}(\mathbf{x}_t)\rangle_{q_t} = \left\langle \begin{bmatrix} \sigma(y_t - x_t) \\ \rho x_t - y_t - x_t z_t \\ x_t y_t - \beta z_t \end{bmatrix} \right\rangle_{q_t} \tag{D.134}$$

$$= \begin{bmatrix} \langle \sigma(y_t - x_t)\rangle_{q_t} \\ \langle \rho x_t - y_t - x_t z_t\rangle_{q_t} \\ \langle x_t y_t - \beta z_t\rangle_{q_t} \end{bmatrix} \tag{D.135}$$

$$= \begin{bmatrix} \sigma(\langle y_t\rangle_{q_t} - \langle x_t\rangle_{q_t}) \\ \rho\,\langle x_t\rangle_{q_t} - \langle y_t\rangle_{q_t} - \langle x_t z_t\rangle_{q_t} \\ \langle x_t y_t\rangle_{q_t} - \beta\,\langle z_t\rangle_{q_t} \end{bmatrix} \tag{D.136}$$

$$= \begin{bmatrix} \sigma(m_y - m_x) \\ \rho m_x - m_y - S_{xz} - m_x m_z \\ S_{xy} + m_x m_y - \beta m_z \end{bmatrix}_t , \tag{D.137}$$

where m_x, m_y and m_z are the marginal means on each dimension (i.e. $\mathbf{m}_t = [m_x\ m_y\ m_z]_t^\top$) and also we have used the identity $\langle x_t z_t\rangle_{q_t} = S_{xz} + m_x m_z$, where S_{xz} is the covariance between x and z.

Furthermore Eq.(D.133) requires the expectation of the Jacobian matrix $\nabla_{\mathbf{x}_t}\mathbf{f}(\mathbf{x}_t)$, with respect to the Gaussian approximation q_t. This is computed as:

$$\langle \nabla_{\mathbf{x}_t}\mathbf{f}(\mathbf{x}_t)\rangle_{q_t} = \left\langle \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho - z_t & -1 & -x_t \\ y_t & x_t & -\beta \end{bmatrix} \right\rangle_{q_t} \tag{D.138}$$

$$= \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho - \langle z_t\rangle_{q_t} & -1 & -\langle x_t\rangle_{q_t} \\ \langle y_t\rangle_{q_t} & \langle x_t\rangle_{q_t} & -\beta \end{bmatrix} \tag{D.139}$$

$$= \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho - m_z & -1 & -m_x \\ m_y & m_x & -\beta \end{bmatrix}_t , \tag{D.140}$$

with 't' denoting dependency on time.

Gradients with respect to the (hyper-) parameters

The (hyper-) parameters that need to be estimated in the three dimensional stochastic Lorenz system is a set of six parameters, three in the drift vector $\boldsymbol{\theta} = [\sigma\ \rho\ \beta]^\top$ and the diagonal three elements of the system noise covariance matrix $\boldsymbol{\Sigma}$, (i.e. Σ_x, Σ_y and Σ_z). Below it is shown how the gradients of E_{sde} with respect to these parameters can be computed without approximation errors.

Gradient of E_{sde} w.r.t. drift parameter θ: Starting with the drift vector parameters, we recall that the general expression given in Appendix (A), is:

$$\nabla_{\boldsymbol{\theta}} E_{sde}(t) = \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \boldsymbol{\Sigma}^{-1} (\nabla_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}_t)) \right\rangle_{q_t} \tag{D.141}$$

In order to proceed the gradient of the drift function with respect to the parameter vector $(\nabla_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}_t))$ has to computed. As shown below this is given by:

$$\nabla_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}_t) = \begin{bmatrix} \frac{\partial(\sigma(y-x))}{\partial\sigma} & \frac{\partial(\sigma(y-x))}{\partial\rho} & \frac{\partial(\sigma(y-x))}{\partial\beta} \\ \frac{\partial(\rho x-y-xz)}{\partial\sigma} & \frac{\partial(\rho x-y-xz)}{\partial\rho} & \frac{\partial(\rho x-y-xz)}{\partial\beta} \\ \frac{\partial(xy-\beta z)}{\partial\sigma} & \frac{\partial(xy-\beta z)}{\partial\rho} & \frac{\partial(xy-\beta z)}{\partial\beta} \end{bmatrix}_t \tag{D.142}$$

$$= \begin{bmatrix} (y-x) & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & -z \end{bmatrix}_t , \tag{D.143}$$

where the sub-script ’t’ denotes time dependency. Using the result from Equation D.143, the derivation of the necessary gradient yields:

$$\nabla_{\boldsymbol{\theta}} E_{sde}(t) = \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \boldsymbol{\Sigma}^{-1} (\nabla_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}_t)) \right\rangle_{q_t} \tag{D.144}$$

$$= \left\langle \mathbf{v}_t^\top \times \begin{bmatrix} \Sigma_x & 0 & 0 \\ 0 & \Sigma_y & 0 \\ 0 & 0 & \Sigma_z \end{bmatrix}^{-1} \times \begin{bmatrix} (y_t - x_t) & 0 & 0 \\ 0 & x_t & 0 \\ 0 & 0 & -z_t \end{bmatrix} \right\rangle_{q_t} \tag{D.145}$$

$$= \left\langle \mathbf{v}_t^\top \times \begin{bmatrix} \Sigma_x^{-1}(y_t - x_t) & 0 & 0 \\ 0 & \Sigma_y^{-1} x_t & 0 \\ 0 & 0 & -\Sigma_z^{-1} z_t \end{bmatrix} \right\rangle_{q_t} \tag{D.146}$$

$$= \left\langle \begin{bmatrix} v_1 \Sigma_x^{-1}(y-x) \\ v_2 \Sigma_y^{-1} x \\ -v_3 \Sigma_z^{-1} z \end{bmatrix}_t \right\rangle_{q_t} \tag{D.147}$$

$$= \begin{bmatrix} \Sigma_x^{-1} \langle v_1 * (y-x) \rangle_{q_t} \\ \Sigma_y^{-1} \langle v_2 * x \rangle_{q_t} \\ -\Sigma_z^{-1} \langle v_3 * z \rangle_{q_t} \end{bmatrix}_t , \tag{D.148}$$

where the vector $\mathbf{v}_t = [v_1 \; v_2 \; v_3]^\top$ as defined in Equation (D.104).

Gradient of E_{sde} w.r.t. noise covariance matrix $\boldsymbol{\Sigma}$: Similarly, the gradient of E_{sde} with respect to the system noise covariance can be computed by using the general multivariate expression. However, things here are more simple because as one can see from Equation (D.150), all the necessary expectations have already been precomputed in previous steps, hence there is no need for additional computational burden.

$$\nabla_{\mathbf{\Sigma}} E_{sde}(t) = -\frac{1}{2}\mathbf{\Sigma}^{-1}\left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^{\top}\right\rangle_{q_t}\mathbf{\Sigma}^{-1} \tag{D.149}$$

$$= -\frac{1}{2}\mathbf{\Sigma}^{-1}\left\langle \mathbf{v}_t\mathbf{v}_t^{\top}\right\rangle_{q_t}\mathbf{\Sigma}^{-1} \tag{D.150}$$

Optimal initial values of the variational parameters

Unlike the univariate systems where the initial values for the all the *discretized* variational parameters could be represented in compact notation as $\mathbf{a}(k)$ and $\mathbf{b}(k)$, here for the multivariate Lorenz system the notation is slightly altered to $\mathbf{A}_t(k)$ and $\mathbf{b}_t(k)$, with subscript 't' denoting discrete time instant, whereas the index 'k' represents algorithmic time in the optimisation procedure (i.e. number of iterations).

Initial linear parameter: The general expression for the linear parameter at time 't' is given by:

$$\mathbf{A}_t(k) = -\left\langle \nabla_{\mathbf{x}_t}\mathbf{f}(\mathbf{x}_t)\right\rangle_{q_t} + 2\mathbf{\Sigma}\mathbf{\Psi}_t(k) \tag{D.151}$$

and using Equation (D.140), the initial iteration $k = 0$ becomes:

$$\mathbf{A}_t(0) = -\left\langle \nabla_{\mathbf{x}_t}\mathbf{f}(\mathbf{x}_t)\right\rangle_{q_t} \tag{D.152}$$

$$= -\begin{bmatrix} -\sigma & \sigma & 0 \\ \rho - m_z & -1 & -m_x \\ m_y & m_x & -\beta \end{bmatrix}_{t,\ k=0} \tag{D.153}$$

$$= \begin{bmatrix} \sigma & -\sigma & 0 \\ m_z - \rho & 1 & m_x \\ -m_y & -m_x & \beta \end{bmatrix}_{t,\ k=0} \tag{D.154}$$

where $\mathbf{A}_t(0) \in \Re^{3\times 3}$. In order to initialize the variational linear parameter one must have an initial set of values for the marginal means ($m_{xt}(k=0)$, $m_{zt}(k=0)$ and $m_{zt}(k=0)$). This problem can be solved by interpolating the observations with cubic splines (on each dimension separately), or using any other method that will produce a smooth approximation of the mean path.

Initial bias parameter: Similarly the general expression for the offset parameter at time 't' is given by:

$$\mathbf{b}_t(k) = \left\langle \mathbf{f}(\mathbf{x}_t)\right\rangle_{q_t} + \mathbf{A}_t(k) * \mathbf{m}_t(k) - \mathbf{\Sigma}\boldsymbol{\lambda}_t(k) \tag{D.155}$$

The expectation of the drift function for the L3D is given by Equation (D.137). Hence the initial iteration, $k = 0$ is:

$$\mathbf{b}_t(0) = \langle \mathbf{f}(\mathbf{x}_t) \rangle_{q_t} + \mathbf{A}_t(0) * \mathbf{m}_t(0) \tag{D.156}$$

$$= \begin{bmatrix} \sigma(m_y - m_x) \\ \rho m_x - m_y - S_{xz} - m_x m_z \\ S_{xy} + m_x m_y - \beta m_z \end{bmatrix}_{t,\ k=0} + \begin{bmatrix} \sigma & -\sigma & 0 \\ m_z - \rho & 1 & m_x \\ -m_y & -m_x & \beta \end{bmatrix}_{t,\ k=0} \times \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}_{t,\ k=0} \tag{D.157}$$

$$= \begin{bmatrix} \sigma(m_y - m_x) \\ \rho m_x - m_y - S_{xz} - m_x m_z \\ S_{xy} + m_x m_y - \beta m_z \end{bmatrix}_{t,\ k=0} + \begin{bmatrix} \sigma(m_x - m_y) \\ m_x(m_z - \rho) + m_y + m_x m_z \\ -m_x m_y - m_x m_y + \beta m_z \end{bmatrix}_{t,\ k=0} \tag{D.158}$$

$$= \begin{bmatrix} 0 \\ m_x m_z - S_{xz} \\ S_{xy} - m_x m_y \end{bmatrix}_{t,\ k=0} \tag{D.159}$$

where $\mathbf{b}_t(0) \in \Re^3$ and $m_{xt}(k = 0)$, $m_{zt}(k = 0)$, $m_{zt}(k = 0)$ are given as above for the initial linear parameter $\mathbf{A}_t(0)$, while $S_{xz}(k = 0)$ and $S_{xy}(k = 0)$ represent covariances. To obtain these covariances is not as trivial as for the approximations of the marginal means. Here it is assumed that these values are zero at the beginning of the optimisation process. In practise, that assumption has minor effect in the performance of the algorithm at convergence.

D.4 Approximations using the unscented transformation

An alternative method for obtaining the necessary expectations, for the variational approximation framework, is presented based on the *unscented transformation* (UT) (Uhlmann, 1995). As the previous section for the L3D revealed, the analytic expressions for a multivariate system requires many computations which even when available there are prone to numerical errors (computational and derivations). The approach presented here makes the variational algorithm more generic, although that comes with the cost of introducing additional approximation errors.

The presentation of the equations follows a similar approach as seen in the previous sections, however all the mathematical expressions will remain at a higher-level without going into detail about system specific drift functions and parameter vectors. In this way a general approach will be given that can be applied to any system as long as the drift function is defined specifically. This method was applied successfully to the three and forty dimensional stochastic Lorenz systems (L3D and L40D). For the L3D, where the analytic expressions were also available the comparison between the two versions showed good match, although the unscented transformation needed careful tuning.

Energy from the SDE: Initially, the energy related to the stochastic differential equation that describes the stochastic process is given by:

$$E_{sde}(t) = \frac{1}{2}\left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))\right\rangle_{q_t} \tag{D.160}$$

Setting $\mathbf{z}_t = (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))$ and using the fact the the system noise coefficient matrix $\mathbf{\Sigma}$ is diagonal, the above expression can be rewritten as:

$$\begin{aligned} E_{sde}(t) &= \frac{1}{2}\left\langle \mathbf{z}_t^\top \mathbf{\Sigma}^{-1}\mathbf{z}_t\right\rangle_{q_t} && \text{(D.161)}\\ &= \frac{1}{2}\left[\Sigma_x^{-1}\ \Sigma_y^{-1}\ \Sigma_z^{-1}\right]\left\langle \mathbf{z}_t^2\right\rangle_{q_t}\ , && \text{(D.162)} \end{aligned}$$

where Σ_x, Σ_y and Σ_z represent the noise variance on each separate dimension of the system and the square of the vector $\mathbf{z}_t$, as appears in step (D.162), operates element-wise square. Once the vector $\mathbf{z}_t^2$ is constructed as a function and passed to the UT an approximation to the true expectation will be provided by $\tilde{E}\{\mathbf{z}_t^2\} \approx \langle \mathbf{z}_t^2\rangle_{q_t}$.

Gradient of E_{sde} w.r.t. the marginal mean vector $\mathbf{m}_t$: To compute this gradient we need to write the energy function E_{sde} as an integral and then differentiate with respect to $\mathbf{m}_t$.

$$\begin{aligned} \nabla_{\mathbf{m}_t} E_{sde}(t) &= \nabla_{\mathbf{m}_t}\left[\frac{1}{2}\left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))\right\rangle_{q_t}\right] && \text{(D.163)}\\ &= \frac{1}{2}\nabla_{\mathbf{m}_t}\int (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))q(\mathbf{x}_t)d\mathbf{x}_t && \text{(D.164)}\\ &= \frac{1}{2}\int (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))\nabla_{\mathbf{m}_t} q(\mathbf{x}_t)d\mathbf{x}_t && \text{(D.165)}\\ &= \frac{1}{2}\int (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))\mathbf{S}_t^{-1}(\mathbf{x}_t - \mathbf{m}_t)q(\mathbf{x}_t)d\mathbf{x}_t && \text{(D.166)}\\ &= \frac{1}{2}\left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))\mathbf{S}_t^{-1}(\mathbf{x}_t - \mathbf{m}_t)\right\rangle_{q_t} && \text{(D.167)}\\ &= \frac{1}{2}\left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))\mathbf{S}_t^{-1}\mathbf{x}_t\right\rangle_{q_t}\\ &\quad - \frac{1}{2}\left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))\right\rangle_{q_t}\mathbf{S}_t^{-1}\mathbf{m}_t && \text{(D.168)}\\ &= \frac{1}{2}\left\langle \mathbf{z}_t^\top \mathbf{\Sigma}^{-1}\mathbf{z}_t\mathbf{S}_t^{-1}\mathbf{x}_t\right\rangle_{q_t} - E_{sde}(t)\mathbf{S}_t^{-1}\mathbf{m}_t && \text{(D.169)} \end{aligned}$$

where the vector $\mathbf{z}_t$ is a shorthand notation for $(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))$ and for the Gaussian distribution $q(\mathbf{x}_t|\mathbf{m}_t, \mathbf{S}_t)$, we have used $\nabla_{\mathbf{m}_t} q(\mathbf{x}_t) = \mathbf{S}_t^{-1}(\mathbf{x}_t - \mathbf{m}_t)$. This is proven as follows:

Proof:

$$\nabla_{\mathbf{m}_t} q(\mathbf{x}_t) = \nabla_{\mathbf{m}_t}\left[(2\pi)^{-\frac{D}{2}}|\mathbf{S}_t|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)}\right] \tag{D.170}$$

$$= (2\pi)^{-\frac{D}{2}}|\mathbf{S}_t|^{-\frac{1}{2}}\left(\nabla_{\mathbf{m}_t} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)}\right) \tag{D.171}$$

$$= (2\pi)^{-\frac{D}{2}}|\mathbf{S}_t|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)}\nabla_{\mathbf{m}_t}\left(-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)\right) \tag{D.172}$$

$$= q(\mathbf{x}_t)\left(-\frac{1}{2}(-2\mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t))\right) \tag{D.173}$$

$$= \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)q(\mathbf{x}_t) \tag{D.174}$$

Having precomputed the $E_{sde}(t)$ as shown in Equation (D.162), the new approximate expectation that one has to compute is $\tilde{E}\{\mathbf{z}_t^\top \mathbf{\Sigma}^{-1}\mathbf{z}_t\mathbf{S}_t^{-1}\mathbf{x}_t\}$ (see Equation D.169).

Gradient of E_{sde} w.r.t. the marginal covariance matrix $\mathbf{S}_t$: For the gradient of E_{sde} with respect to $\mathbf{S}_t$, a similar procedure is followed. First E_{sde} is expressed as an integral and then the requested derivative is computed.

$$\nabla_{\mathbf{S}_t} E_{sde}(t) = \nabla_{\mathbf{S}_t}\left[\frac{1}{2}\left\langle(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))\right\rangle_{q_t}\right] \tag{D.175}$$

$$= \frac{1}{2}\nabla_{\mathbf{S}_t}\left\langle(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))\right\rangle_{q_t} \tag{D.176}$$

$$= \frac{1}{2}\int(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))\nabla_{\mathbf{S}_t} q(\mathbf{x}_t)d\mathbf{x}_t \tag{D.177}$$

$$= \frac{1}{2}\int\Bigg((\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))$$
$$\times\frac{1}{2}\left(\mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}-\mathbf{S}_t^{-1}\right)q(\mathbf{x}_t)\Bigg)d\mathbf{x}_t \tag{D.178}$$

$$= \frac{1}{4}\left\langle(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))\mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}\right\rangle_{q_t}$$
$$-\frac{1}{4}\left\langle(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))\right\rangle_{q_t}\mathbf{S}_t^{-1} \tag{D.179}$$

$$= \frac{1}{4}\left\langle(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))^\top \mathbf{\Sigma}^{-1}(\mathbf{f}(\mathbf{x}_t)-\mathbf{g}_L(\mathbf{x}_t))\mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}\right\rangle_{q_t}$$
$$-\frac{1}{2}E_{sde}(t)\mathbf{S}_t^{-1} \tag{D.180}$$

$$= \frac{1}{4}\left\langle\mathbf{z}_t^\top \mathbf{\Sigma}^{-1}\mathbf{z}_t\mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}\right\rangle_{q_t}-\frac{1}{2}E_{sde}(t)\mathbf{S}_t^{-1} \tag{D.181}$$

Hence the new expectation that has to be approximated with the UT is $\tilde{E}\{\mathbf{z}_t^\top \mathbf{\Sigma}^{-1}\mathbf{z}_t\mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}\}$. Furthermore the gradient $\nabla_{\mathbf{S}_t} q(\mathbf{x}_t)$ (see step D.177), is provided below:

Proof:

$$\begin{aligned}
\nabla_{\mathbf{S}_t} q(\mathbf{x}_t) &= \nabla_{\mathbf{S}_t}\left((2\pi)^{-\frac{D}{2}}|\mathbf{S}_t|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)}\right) && \text{(D.182)}\\
&= (2\pi)^{-\frac{D}{2}}\left(\nabla_{\mathbf{S}_t}|\mathbf{S}_t|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)} + |\mathbf{S}_t|^{-\frac{1}{2}}\nabla_{\mathbf{S}_t} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)}\right) && \text{(D.183)}\\
&= (2\pi)^{-\frac{D}{2}}\Bigg(-\frac{1}{2}|\mathbf{S}_t|^{-\frac{1}{2}}\mathbf{S}_t^{-1} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)}\\
&\quad + |\mathbf{S}_t|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)}\nabla_{\mathbf{S}_t}\left(-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)\right)\Bigg) && \text{(D.184)}\\
&= (2\pi)^{-\frac{D}{2}}\Bigg(-\frac{1}{2}|\mathbf{S}_t|^{-\frac{1}{2}}\mathbf{S}_t^{-1} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)}\\
&\quad + \frac{1}{2}|\mathbf{S}_t|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)}\left(\mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}\right)\Bigg) && \text{(D.185)}\\
&= (2\pi)^{-\frac{D}{2}}|\mathbf{S}_t|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)} \times \frac{1}{2}\left(\mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)(\mathbf{x}_t-\mathbf{m}_t)^\top \mathbf{S}_t^{-1} - \mathbf{S}_t^{-1}\right)\\
& && \text{(D.186)}\\
&= \frac{1}{2}\left(\mathbf{S}_t^{-1}(\mathbf{x}_t-\mathbf{m}_t)(\mathbf{x}_t-\mathbf{m}_t)^\top - \mathbf{I}\right)\mathbf{S}_t^{-1} q(\mathbf{x}_t)\ , && \text{(D.187)}
\end{aligned}$$

with $\mathbf{I} \in \Re^{D\times D}$ the identity matrix.

Gradient of E_{sde} w.r.t. the (hyper-) parameters θ and Σ: The general multivariate expressions of the gradients with respect to the (hyper-) parameters are given in Equation (D.189), for the drift parameter vector

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} E_{sde}(t) &= \left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top \boldsymbol{\Sigma}^{-1}\left(\nabla_{\boldsymbol{\theta}}\mathbf{f}(\mathbf{x}_t)\right)\right\rangle_{q_t} && \text{(D.188)}\\
&= \left\langle \mathbf{z}_t^\top \boldsymbol{\Sigma}^{-1}\left(\nabla_{\boldsymbol{\theta}}\mathbf{f}(\mathbf{x}_t)\right)\right\rangle_{q_t}\ , && \text{(D.189)}
\end{aligned}$$

and in Equation (D.191) for the system noise matrix

$$\begin{aligned}
\nabla_{\boldsymbol{\Sigma}} E_{sde}(t) &= -\frac{1}{2}\boldsymbol{\Sigma}^{-1}\left\langle (\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))(\mathbf{f}(\mathbf{x}_t) - \mathbf{g}_L(\mathbf{x}_t))^\top\right\rangle_{q_t}\boldsymbol{\Sigma}^{-1} && \text{(D.190)}\\
&= -\frac{1}{2}\boldsymbol{\Sigma}^{-1}\left\langle \mathbf{z}_t\mathbf{z}_t^\top\right\rangle_{q_t}\boldsymbol{\Sigma}^{-1}\ . && \text{(D.191)}
\end{aligned}$$

It is obvious from Equation (D.189), that before computing this expectation, with the UT approximation, the gradient of drift function with respect to the drift parameters must be computed in advance for the system studied. On the contrary the gradient with respect to the system noise requires no additional computations since the required expectation $\left\langle \mathbf{z}_t\mathbf{z}_t^\top\right\rangle_{q_t}$ has already been approximated when computing the energy term E_{sde} (see Equation D.162).

E Approximate solutions of the moment equations

Chapter 4 shows that the dynamics of the proposed linear approximation q_t to the true posterior process p_t can be described by a set of ordinary differential equations; one for the marginal, at time ’t’, means Eq. (4.12) and one for the variances Eq. (4.13). The solution to the above ODEs is provided by a first order Euler discretisation scheme with a small time step (e.g. $\delta t = 0.01$), to achieve good accuracy. Chapter 6 hinted that the approximation of the variational parameters $\mathbf{A}_t$ and $\mathbf{b}_t$ with local polynomials can further improve the precision of the ODE solutions by applying higher order integration schemes (such as Runge-Kutta 2’nd order). However, the ODEs must still be discretised and solved iteratively, which reduces the applicability of the algorithm to very high dimensional systems.

This Appendix provides approximate solutions to the moment equations for the marginal means and variances and shows that under the current variational framework, closed form solutions for Equations (4.12) and (4.13) are not possible. Nevertheless, future work in parametrising and making further assumptions about the linear variational parameter $\mathbf{A}_t$ might provide more efficient ways to solve the moment equations. To ease the presentation of the following derivations only univariate systems are considered.

E.1 Marginal means

The equation that provides the marginal mean (Chapter 4) is given by:

$$\frac{dm(t)}{dt} = -a(t)m(t) + b(t)\ , \quad m(t_0) = m_0\ . \tag{E.1}$$

with $m_0 \in \Re$ denoting the initial condition and $a(t),\ b(t) \in \Re$ the time dependent linear and offset variational parameters.

Set:

$$Q(t) = -\int_{t_0}^{t} a(k)dk\ , \quad \text{with} \tag{E.2}$$

$$\frac{dQ(t)}{dt} = -a(t)\ . \tag{E.3}$$

Then differentiating $m(t)\exp\{-Q(t)\}$ w.r.t. the time t, yields:

$$\frac{d}{dt}\left(m(t)\exp\{-Q(t)\}\right) = \dot{m}(t)\exp\{-Q(t)\} + m(t)\frac{d}{dt}\exp\{-Q(t)\} \tag{E.4}$$

$$= \dot{m}(t)\exp\{-Q(t)\} - m(t)\frac{dQ(t)}{dt}\exp\{-Q(t)\} \tag{E.5}$$

$$= \dot{m}(t)\exp\{-Q(t)\} + m(t)a(t)\exp\{-Q(t)\} \tag{E.6}$$

$$= \Big(\dot{m}(t) + m(t)a(t)\Big)\exp\{-Q(t)\} \tag{E.7}$$

$$= b(t)\exp\{-Q(t)\}\ , \tag{E.8}$$

where $\frac{dQ(t)}{dt} = -a(t)$, from Equation (E.3), and $\dot{m}(t)$ denotes the time derivative.

Integrating both sides of Equation (E.8) yields:

$$\int_{t_0}^{t} \frac{d}{dk}\left(m(k)\exp\{-Q(k)\}\right)dk = \int_{t_0}^{t} b(k)\exp\{-Q(k)\}dk\ . \tag{E.9}$$

The first integral on the right hand side of Equation (E.9) is:

$$\int_{t_0}^{t} \frac{d}{dk}\left(m(k)\exp\{-Q(k)\}\right)dk = \left[m(k)\exp\{-Q(k)\}\right]_{t_0}^{t} \tag{E.10}$$

$$= m(t)\exp\{-Q(t)\} - m(t_0)\underbrace{\exp\{-Q(t_0)\}}_{=1} \tag{E.11}$$

$$= m(t)\exp\{-Q(t)\} - m_0\ , \tag{E.12}$$

where $m(t_0) = m_0$ and the exponent $-Q(t)$ for $t = t_0$ becomes $-Q(t_0) = \int_{t_0}^{t_0} a(k)dk = 0$, therefore $\exp\{0\} = 1$. Hence:

$$m(t)\exp\{-Q(t)\} - m_0 = \int_{t_0}^{t} b(k)\exp\{-Q(k)\}dk\ . \tag{E.13}$$

Finally, multiplying both sides of Equation (E.13) with $\exp\{Q(t)\}$ and then re-arranging results:

$$m(t) = \left(m(t_0) + \int_{t_0}^{t} b(k)\exp\{-Q(k)\}dk\right)\exp\{Q(t)\}\ . \tag{E.14}$$

At this point one can make use of the polynomial approximations of $a(t)$ and $b(t)$:

$$\tilde{a}(t) = \sum_{i=0}^{M_a} a_i t^i \quad \text{and} \quad \tilde{b}(t) = \sum_{i=0}^{M_b} b_i t^i \ . \tag{E.15}$$

where M_a, $M_b \in N$ denote the order of the polynomial approximation for $a(t)$ and $b(t)$ respectively. Here, unlike the presentation in Chapter 6, the order of the polynomials for the two variational parameters is allowed to be different to provide a more general presentation.

To compute the exponential $\exp\{Q(t)\}$ we make use of the above result:

$$-\int_{t_0}^{t} a(k)dk \approx -\int_{t_0}^{t} \tilde{a}(k)dk \tag{E.16}$$

$$\approx -\int_{t_0}^{t} \sum_{i=0}^{M_a} a_i k^i dk \tag{E.17}$$

$$\approx -\sum_{i=0}^{M_a} a_i \int_{t_0}^{t} k^i dk \tag{E.18}$$

$$\approx -\sum_{i=0}^{M_a} a_i \left[\frac{k^{i+1}}{i+1}\right]_{t_0}^{t} \tag{E.19}$$

$$\approx -\sum_{i=0}^{M_a} a_i \left[\frac{t^{i+1} - t_0^{i+1}}{i+1}\right] , \tag{E.20}$$

which leads to:

$$\exp\{Q(t)\} \approx \exp\{-\sum_{i=0}^{Mo} a_i \left[\frac{t^{i+1} - t_0^{i+1}}{i+1}\right]\} \ . \tag{E.21}$$

The second integral on the left hand side of Equation (E.9) is:

$$\int_{t_0}^{t} b(k)\exp\{-Q(k)\}dk \approx \int_{t_0}^{t} \tilde{b}(k)\exp\{-Q(k)\}dk \tag{E.22}$$

$$\approx \int_{t_0}^{t} \sum_{i=0}^{M_b} b_i k^i \exp\{-Q(k)\}dk \tag{E.23}$$

$$\approx \sum_{i=0}^{M_b} b_i \int_{t_0}^{t} k^i \exp\{-Q(k)\}dk \tag{E.24}$$

$$\approx \sum_{i=0}^{M_b} b_i \int_{t_0}^{t} k^i \exp\{\sum_{j=0}^{M_a} a_j \left[\frac{k^{j+1} - t_0^{j+1}}{j+1}\right]\}dk \ . \tag{E.25}$$

Hence, the final expression for the marginal means becomes:

$$m(t) \approx \left(m_0 + \sum_{i=0}^{M_b} b_i \int_{t_0}^{t} k^i \exp\{\sum_{n=0}^{M_a} a_n \left[\frac{k^{n+1} - t_0^{n+1}}{n+1}\right]\}dk\right) \exp\{-\sum_{j=0}^{M_a} a_j \left[\frac{t^{j+1} - t_0^{j+1}}{j+1}\right]\} \ . \tag{E.26}$$

Setting the initial time instant to $t_0 = 0$ yields:

$$m(t) \approx \left(m_0 + \sum_{i=0}^{M_b} b_i \underbrace{\int_{0}^{t} k^i \exp\{\sum_{n=0}^{M_a} a_n \left[\frac{k^{n+1}}{n+1}\right]\}dk}_{\text{Int1}}\right) \exp\{-\sum_{j=0}^{M_a} a_j \left[\frac{t^{j+1}}{j+1}\right]\} \ . \tag{E.27}$$

This equation, to the author’s best knowledge, does not have a closed form solution because integral “Int1” cannot be solved analytically for arbitrary values of M_a and M_b (Gradshteyn and Ryzhik, 2007).

E.2 Marginal variances

In a similar way the equation that gives the marginal variance is given by:

$$\frac{ds_t}{dt} = -2a_t s_t + \sigma^2 \; , \quad s(t_0) = s_0 \tag{E.28}$$

with $s_0 \in \Re$ denoting the initial condition and $\sigma^2 \in \Re$ the (constant) system noise variance.

Setting:

$$Z(t) = -2\int_{t_0}^{t} a(k)dk \quad \text{with} \quad \frac{dZ(t)}{dt} = -2a(t) \tag{E.29}$$

and working the same way as in the previous section the following equation is derived:

$$s(t) \approx \left(s_0 + \sigma^2 \int_{t_0}^{t} \exp\{-Z(k)\}dk\right) \exp\{Z(k)\} \; , \tag{E.30}$$

which leads to the final expression for the approximate marginal variances:

$$s(t) \approx \left(s_0 + \sigma^2 \int_{t_0}^{t} \exp\{2\sum_{j=0}^{M_a} a_j \left[\frac{k^{j+1} - t_0^{j+1}}{j+1}\right]\}dk\right) \exp\{-2\sum_{i=0}^{M_a} a_i \left[\frac{t^{i+1} - t_0^{i+1}}{i+1}\right]\} \tag{E.31}$$

and for $t_0 = 0$ the expression simplifies to:

$$s(t) \approx \left(s_0 + \sigma^2 \int_{0}^{t} \exp\{2\sum_{j=0}^{M_a} a_j \left[\frac{k^{j+1}}{j+1}\right]\}dk\right) \exp\{-2\sum_{i=0}^{M_a} a_i \left[\frac{t^{i+1}}{i+1}\right]\} \tag{E.32}$$

F Gaussian moments and related derivatives

This Appendix provides the uncentered moments, up to and including 8’th order, of a univariate Gaussian random variable x_t, where m_t and s_t are respectively the marginal mean and variance at time ’t’.

Uncentered moments:

$$\left\langle x_t^0 \right\rangle_{q_t} = 1 \tag{F.1}$$

$$\left\langle x_t^1 \right\rangle_{q_t} = m_t \tag{F.2}$$

$$\left\langle x_t^2 \right\rangle_{q_t} = m_t^2 + s_t \tag{F.3}$$

$$\left\langle x_t^3 \right\rangle_{q_t} = m_t^3 + 3m_t s_t \tag{F.4}$$

$$\left\langle x_t^4 \right\rangle_{q_t} = m_t^4 + 6m_t^2 s_t + 3s_t^2 \tag{F.5}$$

$$\left\langle x_t^5 \right\rangle_{q_t} = m_t^5 + 10m_t^3 s_t + 15m_t s_t^2 \tag{F.6}$$

$$\left\langle x_t^6 \right\rangle_{q_t} = m_t^6 + 15m_t^4 s_t + 45m_t^2 s_t^2 + 15s_t^3 \tag{F.7}$$

$$\left\langle x_t^7 \right\rangle_{q_t} = m_t^7 + 21m_t^5 s_t + 105m_t^3 s_t^2 + 105m_t s_t^3 \tag{F.8}$$

$$\left\langle x_t^8 \right\rangle_{q_t} = m_t^8 + 28m_t^6 s_t + 210m_t^4 s_t^2 + 420m_t^2 s_t^3 + 105s_t^4 \tag{F.9}$$

From here, it is easy to derive the related derivatives with respect to the marginal means and variances at each time ’t’.

Derivative of $\left\langle x_t^k \right\rangle_{q_t}$ w.r.t. m_t:

$$\nabla_{m_t} \left\langle x_t^0 \right\rangle_{q_t} = 0 \tag{F.10}$$

$$\nabla_{m_t} \left\langle x_t^1 \right\rangle_{q_t} = 1 \tag{F.11}$$

$$\nabla_{m_t} \left\langle x_t^2 \right\rangle_{q_t} = 2m_t \tag{F.12}$$

$$\nabla_{m_t} \left\langle x_t^3 \right\rangle_{q_t} = 3(m_t^2 + s_t) \tag{F.13}$$

$$\nabla_{m_t} \left\langle x_t^4 \right\rangle_{q_t} = 4m_t^3 + 12m_t s_t \tag{F.14}$$

$$\nabla_{m_t} \left\langle x_t^5 \right\rangle_{q_t} = 5m_t^4 + 30m_t^2 s_t + 15s_t^2 \tag{F.15}$$

$$\nabla_{m_t} \left\langle x_t^6 \right\rangle_{q_t} = 6m_t^5 + 60m_t^3 s_t + 90m_t s_t^2 \tag{F.16}$$

$$\nabla_{m_t} \left\langle x_t^7 \right\rangle_{q_t} = 7m_t^6 + 105m_t^4 s_t + 315m_t^2 s_t^2 + 105s_t^3 \tag{F.17}$$

$$\nabla_{m_t} \left\langle x_t^8 \right\rangle_{q_t} = 8m_t^7 + 168m_t^5 s_t + 840m_t^3 s_t^2 + 840m_t s_t^3 \tag{F.18}$$

Derivative of $\left\langle x_t^k \right\rangle_{q_t}$ w.r.t. s_t:

$$\nabla_{s_t} \left\langle x_t^0 \right\rangle_{q_t} = 0 \tag{F.19}$$

$$\nabla_{s_t} \left\langle x_t^1 \right\rangle_{q_t} = 0 \tag{F.20}$$

$$\nabla_{s_t} \left\langle x_t^2 \right\rangle_{q_t} = 1 \tag{F.21}$$

$$\nabla_{s_t} \left\langle x_t^3 \right\rangle_{q_t} = 3m_t \tag{F.22}$$

$$\nabla_{s_t} \left\langle x_t^4 \right\rangle_{q_t} = 6(m_t^2 + s_t) \tag{F.23}$$

$$\nabla_{s_t} \left\langle x_t^5 \right\rangle_{q_t} = 10m_t^3 + 30m_t s_t \tag{F.24}$$

$$\nabla_{s_t} \left\langle x_t^6 \right\rangle_{q_t} = 15m_t^4 + 90m_t^2 s_t + 45s_t^2 \tag{F.25}$$

$$\nabla_{s_t} \left\langle x_t^7 \right\rangle_{q_t} = 21m_t^5 + 210m_t^3 s_t + 315m_t s_t^2 \tag{F.26}$$

$$\nabla_{s_t} \left\langle x_t^8 \right\rangle_{q_t} = 28m_t^6 + 420m_t^4 s_t + 1260m_t^2 s_t^2 + 420s_t^3 \tag{F.27}$$