Topics in Sequential Monte Carlo Samplers

A dissertation submitted to the University of Cambridge for the degree of Master of Science MSc.

Gareth William Peters, Selwyn College January 2005

Signal Processing Group
Department of Engineering
University of Cambridge

To my mother and my fiancée, for all their love and support.

Don't judge each day by the harvest you reap, but by the seeds you plant.

-Robert Louis Stevenson, 1850-1894

Declaration

This dissertation is submitted for the degree of Master of Science. The dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. No part of this has been submitted to any other university. This dissertation contains fewer figures than the limit of 150 and fewer than 65,000 words; there are approximately 21 figures and 46,000 words.

Gareth William Peters

Acknowledgements

I am very grateful to my supervisor Dr. Arnaud Doucet for his advice, guidance and encouragement throughout the thesis. I would also like to thank him for the many interesting academic discussions that we have had during the course of this research, I have benefited immensely. I must also thank Professor Pierre Del Moral for his kind hospitality in Toulouse and his patience. Prof. Del Moral and Dr. Doucet have taught me a great deal about scientific research and successful research practice, which I am sure will prove invaluable.

I am thankful for the discussions that I have had with many people in the Cambridge University Signal Processing Group, in particular, Adam Johansen, Mark Briers and Mathew Orton who have always been happy to spend time discussing ideas over coffee. I must also thank the head of the group Prof. William Fitzgerald who has always taken an interest in my research and been encouraging. His advice has always been insightful. I would like to thank the many people who have read this thesis and provided useful feedback, Dr. Arnaud Doucet, Dr. Simon Hill, Adam Johansen and Mark Briers.

I am also thankful to the Signal Processing group at CUED for their constant coffee supply, as without this, the thesis literally would not be written. I must thank all my friends in the Signal Processing group for making the lab a great place to work. I would like to thank Selwyn College for accepting me and also the Cambridge Commonwealth trust for their kind support.

Importantly, I would like to thank my fiancée for her constant support, her love, understanding and commitment. I would like to thank my mother for making immense sacrifices so that I may have the opportunity to study at Cambridge University. She has always encouraged and believed in my academic endeavours and without her love and support this thesis would not have been possible.

Summary

This thesis is concerned with the development of methodology for nonstandard Sequential Monte Carlo algorithms. It is very important in practice to be able to sample from a target probability distribution which can be evaluated only up to a normalising constant and does not have a standard form. Scientific disciplines in which this problem arises include; statistics, engineering, bioinformatics, finance and computer vision. In many cases using standard sampling techniques such as inversion or rejection to sample from a target distribution is not possible or proves too much of a computational burden. This has led to the development in recent years of much more advanced algorithms which allow one to obtain the required samples from the target distribution. In batch settings one typically utilises some variant of the well regarded Metropolis-Hasting algorithm. However, in on-line settings in which data is arriving sequentially, often Metropolis-Hastings is no longer a viable alternative and as a result, Sequential Monte Carlo techniques have been developed to tackle these problems. Sequential Monte Carlo utilises the idea of Importance Sampling to perform the task of sampling in on-line scenarios. It is a technique which uses a collection of particles or samples to represent the inferred posterior distribution and updates the particles as more observations are received. The algorithms developed using Sequential Monte Carlo sampling have enjoyed wide-spread use in tracking and computer vision due to the fact that they provide a natural means of sampling a state distribution of a target sequentially in time. It was not until recently that Sequential Monte Carlo approaches have started to be applied in more traditional statistical problems which would typically be handled by batch algorithms.

It is the focus of this thesis to develop a methodology that will allow one to obtain samples from a sequence of distributions which are all defined on the same fixed dimensional space. This is a non-standard idea, since standard Sequential Monte Carlo algorithms deal with situations in which the space on which the sequence of target posteriors are defined upon, grows with each iteration, as a product space. Therefore, one may view the work in this thesis as a means of turning problems, which would typically be

solved using classical batch algorithms, into a sequential problem in which the solution utilises qualities of the Sequential Monte Carlo framework. The advantage of such an approach is detailed throughout the thesis and guidelines as to when this methodology will be a viable alternative to Metropolis-Hastings have been presented. Finally several detailed examples have been provided to demonstrate how effectively the new Sequential Monte Carlo methodology performs relative to several standard algorithms.

List of Abbreviations

IS	Importance Sampling
SIS	Sequential Importance Sampling
SMC	Sequential Monte Carlo
MCMC	Markov Chain Monte Carlo
RJMCMC	Reversible Jump Markov Chain Monte Carlo
i.i.d.	independent identically distributed
ave., std.	average, standard deviation
MAP	Maximum a Posteriori
MMSE	Minimum Mean Square Error
(R)MSE	$(Root)$ Mean Square Error
SMC Samplers	Sequential Monte Carlo Samplers
TDSMC	Trans-Dimensional Sequential Monte Carlo
AIS	Annealed Importance Sampling
SA	Simulated Annealing
SVM	Support Vector Machine
RVM	Relevance Vector Machine
SLLN	Strong Law of Large Numbers
CLT	Central Limit Theorem
E_{ff}	Effective Sample Size
GLM	General Linear Model

Notation

$\mathbb{E}(.)$	expectation operator
$p(.), p(.\|.)$	probability density, conditional probability density
$p(d.)$	probability distribution
$\mathcal{N}(.; m, \sigma^2)$	Normal distribution with mean m and variance σ^2
$\mathcal{G}a(.; \mu, \upsilon)$	Gamma distribution with parameters μ and υ
$\mathcal{IG}(.; \alpha, \beta)$	Inverse Gamma distribution with parameters α and β
$\mathcal{U}(.; a, b)$	Uniform distribution over $[a, b]$
$\mathcal{P}(.; \lambda)$	Poisson distribution parameter λ
$\exp(.; \mu)$	Exponential distribution with parameter μ
$\lfloor . \rfloor$	integer part
A^T	transpose of matrix A
A'	matrix at previous iteration or neighbouring time step
A_j	matrix with j^{th} column changed
$K(x', x), K(x\|x')$	forward transition kernel giving probability
$(K_t(x', x) K_t(x\|x'))$	of moving from x' to x (at time t)
$L(x', x) L(x\|x')$	backward transition kernel giving probability
$(L_t(x', x), L_t(x\|x'))$	of moving from x' to x (at time t)
$X_{1:k,t}$	vector or sequence of k random variables at time t
$X_{1:k,1:t}$	path history of k random variables from time 1 to time t
$k_t, X_{1:k,t}$	time t have model order k_t and $X_{1:k,t}$ is interpreted as the vector or sequence of k_t random variables at time t
$X_{k_{t-\Delta}:k,t}$	$X_{k_{t-\Delta}:k,t}$ is the vector of random variables, at time t, located temporally in a window of time $[t-\Delta, t]$
$A_{k_{t-\Delta}:k,t}$	matrix constructed using $X_{k_{t-\Delta}:k,t}$ random variables
$X_{1:k\backslash a}$	vector $(X_{1:a-1}, X_{a+1:k})$, basically all the parameters except the ath
$\Rightarrow$	convergence in distribution

Contents

1 Bayesian Analysis and Models **12**

1.1 Introduction 12

1.2 Bayesian Inference 13

1.3 Definitions and Notation 16

1.4 General Aim of Analysis 17

1.5 Structure of Thesis 17

2 Monte Carlo Methods **20**

2.1 Introduction 20

2.2 Monte Carlo Methods 21

2.3 Markov Chain Monte Carlo Methods 25

2.3.1 Metropolis-Hastings 26

2.3.2 Gibbs Sampling 29

2.3.3 Reversible Jump Markov Chain Monte Carlo 30

2.4 Importance Sampling 33

2.5 Sequential Monte Carlo Methods 35

2.5.1 Sequential Importance Sampling, Resampling and MCMC Diversification Move 36

2.6 Summary 45

3 Sequential Monte Carlo Samplers **46**

3.1 Introduction 46
3.2 Motivation for SMC Samplers 47
3.3 SMC Samplers Methodology 48
3.4 SMC Samplers Specifics: Theoretical and Algorithmic Considerations 52
3.4.1 Asymptotic Analysis of Variance 52
3.4.2 Auxiliary Kernels $\{L_t\}$ 55
3.5 Applications of SMC Samplers 62
3.5.1 Bayesian Variable Selection 63
3.5.2 Application 1: Sampling from $p\left(i_{1:M} \middle| y_{1:T}, x_{1:T}\right)$ 65
3.5.3 Application 2 : Optimization of $p\left(i_{1:M} \middle| y_{1:T}, x_{1:T}\right)$ to find the Mode 71
3.6 Summary 75

4 Trans-Dimensional Sequential Monte Carlo (TDSMC) 76

4.1 Introduction 76
4.2 Motivation of TDSMC 77
4.3 TDSMC Methodology 79
4.4 TDSMC Specifics: Theoretical and Algorithmic Considerations 83
4.4.1 Multiple Moves 83
4.4.2 Construction with Auxiliary Random Variables 85
4.4.3 Update Move 86
4.4.4 Birth Move 87
4.4.5 Death Move 88
4.4.6 Adjustment Move 90
4.4.7 TDSMC Algorithm 92
4.4.8 Asymptotic Variance for TDSMC Algorithm 94
4.5 Application of TDSMC Algorithm 97
4.5.1 Application 1: Sequential Kernel Regression 98

4.6 Summary 110

5 Applications **111**

5.1 Inhomogeneous Poisson Processes 111

5.1.1 Construction and Conditions for an Inhomogeneous Poisson Process 112

5.1.2 Bayesian Model for Estimation of the Rate of an Inhomogeneous Poisson Process 113

5.1.3 Moves Used In Sequential Estimation of the Underlying Rate Function in an Inhomogeneous Poisson Process 116

5.1.4 Simulation Examples 122

5.1.5 Summary 131

5.2 General Linear Model Basis Function Regression 132

5.2.1 Rao-Blackwellised TDSMC: GLM 133

5.2.2 Move Details 135

5.2.3 Simulation Results 139

5.3 Summary 153

6 Conclusions **155**

7 Appendix **160**

7.1 Appendix 1 160

7.2 Appendix 2 162

7.3 Appendix 3 163

7.4 Appendix 4 166

7.5 Appendix 5 167

7.5.1 Simulation 1: Chapter 5 167

7.5.2 Simulation 2: Chapter 5 170

7.5.3 Simulation 3: Chapter 5 176

Chapter 1

Bayesian Analysis and Models

1.1 Introduction

This chapter provides a review of the fundamental ideas required for Bayesian analysis. There are two well studied approaches to performing probabilistic inference in the analysis of data, namely the frequentist approach and the Bayesian approach. In the classical frequentist approach one takes the view that probabilities may be seen as relative frequencies of occurrence of random variables. This approach is often associated with the work of J. Neyman and E. Pearson who described the logic of statistical hypothesis testing. Other key figures include J. Venn, R. A. Fisher, and R. von Mises. The second approach known as the Bayesian paradigm takes a different view. In a Bayesian analysis the distinction between random variables and model parameters is artificial, and all quantities may have a probability distribution associated with them, this probability represents a degree of plausibility. Basically, "Bayesians" condition on the observed data and use a probability distribution over the hypotheses. It is beyond the scope of this thesis to enter into the well documented debate over the merits of either method, instead the author suggests that the interested reader can find a more in-depth philosophical discussion on the details of each approach in [15], and the many papers of J. Berger.

1.2 Bayesian Inference

The premise of the work presented in this thesis revolves around a statistical analysis built on Bayesian methodology. The Bayesian approach to data analysis is a widely accepted means by which one may carry out modern statistical data analysis. Bayesian analysis is so named as it centres around Bayes' rule shown below in equation (1.1). It should be noted that, when one uses x and y this may be understood to represent both a single variable or a multi-dimensional vector of random variables, respectively observations. The following terminology is used; $p(x|y)$ is known as the posterior probability, $p(y|x)$ is the likelihood, $p(x)$ is the prior probability and $p(y)$ is the evidence.

$$\begin{aligned} p\left(x|y\right) &= \frac{p\left(y|x\right)p\left(x\right)}{p\left(y\right)} \\ &= \frac{p\left(y|x\right)p\left(x\right)}{\int p\left(y|x\right)p\left(x\right)dx} \end{aligned} \tag{1.1}$$

The Bayesian approach [12] involves estimation of unknown "states" from a set of observations. Generally, one has prior knowledge of the system being modelled which can be formulated, in a Bayesian framework, as a prior distribution. Then using the mathematical model that one has to approximate the physical phenomena being observed, one may obtain the likelihood which relates the prior knowledge to the observations. This is then used to construct the posterior distribution for the "state" of the system given the observation sequence obtained.

In this Bayesian framework the unknown parameters are treated as random variables and their prior distribution is updated via Bayes theorem to provide the posterior distribution which is conditioned on the set of observations. Then all inference that is of interest is carried out with the aim being to obtain estimates of the posterior probability of a state given the observations. It is also important to mention that much literature has been devoted to understanding how one can sensibly assign prior probability distributions and what they mean in different contexts. There is a multitude of references available on this topic and the author recommends; [19], [15],[48],[75].

Bayesian Parameter Estimation

As stated, the focus of Bayesian analysis revolves around the posterior distribution. However, when one wants to perform parameter estimation, the value of the unknown parameter vector x can be estimated in several ways. The two most common methods used to obtain a parameter estimate from the posterior distribution of interest are the Maximum A Posteriori (MAP) criterion and the Minimum Mean Square Error (MMSE) or minimum variance estimator [79].

The MAP estimate depends on the likelihood function weighted by the prior probability and is given as follows,

$$\widehat{x}_{MAP} = \arg\max_{x} p\left(x|y\right).$$

The MMSE estimate is given by,

$$\widehat{x}_{MMSE} = \int xp\left(x|y\right)dx.$$

Bayesian Model Selection

There are three broad approaches to understanding model selection which are labelled, according to [15], [44] as the $\mathcal{M}_{open}, \mathcal{M}_{completed}$ and the $\mathcal{M}_{closed}$ modelling perspectives. The $\mathcal{M}_{closed}$ approach takes the view that the class of models under consideration contains the true model. The $\mathcal{M}_{completed}$ view corresponds to the case where although a formulated belief model is known, due to intractability of analysis other models are considered. The $\mathcal{M}_{open}$ approach takes the view that none of the models under consideration completely captures the intricate relationship between the inputs and the outputs, [34] page 24. Hence, the $\mathcal{M}_{completed}$ and $\mathcal{M}_{open}$ approaches places prior probabilities on each model which reflect the relative degree of belief in each model, for the class of models being considered. All of these approaches lend themselves to a Bayesian analysis. A key aspect of Bayesian model selection is that one can improve the quality of the model

selected through the introduction of prior quantitative or qualitative knowledge. This is achieved by assigning prior distributions to the model parameters and then updating these parameters in light of the observations. Bayesian model selection involves the selection of the model, usually from a finite set of possible models denoted $\{M_i\}$, which most accurately represents the observations, according to some criterion of interest. Hence Bayesian model selection can be considered as the process of determining the most plausible model for the data given the set of possible models to choose from. The Bayesian model selection approach is,

$$\begin{aligned} p(M_i|y) &= \frac{p(y|M_i)\,p(M_i)}{p(y)} \\ &= \frac{\int p(y|x, M_i)\,p(x|M_i)\,p(M_i)\,dx}{\sum_j \int p(y|x, M_j)\,p(x|M_j)\,p(M_j)\,dx}. \end{aligned}$$

It is important to mention that one should not forget that the outcome obtained from the above analysis results in a distribution and hence reflects a probability of each given possible model choice; in the continuous range of models scenario one will obtain a density. Hence one can decide to either perform subsequent evaluations using weighted model averages or to use a point estimate such as a MAP estimate. A discussion of the pros and cons of this method are presented lucidly in [83],[72],[71]. A very insightful discussion of the merits of both open and closed perspectives of model selection, in terms of Bayes factors or loss functions, is presented in [34].

It is important to mention that model selection can be computationally challenging. Often an exhaustive search of the model space to determine the best model for a given situation proves to be a massive computational effort and is therefore infeasible or impractical. For this reason many techniques have been developed to aid in the search for the optimal model, these include Greedy searches, Leaps and bounds, EM algorithm [33], Simulated Annealing [57] and Genetic algorithms. This thesis will also present a new technique to perform model selection, which efficiently explores the model space to find the optimum model.

1.3 Definitions and Notation

This section shall be used to introduce some notation which will be used throughout the thesis. It shall be assumed that a random variable X can be defined on a probability space of the form $(E, \mathcal{E}, \mathbb{P})$. Where, E will represent the space of all outcomes which may be either discrete or continuous and may be of multiple dimension, but will always be real. $\mathcal{E}$ will represent $\sigma(E)$ which is the sigma algebra generated by the space E, which is the set of all possible outcomes and $\mathbb{P}$ will be a probability measure on the space E. The notation $\pi(dx)$ shall be used to represent the law or distribution of the random variable X, which is a probability measure given by the image measure on the space in question. One may then assume that given the law of the random variable X, one can define a Radon-Nikodym derivative with respect to the dominating measure dx. This is equivalent to stating that $\pi(dx)$ admits as a density $\pi(x)$ with respect to dominating measure dx. Additionally, the following notation was used throughout this thesis, where for any probability density π and sequence of transition kernels $\{K_s\}$,

$$\pi K_{i:j}(x_j) \triangleq \int \pi(x_{i-1}) \prod_{s=i}^{j} K_s(x_{s-1}, x_s)\, dx_{i-1:j-1}.$$

In terms of notation it shall also be assumed that the un-normalised version of the density $\pi(x)$ is given by $f(x)$. In all models considered in this thesis the spaces of interest will be either discrete or continuous, open or compact subsets of Euclidean space. Furthermore, it shall be assumed that all distributions of interest admit densities with respect to either the counting measure or Lebesgue measure. Having established this notation which shall be used throughout the thesis, it is now important to highlight the aims of the thesis and how the thesis will be structured.

1.4 General Aim of Analysis

The general aim of the analysis contained in this thesis can be summarised as trying to estimate a posterior distribution, given a noisy observation sequence. This estimation can either involve receiving sequential observations and carrying out updates to the posterior in light of the new observations, or batch scenarios in which all the observations are available and the aim is to estimate the posterior conditional on knowledge of the batch of noisy observations. Markovian, non-linear and non-Gaussian signals will be considered in this thesis. Systems which are linear and Gaussian are not of particular interest, in the sense that they have an optimal solution known as the Kalman filter, which is well studied and widely implemented in practice.

1.5 Structure of Thesis

To a certain extent, each chapter in this thesis may be read independently as they are fairly self contained. However, the chapters do lead into one another, in the sense that each chapter builds on previous chapters. The second chapter provides a literature review, which motivates the reason for developing the new methodology forming the body of this thesis. The third chapter develops the fundamental framework and provides guidelines for the use of the new methodology developed in this thesis, termed Sequential Monte Carlo Samplers (SMC Samplers). Chapter three also provides an application and then the results of simulations obtained using SMC Samplers methodology are compared to existing algorithms in the literature.

The fourth chapter extends the Sequential Monte Carlo Samplers methodology to provide a new framework for trans-dimensional analysis, which is termed Trans-Dimensional Sequential Monte Carlo (TDSMC). Again, applications are provided with comparison to existing techniques. The fifth chapter is devoted to two detailed applications, the estimation of an inhomogeneous Poisson Process rate function and secondly, basis function regression for the General Linear Model. These applications provide the reader with

a detailed account of how to implement the new algorithms for TDSMC in real world problems. Simulation results which demonstrate how effective the new algorithms are in comparison to existing techniques and algorithms are also provided. The final chapter is conclusions.

Chapter 2: Monte Carlo Methods

This chapter will provide a literature review of Monte Carlo methods and the justification for new methods, which are not as computationally constrained as the standard classical Monte Carlo approaches which utilise sampling techniques such as inversion sampling and rejection sampling. The sampling techniques presented include, batch sampling algorithms such as Metropolis-Hastings algorithm, the Gibbs sampler and the methodology of Reversible Jump Markov Chain Monte Carlo to carry out trans-dimensional analysis. In the sequential setting the basics of Importance Sampling is presented followed by the methodology of Sequential Monte Carlo.

Chapter 3: Sequential Monte Carlo Samplers

Initially, this chapter provides justification for developing the new SMC Samplers methodology, whilst motivating its use in several situations which include; utilising SMC in situations typically associated with MCMC, optimisation and moving from an easy to sample distribution to a difficult distribution, through a sequence of intermediate distributions. Guidelines are provided to aid effective implementation of SMC Samplers algorithms, along with theoretical analysis to support these algorithmic guidelines. Finally, an example is presented which deals with Bayesian variable selection. This example provides comparison to existing algorithms such as Annealed Importance Sampling, MCMC, parallel MCMC and Simulated Annealing.

Chapter 4: Trans-Dimensional Sequential Monte Carlo

This chapter presents an extension of the ideas developed in the previous chapter. The TDSMC algorithm is developed and motivated through analogy to RJMCMC. In this respect it is demonstrated that TDSMC is to SMC Samplers methodology, what RJMCMC is to MCMC methodology. Then guidelines are presented for efficient applica-

tion of TDSMC. Finally, a comparison between existing algorithms such as SVM, RVM and MCMC is made through application of TDSMC to two real data sets, which are considered bench-marks for comparison of algorithms. This application involves radial basis function regression.

Chapter 5: Applications

This chapter is dedicated solely to developing two detailed applications of the TDSMC algorithm. The first application involves the estimation of an inhomogeneous Poisson Process rate function, using a simple piecewise linear function approximation. Several examples are presented, culminating in application of the new TDSMC algorithm to the coal mining disasters between 1851-1962 data set. This allows for a comparison between the RJMCMC algorithm and the TDSMC algorithm, in a batch data scenario. The second application involves basis function regression for the General Linear Model. A generic algorithm is developed which includes developing different types of "moves" which are very general and may be applied in a range of situations. The application presented here involves estimation of parameters of exponential basis functions, in a scenario in which noisy observations are arriving sequentially.

Chapter 6: Discussions, Conclusions and Future Research

A summary of the work undertaken and ideas for future research are presented.

Chapter 2

Monte Carlo Methods

2.1 Introduction

This chapter provides an overview of several methodologies which have been developed to produce samples from a target distribution, $\pi(dx)$. This has been the focus of a significant amount of scientific research for the past few decades and in the context of this thesis the general aim of the analysis can be summarised as trying to estimate a target posterior distribution given a noisy observation sequence. In this Bayesian inference setting the target distribution $\pi(dx)$ takes the form of a posterior distribution $p(dx|y)$. The Bayesian inference approach used in this thesis requires the ability to simulate from the posterior distribution of interest. Several techniques have been developed to obtain samples or realisations of random variable X which is distributed according to $\pi(dx)$. This is a significant problem in many fields of research as the samples obtained may have several applications as will be explained. One of the fundamental uses of these samples is to help characterise the distribution, $\pi(dx)$, through empirical estimates of the moments and sufficient statistics. Another, very important use of samples drawn from a distribution, which has spawned several fields of research, involves the casting of difficult integrals which are in high dimensional spaces in the form of expectations with respect to the distribution, $\pi(dx)$. This plays a particularly significant role in Bayesian

inference. Other significant uses of these samples involve; optimization and obtaining estimates of solutions to many inference problems which are contained in the fields of electrical engineering, communications, control, bioinformatics and finance, to mention a few.

In the Bayesian framework presented previously, one generally requires the ability to solve multidimensional integrals to determine things such as the model evidence, or the marginal posterior distributions in situations such as filtering recursions or smoothing, expectations and moments with respect to some known function and the removal of nuisance parameters. These integrals are generally difficult and may not have tractable closed form solutions, hence there was a strong need to develop simulation techniques to approximate the solutions of these integrals. Classical numerical integration techniques, such as Gaussian Quadrature and Simpsons rule, are fine in low dimensions, however as the integrals become more complicated and higher dimensional the computational requirements of such techniques rapidly becomes too costly for these techniques to be viable [79]. This is especially bad for "on-line" applications in which the integrals are solved progressively in time, as new data or observations are recorded. The success of Monte Carlo techniques stems from the fact that unlike the classical techniques mentioned above which require a grid of points, the Monte Carlo techniques to be discussed do not have this dimensional constraint. That is there is no *direct dependence* between computational requirements and dimension in Monte Carlo integration [37]. Obviously as the dimension grows, the number of samples required will need to be increased, and one can also not typically say how many samples would be required for a given problem in a given dimension.

2.2 Monte Carlo Methods

The power of Monte Carlo techniques to solve high dimensional integrals has been utilised extensively throughout many fields. The reason why these techniques have been so suc-

cessful is that they are not subject to any constraints on linearity or Gaussianity and hence prove to be very general in nature. The importance of the method lies in the fact that one may consider difficult integrals as expectations, and thus may draw samples from the distribution with respect to which the expectation is defined and compute an approximation of the integral as a sample average. Furthermore, convergence results for several key classes of Monte Carlo approximation techniques have been studied and are now well understood. This allows one to optimise Monte Carlo techniques and places them on a sound mathematical footing, which enables practitioners to be confident that the results obtained through application are mathematically consistent, logical and reproducible.

The basic idea behind Monte Carlo methods is that any probability measure, π, defined with respect to a measurable space, $(E, \mathcal{E})$, can be approximated using the following empirical measure:

$$\pi^N(dx) = \frac{1}{N}\sum_{i=1}^{N}\delta_{X^{(i)}}(dx)$$

where, $\left\{X^{(i)}\right\}_{i=1:N}$, is a sequence of N i.i.d. samples of law, π, and one assumes $\pi(dx)$ admits a density with respect to Lebesgue measure denoted $\pi(x)$.

This approximation has led to wide-spread use of Monte Carlo techniques, specifically with respect to approximating difficult integrals. The classical approach to Monte Carlo integration can be understood by looking at the generic problem shown below, where one requires a solution to the integral.

$$\mathbb{E}_\pi[\varphi(X)] = \int_E \varphi(x)\,\pi(x)\,dx$$

In what is known as "Perfect Monte Carlo Sampling", one can generate samples, $\left(X^{(1)}, ..., X^{(N)}\right)$, from the density, $\pi(x)$, using some technique such as rejection sampling, inversion sampling or a technique such as Box Muller. Then these samples may be used to obtain an empirical average, which can be used as an approximation to the solution

of the integral in question,

$$\overline{\varphi}_N = \frac{1}{N}\sum_{i=1}^{N}\varphi\left(X^{(i)}\right).$$

Then, applying the Strong Law of Large Numbers (SLLN), it can be seen that $\overline{\varphi}_N$ converges almost surely (π-a.s.) to $\mathbb{E}_\pi\left[\varphi\left(X\right)\right]$, for a suitable class of functions. The second thing to note is that when the second moment is finite, then not only is it known that a.s. convergence applies but one can also obtain a rate of convergence of $\overline{\varphi}_N$ to $\mathbb{E}_\pi\left[\varphi\left(X\right)\right]$, assuming that φ is an element of the class of square integrable functions. This rate of convergence is obtained by estimating the variance using the generated samples, $\left(X^{(1)},....,X^{(N)}\right)$, as follows,

$$V_N = \frac{1}{N^2}\sum_{i=1}^{N}\left[\varphi\left(X^{(i)}\right) - \overline{\varphi}_N\right]^2$$

Combining this information one may invoke the Central Limit Theorem (CLT) to determine that $\frac{\overline{\varphi}_N - \mathbb{E}_\pi[\varphi(X)]}{\sqrt{V_N}} \sim N\left(0,1\right)$. This has the advantage that now one may obtain confidence bounds on the estimator, $\overline{\varphi}_N$. Furthermore, the rate of convergence is clearly independent of the dimension of the integrand. It is important to realise that this all relies on the fact that it is possible and not too computationally difficult to obtain samples from the distribution of interest, $\pi(x)$.

There is a large literature on methods for simulating from a target distribution, $\pi\left(dx\right)$. The most basic of these techniques involves uniform random variate generation followed by inverse transforms, there is a long list of these general transform techniques which include methods such as Box-Muller. Many of these techniques are thoroughly detailed in the two excellent texts [35] and [75].

However, most distributions which are of importance in modelling real world systems are too complicated to obtain samples from using direct inversion techniques as they may be multi-variate, non-standard and only known up to proportionality. In these cases, one may attempt another class of simulation techniques known as Accept-Reject, which only

requires a knowledge of the functional form of the density of interest, up to normalisation. Given a target density of interest $\pi(x) \propto f(x)$ and a density $q(x) \propto g(x)$ which is easier to simulate from than the target density, the first requirement is to determine a constant M such that

$$f(x) \leq Mg(x)$$

is true on the support of $f(x)$, [74]. The Accept-Reject algorithm then proceeds as shown below,[76] page 49.

Accept-Reject Algorithm

- 1. Generate $X \sim g$, $U \sim \mathcal{U}[0,1]$
  2. Accept $Z = X$ if $U \leq f(x)/Mg(x)$
  3. Return to 1. otherwise

The proof of this procedure for obtaining samples form the target distribution of interest is very simple. The distribution of Z is given by,

$$\begin{aligned}
\Pr(Z \leq z) &= \Pr\left(X \leq z | U \leq \frac{f(x)}{Mg(x)}\right) = \frac{\Pr\left(X \leq z, U \leq \frac{f(x)}{Mg(x)}\right)}{\Pr\left(U \leq \frac{f(x)}{Mg(x)}\right)} \\
&= \frac{\int_{-\infty}^{z} \int_{0}^{f(x)/Mg(x)} du\, g(x)\, dx}{\int_{-\infty}^{\infty} \int_{0}^{f(x)/Mg(x)} du\, g(x)\, dx} = \frac{\frac{1}{M}\int_{-\infty}^{z} f(x)\, dx}{\frac{1}{M}\int_{-\infty}^{\infty} f(x)\, dx} = \int_{-\infty}^{z} f(x)\, dx
\end{aligned}$$

which proves the required result. These techniques are widely studied and it is not the aim of this thesis to provide a detailed review of these fundamental techniques, the interested reader is referred to [75] and [18].

In situations where the described techniques either fail or become too computationally intensive, researchers have developed other techniques to utilise the framework of Monte Carlo simulation. The first of these techniques to be discussed will be Markov Chain Monte Carlo (MCMC) techniques.

2.3 Markov Chain Monte Carlo Methods

In order to overcome the problems discussed, with regard to difficulties in obtaining samples from the required target distribution. The MCMC approach constructs an ergodic Markov Chain, $\{X_1, ..., X_N\}$, taking values in a measurable space, E. This Markov chain is constructed to have the property that it has a limiting, invariant distribution, which is the target distribution of interest, $\pi(dx)$. This invariant distribution is the target distribution that we require samples from in order to calculate Monte Carlo estimates. Now for the Markov Chain samples to be used as samples from the target distribution, it is necessary that there exists a unique invariant distribution which is the target distribution and that the Markov Chain is ergodic. The requirement of ergodicity, in the most simple situation of the discrete state space, effectively requires that there is a single non-empty closed class which is aperiodic and that there exists a state, j_0, such that the expected recurrence time, $\mathbb{E}\tau_{j_0}$, is finite. An excellent review of the properties of more general state space Markov chain theory and the analogous definitions can be found in the following references [67], [46], [75]. It should also be mentioned that a lot of work has been focused on reversible chains that satisfy the condition shown in(2.1), where $\pi(dx)$ is the stationary distribution and $K(x, z)$ the transition kernel.

$$\pi(dx) K(x, dz) = \pi(dz) K(z, dx) \tag{2.1}$$

When these sufficient conditions are satisfied one can use the Markov Chain iterations, $\{X_1, ..., X_N\}$, in the Monte Carlo integral to obtain the estimator $\overline{\varphi}_N$. This estimate can be considered as an ergodic average and convergence to the required expectation is ensured by the ergodic theorem. A technical discussion on some of the properties of this convergence is found in Roberts and Tierney's sections of [46]. There are several methods of constructing a Markov Chain which has as its stationary distribution the required target distribution, however they are all special cases of the general framework established by Metropolis and Hastings [46]. The two methods that will be presented in

this chapter are the Metropolis-Hastings algorithm and the Gibbs sampler.

It is worth drawing to the attention of the reader that, although a lot of research has been carried out using reversible Markov chains, recent work by Diaconis, Holmes and Neal [36], [68] has focused on non-reversible chains. It has been shown that a reversible Markov chain on a finite state space, that is irreducible, can be used to construct a non-reversible Markov chain. When estimates are carried out using the samples from the non-reversible Markov chain, the variance of the estimate has been proven to be at least as low as that obtained with the reversible chain. This suggests potential for more exploration and it would be interesting to consider more general state spaces such as continuous state spaces to see if these results still hold.

2.3.1 Metropolis-Hastings

The Metropolis-Hastings algorithm was first developed by Metropolis *et al.* (1953) [66] and then later extended by Hastings (1970) [55]. The Metropolis-Hastings algorithm has a proposal distribution, $q(x_t, .)$, which conditioned on the current state, is used to sample a proposed new state, Z_{t+1}, for the Markov chain. Then this proposed new state along with the current state of the Markov chain, X_t, are used to calculate an acceptance probability. The acceptance probability is the probability of whether the Markov chain makes a state transition to the new sampled state, otherwise the Markov chain remains in the state it was in at the previous iteration. Hence, this acceptance probability is crucial as it ensures that the Markov chain that is being constructed will have the required stationary distribution. For clarity, the reader is reminded that the notation X_t will be used to represent the state of the Markov chain at time t, and will in general be a vector.

It should also be mentioned that the Metropolis-Hastings algorithm allows one to construct a Markov chain which is free to make moves in any direction and to anywhere in the state space, defined by the support of the target distribution of interest. Additionally, when calculating a Monte Carlo ergodic average estimate for the integral, one would like the variance of the estimate to be as small as possible. One way of helping to ensure

that the variance of the estimate is kept to a minimum is to use only samples from the Markov chain created, which one is fairly confident come from the Markov chain once it has reached its stationary regime. This is achieved in most simulations by discarding a certain number of initial samples known as "burn in" samples. For a detailed discussion of these ideas there are several good references, the author directs the reader to [46], [75], [84], [25], [66], [55], [63].

Metropolis-Hastings Algorithm

- Initialisation : $t = 0$, $X_0 = x_0$
- For $t = 1 : N$

  1. Draw proposal $Z_{t+1}$from proposal distribution $q(x_t, .)$
  2. Evaluate the acceptance probability :
     $\alpha(x_t, z_{t+1}) = \min\left(1, \frac{\pi(z_{t+1})q(z_{t+1}, x_t)}{\pi(x_t)q(x_t, z_{t+1})}\right)$
  3. Sample random variate $U \sim \mathcal{U}[0, 1]$
  4. If $U \leq \alpha(X_t, Z_{t+1})$

     $X_{t+1} = Z_{t+1}$

     else

     $X_{t+1} = X_t$

     end

The Markov chain created by this algorithm is reversible and has the required target distribution, $p(x)$. The transition kernel of the Markov chain created in the Metropolis-Hastings algorithm has form,

$$\begin{aligned} K(x_t, dx_{t+1}) &= q(x_t, dx_{t+1})\,\alpha(x_t, x_{t+1}) \\ &\quad + \left[1 - \int q(x_t, z)\,\alpha(x_t, z)\,dz\right] \mathbb{I}(x_{t+1} = x_t). \end{aligned}$$

Where $\mathbb{I}(.)$ is the indicator function, the first term represents the acceptance probability of the proposed state and the second term represents the rejection probability of the proposed state. The choice of proposal distribution is very general, however blind selection can lead to slow mixing of the chain and long burn in times. This will be reflected in the acceptance probability ratio. For example, for the Gaussian random walk proposal it has been shown that ideally one should use acceptance probabilities between 15% and 50% [78], [77] as a general guide. There have been a few studies for optimal acceptance rates using different types of proposal distribution in different dimensions, a summary of these may be found in [46] on page 55. This will ensure that the chain is not proposing steps which are too large, hence rejecting lots of moves and also not steps which are too small and hence accepting most moves, but exploring the state space very slowly.

There have been many versions of this algorithm developed all of which have different properties with respect to the manner in which the Markov chain created explores the state space. The most commonly known algorithms include the Metropolis algorithm, which has only symmetric proposal distributions [66], the Independence sampler [46], Random walk Metropolis [46], Configurational Bias Monte Carlo [80], Multiple Try Metropolis [65] and the single component Metropolis-Hastings algorithm. The single component Metropolis-Hastings algorithm is so named since it does not update every component of the state vector X_t in a block at each iteration. Instead it is more convenient and computationally efficient to divide X_t into sub components, of possibly differing dimension and then update them one by one. This can be done either one element at a time or larger sub-blocks can be used. The other consideration is that the ordering of which components should be updated can be decided randomly or deterministically. If the deterministic scan sampler is used, which consists of say d consecutive reversible components, it is important to keep in mind that although each component is reversible the over all sampler is not reversible. A method of obtaining a reversible sampler would be to use the random scan, as discussed in [46] on page 51, where the component block to be updated at each iteration is determined randomly.

2.3.2 Gibbs Sampling

Gibbs sampling is the most widely used form of the single component Metropolis-Hastings algorithm. It involves sampling from full conditional distributions shown below,

$$\pi\left(x_i|x_{-i}\right) = \frac{\pi\left(x\right)}{\int \pi\left(x\right) dx_i}.$$

Due to the construction of the Gibbs importance distribution, the acceptance probability of a proposed new state for the Markov chain being simulated is always identically one. It is important to understand the nature of the moves that are possible for any given Markov chain construction method, since the types of move possible will affect the rate at which the Markov chain mixes. This ultimately has effects on factors such as the chain length required for the "burn in" stage, and the variance and validity of the estimate obtained using the Markov chain variates in the Monte Carlo approximation. The Gibbs sampler only permits moves which, at any given time t, are parallel to the axis of the component of the state which is to be updated. As mentioned this can affect the ability of the Gibbs sampler to explore the state space thoroughly. For an in depth discussion of these factors the reader is referred to [46],[75], [63].

Gibbs Sampling Algorithm

- Initialisation :$t = 0$, $X_0 = x_0$
- For $t = 1 : N$
  - Iterate from $s = 1 : p$ where p is the number of sub-blocks

    Sample $X_{s,t} \sim \pi\left(.|x_{-s,t}\right)$ where

$$X_{-s,t} = \{X_{1,t}, ..., X_{s-1,t}, X_{s+1,t-1}, ...X_{p,t-1}\}$$

Now the transition kernel for the Gibbs sampler is given by the following expression,

$$K\left(x_t, x_{t+1}\right) = \prod_{k=1}^{p} \pi\left(x_{k,t+1} | x_{-k,t+1}\right).$$

For an in-depth discussion on the finer details of convergence and other properties of Markov Chains the reader is referred to [67].

Finally another important methodology, that is of key relevance to this thesis, is Reversible Jump Markov Chain Monte Carlo (RJMCMC) which was first introduced in its current form by Green [52]. However, earlier work by Grenander and Miller [53] presented an algorithm for continuous time models which they termed jump-diffusion. For the purpose of this report the author will be most interested in thinking about RJMCMC as methodology to deal with problems which are trans-dimensional in nature. For an in-depth discussion and measure theoretic presentation of RJMCMC methodology the reader is directed to [52], [51] and [91].

2.3.3 Reversible Jump Markov Chain Monte Carlo

When one wants to carry out Bayesian analysis in a situation where there is a range of models which have parameter spaces of differing dimensionality, it is usual to account for the model uncertainty by assigning a prior distribution over the collection of competing models. In such situations the posterior distribution over the unknown models and model parameters, cannot be analysed using the standard Metropolis-Hastings framework. The difficulty that arises when trying to perform model selection on such general state spaces, which include the model indicator and each models parameters, is that it no longer makes sense to consider ratios of densities in the acceptance probability which have support in different dimensions. RJMCMC solves this problem by extending the basic Metropolis-Hastings algorithm to these general state spaces. That is, RJMCMC methodology is designed to create a Markov chain which has as its invariant distribution a posterior distribution which takes its support on such general state spaces. This extension means

that now one must work with a target probability measure, $\pi(dx)$, and a proposal kernel, $q(x, dz)$, since comparing densities in different dimensions has no real meaning. So, in working with distributions instead of densities, it is possible to ensure that we only make comparisons under the same volume measure, which as stated earlier, we assume to be either counting measure or Lebesgue. Hence, now the acceptance probability will contain the ratio of densities and the ratio of the measures, leading to an additional Jacobian term in the formulation of the acceptance expression.

The next idea of Green [52] was to realise that instead of doing the model search in the full product space that would arise if one sampled over the model indicator and the parameters. Alternatively, one could focus on disjoint union spaces of the form $\{(k, x_k)\} = \cup_{k\in\mathcal{K}} (\{k\} \times \mathcal{X}_k)$ and the target distribution defined on such a space is then given by,

$$\pi(k, dx) = \sum_{m=1}^{M} \pi(m, dx_m) \mathbb{I}_{\{m\}\times\mathcal{X}_m}(k, x)$$

where M is the family of models and $x_m \in \mathcal{X}_m$ are the model dependent parameters.

So RJMCMC, in Green's formulation, allows the Markov chain to explore within the sub-spaces and also jump between the sub-spaces, say from $\mathcal{X}_m$ to $\mathcal{X}_n$. It is important to mention that to allow this behaviour one must extend the spaces to $\overline{\mathcal{X}}_{m,n} \triangleq \mathcal{X}_m \times \mathcal{U}_{m,n}$ and $\overline{\mathcal{X}}_{n,m} \triangleq \mathcal{X}_n \times \mathcal{U}_{n,m}$ and also define a deterministic diffeomorphism, dimension matching function between these extended spaces, labelled h_{nm}. Borrowing the notation of [2], this basically means that the user must define the proposal distributions $q_{mn}(.|m, x_m)$ and $q_{nm}(.|n, x_n)$ which go from (n, x_n) to (m, x_m) and back again, the extended state spaces $\overline{\mathcal{X}}_{m,n}$ and $\overline{\mathcal{X}}_{n,m}$ and the deterministic transform between these spaces h_{mn}. Now as explained in [2], in a move which goes from (n, x_n) to (m, x_m) one must first generate $u_{n,m} \sim q_{nm}(.|n, x_n)$ and then evaluate $(x_m, u_{m,n}) = h_{nm}(x_n, u_{n,m})$ where the notation $x_m^* = h_{nm}^x(x_n, u_{n,m})$ is used for the x_m component of the function h_{nm}. This move will then be accepted according to the following acceptance probability of a dimension

changing move as shown in equation (2.2) below,

$$\min\left\{1, \frac{\pi\left(m, x_m^*\right) q\left(n|m\right) q_{mn}\left(u_{m,n}|m, x_m^*\right)}{\pi\left(n, x_n\right) q\left(m|n\right) q_{nm}\left(u_{n,m}|n, x_n\right)} \left|\det \frac{\partial h_{nm}\left(x_m, u_{m,n}\right)}{\partial\left(x_m, u_{m,n}\right)}\right|\right\} \tag{2.2}$$

where the term $\left|\det \frac{\partial h_{nm}(x_m,u_{m,n})}{\partial(x_m,u_{m,n})}\right|$ is the Jacobian of the function h_{nm}. The Jacobian term in RJMCMC is an important part of the analysis of RJMCMC and hence warrants a brief discussion. The dimension changing move, performed by the function h_{nm}, must obey the change of variables theorem. This theorem effectively describes how volumes are distorted by differentiable functions. The change of variables theorem reduces the problem of determining the distortion of the volume to understanding the infinitesimal distortion given by the linear map's determinant. Hence, if S is any subset of $\mathbb{R}^n$ and the move involves a function $h_{nm} : \mathbb{R}^n \to \mathbb{R}^m$, then the volume of h_{nm}'s image is given by $\left|\det \frac{\partial h_{nm}(x_m,u_{m,n})}{\partial(x_m,u_{m,n})}\right|$ times the original volume, which explains why the Jacobian now appears in the Metropolis-Hastings ratio. Further examples of how to formulate different types of moves may be found in [51], [52], [2], [20].

The big restriction in this methodology is that the trans-dimensional moves that are made must be reversible in nature. This means that the Metropolis-Hastings type proposal moves between dimensions must have an acceptance probability which preserves detailed balance or equivalently reversibility. Note there has been some discussion about the Jacobian term which is required for the RJMCMC methodology, however the author points out that when one proposes moves directly in the new parameter space as opposed to dimensional matching of random variables, then the Jacobian term is unity in the acceptance probability expression.

RJMCMC is applied in; mixture modelling where the number of mixture components is unknown, the number of splines in a multi-variate adaptive splines regression (MARS) model, non-parametric Bayesian smoothing, linear regression with varying number of covariates and finite point processes, for more details see [91]. Further references which provide detailed insight into RJMCMC are; [52], [91], [8], [7], [2],[20], [34].

2.4 Importance Sampling

As has been discussed, the ability to estimate integrals using a collection of random samples is very important. Importance sampling avoids the problem of trying to sample directly from the target distribution by instead sampling from an importance distribution, $q(x)$, which is selected to have the property that it is simpler to obtain samples from than the target distribution. Then one must correct for the fact that these samples were not taken from the distribution of interest, $\pi(x)$, but instead from the importance distribution, $q(x)$. This correction step is known as importance weighting. Together these steps produce the point mass representation of the target distribution presented previously. Integrals of some bounded, integrable test function, φ, with respect to the target distribution,

$$\mathbb{E}_{\pi}[\varphi(X)] = \int \varphi(x)\pi(x)\,dx = \int \frac{\varphi(x)\pi(x)}{q(x)} q(x)\,dx = \mathbb{E}_q\left[\varphi(X)\frac{\pi(X)}{q(X)}\right]$$

may then be approximated as

$$\widehat{I_q^{N*}}(\varphi) = \frac{1}{N}\sum_{i=1}^{N} W^*\left(X^{(i)}\right)\varphi\left(X^{(i)}\right) \tag{2.3}$$

where the importance weight is $W^*(x) = \pi(x)/q(x)$, and the particles, $X^{(i)}$, are samples from the importance distribution, $q(x)$. This will produce an unbiased estimate since

$$\mathbb{E}\left[\widehat{I_q^{N*}}(\varphi W^*)\right] = I_q(\varphi W^*) = I_{\pi}(\varphi)$$

and the variance of the estimate will be inversely proportional to the number of particles N.

Importance sampling is performed as demonstrated below. Version one demonstrates importance sampling in which the target distribution can be evaluated point wise and version two demonstrates the situation in which the target distribution can only be evaluated pointwise *up to a normalising constant*, which occurs most frequently in applications.

Importance Sampling IS: (version 1)

- e.g. $\pi(x)$ is difficult to sample, yet can be evaluated analytically
- $\left\{X^{(i)}\right\}_{i=1:N} \sim q(x)$ samples easily generated from Importance density $q(x)$
- $W^{*(i)} = \frac{\pi\left(X^{(i)}\right)}{q\left(X^{(i)}\right)}$ samples are weighted
- $\pi(x) = \sum_{i=1}^{N} W^{*(i)} \delta_{X^{(i)}}(x)$ particle representation of the target density

---

Importance Sampling IS: (version 2)

- e.g. $\pi(x) \propto f(x)$ is difficult to sample, yet can be evaluated analytically up to a normalisation constant
- $\left\{X^{(i)}\right\}_{i=1:N} \sim q(x)$
- $w^{(i)} = \frac{f\left(X^{(i)}\right)}{q\left(X^{(i)}\right)}$

  the samples are weighted and $w^{(i)}$ is the un-normalised weight and $W^{(i)}$ is the normalised weight
- $\pi(x) = \sum_{i=1}^{N} W^{(i)} \delta_{X^{(i)}}(x)$ particle representation of target density

  where $W^{(i)} = \frac{w^{(i)}}{\sum_{j=1}^{N} w^{(j)}}$

When version two of the importance sampling techniques is used then the normalised weights obtained provide an estimate of the "true" importance weights [42], as shown below

$$\widehat{W}^{*(i)} = NW^{(i)}.$$

It should be noted that when one uses importance weights given by $W^{(i)}$ to estimate an integral then the solution will be biased as a result of taking the ratio of estimates. However, it has been shown that under mild assumptions the SLLN yields asymptotic convergence of the estimate formed using these importance weights to the true solution, [42]. Importance sampling can now be used to develop the more widely used sequential version known as Sequential Importance Sampling (SIS).

2.5 Sequential Monte Carlo Methods

In many applications one is interested in "on-line" or sequential data analysis, for this reason much effort has been devoted to developing Sequential Monte Carlo methods, otherwise known as Particle Filters. Generally, in non-linear filtering one is interested in calculating the solution of a non-linear system which takes values in a space of probability measures. The problems encountered usually require one to resort to using numerical approximations in the form of interacting particle systems. Particle filters have developed from this methodology and allow one to approximate distributions, sequentially in time, using point-masses. The sequence of probability distributions that are being approximated shall be denoted $\{\widetilde{\pi}_t\}_{t\geq 1}$. These distributions shall be defined to take support on $\{E_t\}_{t\geq 1}$ such that $\dim(E_t) < \dim(E_{t+1})$; e.g. $E_1 = E$, $E_t = E^t$ and $\widetilde{\pi}_t(dx_{1:t}) = \widetilde{\pi}_t(x_{1:t})\, dx_{1:t}$ where each $\widetilde{\pi}_t(x_{1:t})$ is known up to a normalizing constant, i.e.

$$\widetilde{\pi}_t(x_{1:t}) = \underbrace{Z_t^{-1}}_{\text{unknown}} \cdot \underbrace{f_t(x_{1:t})}_{\text{known}} \text{ where } x_{1:t} \triangleq (x_1, x_2, \ldots, x_t).$$

This situation arises, for example when one is interested in the sequence of posterior distributions which are formed when updating a posterior distribution in light of new observations, arriving sequentially in time.

Essentially the SMC principle is to approximate each distribution $\widetilde{\pi}_t$ by a weighted sum of random samples/particles $\left\{X_{1:t}^{(i)}, W_t^{(i)}\right\}$ ($W_t^{(i)} > 0$, $\sum_{i=1}^N W_t^{(i)} = 1$); i.e.

$$\widehat{\widetilde{\pi}}_t\left(dx_{1:t}\right) = \sum_{i=1}^N W_t^{(i)} \delta_{X_{1:t}^{(i)}}\left(dx_{1:t}\right) \text{ and } \widehat{\widetilde{\pi}}_t \to \widetilde{\pi}_t \text{ as } N \to \infty$$

This approximation is carried out sequentially by first sampling from $\widetilde{\pi}_1$ then $\widetilde{\pi}_2$ and so on.

Using this approximate representation of the target distribution clearly has advantages as it allows computations of integrals to be carried out easily using the sampling property of the Dirac delta mass. The weights present in the above expression are chosen using the principle of Importance Sampling (IS). As mentioned earlier, it is often very difficult to generate samples from the target distribution using standard techniques. Alternatives for generating such samples from the target distribution, in batch scenarios were presented at the start of this chapter. This section is now going to present sequential sampling methods, on which there is much literature. The reader is directed towards the following far from comprehensive selection of papers and books for further details [41], [37], [63], [64], [24], [30], [70], [11]. For convergence results and Central Limit Theorems relevant to this rich class of algorithms and methodology, the following papers provide excellent insight into the field [32],[27], [62], [39], [29].

2.5.1 Sequential Importance Sampling, Resampling and MCMC Diversification Move

The generic Sequential Importance Sampling situation can now be derived as follows. At time $t-1$, assume a set of weighted particles $\left\{W_{t-1}^{(i)}, X_{1:t-1}^{(i)}\right\}$ ($i = 1, \ldots N$, $W_{t-1}^{(i)} > 0$, $\sum_{i=1}^N W_{t-1}^{(i)} = 1$) approximating $\widetilde{\pi}_{t-1}$ is available, i.e. the empirical measure

$$\widehat{\widetilde{\pi}}_{t-1}\left(dx_{1:t-1}\right) = \sum_{i=1}^N W_{t-1}^{(i)} \delta_{X_{1:t-1}^{(i)}}\left(dx_{1:t-1}\right),$$

is an approximation of $\widetilde{\pi}_{t-1}$. At time t, one extends the path of each particle by sampling from an importance distribution, $q\left(x_t|x_{0:t-1}, y_{1:t}\right)$, which could for example be a Markov kernel, $K_t\left(x, x'\right)$, giving the probability or probability density of moving to x' when the current state is x. Importance sampling can then be used to correct for the discrepancy between the sampling distribution and the target, $\widetilde{\pi}_t\left(x_{1:t}\right)$. In this situation one has;

- If $X_{1:t-1}^{(i)} \sim \mu_{t-1}$ and target is $\widetilde{\pi}_{t-1}$ then

$$W_{t-1}^{(i)} \propto \frac{\widetilde{\pi}_{t-1}\left(X_{1:t-1}^{(i)}\right)}{\mu_{t-1}\left(X_{1:t-1}^{(i)}\right)}, \quad \sum_{i=1}^{N} W_{t-1}^{(i)} = 1.$$

- If $X_t^{(i)} \Big| X_{t-1}^{(i)} \sim K_t\left(X_{t-1}^{(i)}, \cdot\right)$ and target is $\widetilde{\pi}_t$ then

$$\begin{aligned} W_t^{(i)} &\propto \frac{\widetilde{\pi}_t\left(X_{1:t}^{(i)}\right)}{\mu_{t-1}\left(X_{1:t-1}^{(i)}\right) K_t\left(X_{t-1}^{(i)}, X_t^{(i)}\right)} \\ &= W_{t-1}^{(i)} \frac{\widetilde{\pi}_t\left(X_{1:t}^{(i)}\right)}{\widetilde{\pi}_{t-1}\left(X_{1:t-1}^{(i)}\right) K_t\left(X_{t-1}^{(i)}, X_t^{(i)}\right)} \end{aligned}$$

the normalized weights are given by

$$W_t^{(i)} \propto W_{t-1}^{(i)} w_t\left(X_{1:t}^{(i)}\right), \quad \sum_{i=1}^{N} W_t^{(i)} = 1, \tag{2.4}$$

where the incremental weight is equal to

$$w_t\left(x_{1:t}\right) = \frac{\widetilde{\pi}_t\left(x_{1:t}\right)}{\widetilde{\pi}_{t-1}\left(x_{1:t-1}\right) K_t\left(x_{t-1}, x_t\right)}. \tag{2.5}$$

Hence, it has been shown how one may approximate the target distribution using a weighted delta mass or particle representation. It is also worth noting that this representation involves approximating a continuous random variable by a discrete random variable, with random support from the continuous target density. Additionally, the

weighted particle representation will form an increasingly more accurate representation of the target distribution as the number of particles is increased. As was shown in the papers mentioned earlier, convergence results have been established in which the asymptotic limit of the particle representation of the target distribution has been shown to converge to the true target distribution for a range of different classes of convergence.

Of key importance is the fact that in order to obtain a well represented point mass approximation of the target distribution, in which the particles are located in regions of the support where the target distribution has most mass, one must endeavour to select an importance density which resembles the target distribution as closely as possible. Hence, when it is possible, in order to obtain a set of weighted particles which accurately represents the true target distribution, one should strive to select the importance density so as to minimise the variance of the importance weights. It is intuitive therefore that the efficiency of the importance sampling methodology is directly related to the choice of the importance sampling density.

In [41], it is demonstrated that the optimal Importance Sampling density for a general SIS framework is given by $\widetilde{\pi}_t\left(x_t|x_{1:t-1}\right)$. This importance density is optimal in the sense that it minimises the conditional variance of the particle weights. It is important to understand that much work has been spent developing importance densities which may be used as an approximation to this optimal importance density, in applications in which it is not possible or not easy to sample the optimal importance density. The "efficiency", η, of an importance density has been studied and a "rule of thumb" criterion was established by which one could use the estimated efficiency to quantify the effective sample size, as first shown in [14] and [61].

In order to obtain this expression for the effective sample size, one must first consider the efficiency. The efficiency is determined by the following set up :

- Suppose the mean $\mathbb{E}\left[\varphi\left(x\right)\right]$ of some function $\varphi\left(x\right)$ is of interest
- $\widehat{\mathbb{E}}_q^N\left(\varphi\right) = \frac{1}{N}\sum_{i=1}^{N} W^*\left(X^{(i)}\right)\varphi\left(X^{(i)}\right)$ an estimate of the mean using samples drawn from importance density $q(.)$
- $\mathbb{E}_\pi^N\left(\varphi\right) = \frac{1}{N}\sum_{i=1}^{N}\varphi\left(X^{(i)}\right)$ an estimate of the mean using samples drawn from the true target distribution $\pi(.)$

$$\eta_q = \frac{var_q\left(\widehat{\mathbb{E}}_q^N\left(\varphi\right)\right)}{var_\pi\left(\mathbb{E}_\pi^N\left(\varphi\right)\right)} \approx 1 + var_q\left(W^*\left(X\right)\right) = \mathbb{E}_q\left[W^*\left(X\right)^2\right].$$

This approximation has been shown to hold in [61], when test function $\varphi\left(x\right)$ varies sufficiently slowly. This approximation is obtained by utilising the Delta method, which is explained in [87] Chapter 3, and disregarding all but the first two terms of the Taylor expansion. This approximation has had such a wide use due to the fact that in quantifying the efficiency it only depends on the weights obtained from the IS step. This makes it applicable for any scenario. The related yet more common form of quantifying the performance is to consider the effective sample size. The effective sample size is a well established measure used throughout the literature to quantify the performance of a particular importance density. The effective sample size E_{ff} provides a measure of how much the importance distribution differs from the target distribution. If N independent samples are drawn from the importance distribution $q(.)$, then the effective sample size is given by,

$$E_{ff} \triangleq \frac{N}{\eta_q} \approx \frac{N}{\mathbb{E}_q\left(W^*\left(X\right)^2\right)} \approx \left(\frac{1}{N^2}\sum_{i=1}^{N} W^*\left(X\right)^2\right)^{-1}$$

Further consideration is that SIS in its present form suffers from a serious problem which has become known as degeneracy. The degeneracy problem of the SIS algorithm is that in nearly all situations, after only a few iterations, all but a few particles will

have negligible weight. This is a serious problem as it means that the particle estimate of the target distribution is not very good, since all but a few of the particles are located in regions of the target distribution with significant mass. It should be noted that the degeneracy problem will always be present since the variance of the importance weights can only increase over time. However, one should take steps to minimise the degeneracy. One of the most important applications of the effective sample size is in quantifying the degeneracy. A small effective sample size indicates severe degeneracy of the algorithm.

One can take several steps to minimise the degeneracy of an algorithm, the most obvious being to increase the number of particles used until an acceptable effective sample size is obtained. This is not necessarily practical as it presents an excessive computational burden in many situations. The next option is to ensure that one uses an importance density which is as close to the optimal as possible. This will ensure that the variance of the IS weights is minimised and hence the effective sample size will be maximised. This can be explained as a direct result of sampling an importance density which places most of its mass in regions of support to which the target distribution also places most of its mass. The third means of minimising the degeneracy is known as resampling which was first introduced in this context by [50] and then shortly after by [59].

The resampling criterion commonly used, is to resample only when the effective sample size drops below some threshold, which as a rough guide is typically in the range of 30 to 60% of the total number of particles used. The purpose of resampling is to reduce the degeneracy present in a particle filter by eliminating samples which have low importance weights and multiplying samples with large importance weights [11]. There are many methods that one may use to perform resampling, such as multinomial resampling [81], residual resampling [64] and "stratified /systematic/minimum variance" resampling [58]. The multinomial approach is the simplest, involving sampling from a multinomial distribution in which the normalised weights of the particles are used as the probabilities in the multinomial distribution.

All the methods mentioned ensure that the number of times a particle is multiplied as a result of resampling is unbiased, that is $\mathbb{E}\left[N_t^{(i)} | \left\{W_{1:t}^{(i)}\right\}\right] = NW_{1:t}^{(i)}$, however they differ in $var\left[N_t^{(i)}\right]$. The method that the author recommends is that of systematic stratified resampling, which is the minimum variance unbiased resampling technique.

The final point to make is that, although resampling reduces the effects of degeneracy on the particle approximation, it does introduce other problems. Resampling increases Monte Carlo variance, is time consuming and limits the ability of an algorithm to be run on parallel computers, since all particles must be combined for resampling. Secondly, although resampling may aid in the problem of degeneracy, when particles which have high weights are statistically resampled many times, this will lead to a loss in diversity of the particles since the resultant set of resampled particles will contain many repeated samples. This problem is known as sample impoverishment and can be severe when the process noise is too small. In the situation in which one experiences sample impoverishment, since the diversity of the particles paths is reduced, then any smoothed estimates which are based on these particle paths will degenerate, making smoothing inaccurate. In order to counteract this problem of sample impoverishment, by introducing diversity to the resampled batch of particles, it was first suggested in [45] that an MCMC step, may be used in order to add diversity to the repeated particles. This technique, when it is possible to apply an MCMC or "particle diversification" step, can be very effective in reducing the sample impoverishment.

To summarise, the generic SMC algorithm proceeds as presented next. The initial importance distribution is denoted as μ_1.

Initialization; $t = 1$.

Sampling step

- For $i = 1, ..., N$, sample $X_1^{(i)} \sim \mu_1(\cdot)$ and evaluate the normalized weights $W_1^{(i)}$.

$$W_1^{(i)} \propto \frac{\widetilde{\pi}_1\left(X_1^{(i)}\right)}{\mu_1\left(X_1^{(i)}\right)}, \quad \sum_{i=1}^{N} W_1^{(i)} = 1. \tag{2.6}$$

At time n; $n \in \mathcal{N} \setminus \{1\}$.

Sampling step

- For $i = 1, ..., N$, sample $X_t^{(i)} \sim K_t\left(X_{t-1}^{(i)}, \cdot\right)$.
- For $i = 1, ..., N$, evaluate the normalized weights $W_t^{(i)}$ using (2.4) and (2.5).

Resampling step

- If E_{ff} <Threshold then resample particles $\left\{W_t^{(i)}, X_t^{(i)}\right\}$ to obtain N new particles $\left\{N^{-1}, X_t^{(i)}\right\}$.
- Diversification step : MCMC step

To finish of this section on Sequential Monte Carlo methods it will be instructive to provide an example of how the general framework just presented is used in many applications, in practice. The framework which is adopted by many practitioners who use SMC methods involves state space modelling and presenting SMC in the case of filtering for a state conditioned on some noise observation sequence. In these situations the target distribution of interest is typically the posterior distribution of the state conditioned on a

realisation of some noisy observation sequence. Due to the prolific representation of SMC methods in this light, the author feels it instructive to briefly present the basic ideas of casting SMC in this framework. This has relevance in many fields including tracking, control, computer vision and finance and hence the author feels it is important to include as an example in any discussion on SMC methods.

State space modelling is a widely used method in science and engineering, for formulating models of dynamical systems [26], [1], [13]. State space modelling assumes that one has an observed time series (Y_t) which is derived from an unobserved state process (X_t). The state process forms a Markov chain $\{X_0, X_1, ...\}$ and conditionally on this state process the Y_t's are independent, and Y_t depends on X_t. The general model that will be of interest involves two measurable spaces (E, ε) and $(F, \mathcal{F})$, where X_t and Y_t respectively take their values.

It is useful to note that the joint process (X_t, Y_t) is a Markov process on the product space $E \times F$, however the observation process (Y_t) is typically not a Markov process. The transition equation and initial distribution for the state Markov process $\{x_t; t \in N\}$, $x_t \in \mathbb{R}^{n_x}$, will be denoted by $p(x_t|x_{t-1})$ and $p(x_0)$, respectively. The observation process $\{y_t; t \in N\}$, $y_t \in \mathbb{R}^{n_y}$, is assumed to be conditionally independent given the hidden state process, with likelihood $p(y_t|x_t)$. Hence, the general aim of the analysis will be to estimate the posterior distribution $p(x_{0:t}|y_{1:t})$ and its attributes. It is often useful to formulate this problem as shown below:

$$\begin{aligned} x_t &= f_t(x_{t-1}, v_{t-1}) && \text{State equation} \\ y_t &= h_t(x_t, n_t) && \text{Observation equation} \end{aligned}$$

where $f_t(.)$ represents the state equation, v_{t-1} the state noise process and $h_t(.)$ is the observation equation with observation noise n_t.

Given this first order Markov dependence in the state equation, one may write the combined distribution of the state process at time t as shown in equation (2.7). Ad-

ditionally, the joint distribution over the observation process conditioned on the state sequence, at time t, may be written as shown in equation (2.8).

$$p\left(x_{0:t}\right) = p\left(x_0\right) \prod_{n=1}^{t} p\left(x_n | x_{n-1}\right) \tag{2.7}$$

$$p\left(y_{1:t} | x_{1:t}\right) = \prod_{n=1}^{t} p\left(y_n | x_n\right) \tag{2.8}$$

Using the Bayesian methodology explained previously, one may obtain the posterior distribution of the state conditional on the observations, as shown in equation (2.9).

$$\widetilde{\pi}_t\left(x_{0:t}\right) = p\left(x_{1:t} | y_{1:t}\right) = \frac{p\left(y_{1:t} | x_{1:t}\right) p\left(x_{0:t}\right)}{p\left(y_{1:t}\right)} \tag{2.9}$$

In many real world applications one is interested in making a sequential 'on-line' inference on the "state" of the system as new observations are considered, hence a recursive update to the posterior is required. Using the model assumptions stated previously this recursive update can be performed, as shown in equation(2.10).

$$\begin{aligned} \widetilde{\pi}_{t+1}\left(x_{0:t+1}\right) &= \frac{p\left(x_{t+1} | x_t\right) p\left(x_{0:t}\right) p\left(y_{t+1} | x_{t+1}\right) p\left(y_{1:t} | x_{1:t}\right)}{p\left(y_{t+1} | y_{1:t}\right) p\left(y_{1:t}\right)} \\ &= \widetilde{\pi}_t\left(x_{0:t}\right) \frac{p\left(y_{t+1} | x_{t+1}\right) p\left(x_{t+1} | x_t\right)}{p\left(y_{t+1} | y_{1:t}\right)} \end{aligned} \tag{2.10}$$

One may also obtain the filtering distribution by marginalising out the previous states

$$\pi_t\left(x_t\right) = \int p\left(x_{1:t} | y_{1:t}\right) dx_{1:t-1}$$

or, in a recursive setting, by following prediction and update steps, shown below.

$$\begin{aligned} \pi_{t-1}\left(x_t\right) &= \int p\left(x_t | x_{t-1}\right) p\left(x_{t-1} | y_{1:t-1}\right) dx_{t-1} && \text{Prediction} \\ \pi_t\left(x_t\right) &= \frac{p\left(y_t | x_t\right) p\left(x_t | y_{1:t-1}\right)}{p\left(y_t | y_{1:t-1}\right)} && \text{Update} \end{aligned}$$

Now that the general SMC framework has been recast in the form of filtering recur-

sions it is also instructive to present the form of the optimal importance distribution, which is now given by,

$$\begin{aligned} q\left(x_t|x_{t-1}, y_t\right)_{opt.} &= p\left(x_t|x_{t-1}, y_t\right) \\ &= \frac{p\left(y_t|x_t, x_{t-1}\right) p\left(x_t|x_{t-1}\right)}{p\left(y_t|x_{t-1}\right)} \\ &= \frac{p\left(y_t|x_t, x_{t-1}\right) p\left(x_t|x_{t-1}\right)}{\int p\left(y_t|x_t\right) p\left(x_t|x_{t-1}\right) dx_t}. \end{aligned}$$

2.6 Summary

To summarise it is noted that the importance of Monte Carlo methods in simulations of stochastic systems and in estimation of integrals has been presented. It was explained why the key to Monte Carlo methods revolves around the ability of one to efficiently simulate samples from an appropriate probability distribution. It was explained what alternatives are possible when generating samples directly from the desired distribution is not possible. These included importance sampling in which one generates samples from a importance distribution close to the desired target distribution and also Markov Chain Monte Carlo methods in which one produces statistically dependent samples. These samples may then be used in the Monte Carlo analysis. The next chapter deals with the new contributions developed by Pierre Del Moral, Arnaud Doucet and the author of this thesis.

Chapter 3

Sequential Monte Carlo Samplers

3.1 Introduction

This chapter introduces work which is a collaboration between the author of the thesis, Dr. Arnaud Doucet of Cambridge University and Professor Pierre Del Moral, working at Université Paul Sabatier (Toulouse III) at the time of collaboration. This work is found in [31]. The work focused on developing a general methodology to enable one to sample sequentially from a sequence of probability distributions which are known up to a normalizing constant and defined on a common space. In the same manner as standard Sequential Monte Carlo, it will be the aim to approximate these probability distributions by a cloud of weighted random samples which are propagated over time using Sequential Monte Carlo methods. The generality of this methodology, termed SMC Samplers, allows one not only to derive simple algorithms to make parallel Markov Chain Monte Carlo runs interact in a principled manner, but also to obtain new methods for global optimisation and sequential Bayesian estimation. This methodology also paves the way for the discussion and development of Trans-Dimensional Sequential Monte Carlo, TDSMC, which forms the second body of work in this thesis. The algorithms developed using SMC Samplers will then be demonstrated through simulation for various integration and global optimisation tasks arising in the context of Bayesian inference.

3.2 Motivation for SMC Samplers

The idea that has driven the development of SMC Samplers was the need to be able to obtain a particle estimate from a sequence of probability distributions, $(\pi_t)_{t\in\mathcal{N}}$, which are defined on a common measurable space, E, where $\mathcal{N} = \{1, \ldots, p\}$ or $\mathcal{N} = \mathbb{N}^+$. Throughout this thesis t will be referred to as the time index, however this variable is just a counter and need not have any relation with "real time". The idea behind SMC Samplers will be to sample the sequence of distributions π_1, π_2... sequentially. This has many important applications and it should be mentioned that the generality of the method to be presented, comes from the freedom in the choice of the sequence of distributions $(\pi_t)_{t\in\mathcal{N}}$.

Some of the interesting applications involve sequential methods to move from a tractable and easy to sample distribution, π_1, to a distribution of interest, π_p, through a sequence of artificial intermediate distributions as discussed in [69]. In the setting of Bayesian inference one could consider π_t to be the posterior distribution of a parameter given the data collected until time t, where $\pi_t(x) = p(x|\, y_1, \ldots, y_t)$. In a batch setting, in which a fixed set of observations $y_1, \ldots, y_T$ is available, then the sequence of distributions one is interested in could be $p(x|\, y_1, \ldots, y_t)$ for $t \leq T$. There are two reasons for approaching a batch problem in this manner. First, treating batch data sequentially has been shown to provide a beneficial tempering effect [28]. This is especially important for very large data sets, for which the chosen models typically exhibit complex probability surfaces. In these situations treating the data sequentially causes the probability surface to exhibit a natural tempering effect, which results in the ability to move from a simple to an increasingly more complex surface as more data points are included. Thus, a sequential strategy allows an efficient exploration of the probability surface without the need to construct complex annealing schedules. Second, for huge data sets, standard simulation methods such as Markov Chain Monte Carlo (MCMC) methods require a complete "browsing" of the observations, in contrast a sequential strategy may have reduced computational complexity, as discussed in [73]. Another interesting application

can be found in the context of optimisation, and similar to simulated annealing, one could consider the sequence of distributions $\pi_t(x) \propto [\pi(x)]^{\gamma_t}$ for an increasing schedule $\{\gamma_t\}_{t\in\mathcal{N}}$. Finally, one could simply consider the sequence of distributions where $\pi_t = \pi$ for all $t \in \mathcal{N}$. Hence, one can see that the motivation behind developing SMC Samplers methodology is that it would find applications in several areas of interest.

3.3 SMC Samplers Methodology

The framework of Sequential Monte Carlo (SMC) has been discussed in Chapter 2. Standard SMC techniques have been developed to deal with "on-line" applications which involve sampling from a sequence of distributions sequentially in time. Until the development of SMC Samplers, SMC techniques have been solely confined to situations which involve a sequence of probability distributions $(\widetilde{\pi}_t)$ where a distribution, at time t in the sequence, is defined on a measurable product space of the form $E_t = E \times E \times E .... = E^t$, which means that $\dim(E_{t-1}) < \dim(E_t)$. SMC Samplers generalises the methodology of SMC in order to sample sequentially from a sequence of probability distributions (π_t) where now each distribution in the sequence is defined on a common measurable space, E. Typically the methods favoured by statisticians to sample from complex distributions, on a fixed space E, are MCMC methods. The fundamental ideas that underpin MCMC techniques were presented in Chapter 2. Two problems with MCMC are that it is difficult to assess when the Markov chain has reached its stationary regime and it can easily get stuck in local modes. Moreover, it is not ideal to use MCMC in a sequential Bayesian estimation context.

It is important to note that SMC Samplers should be viewed as a complementary approach to MCMC, and that MCMC kernels will in most cases be ingredients of the methods proposed here. Additionally, it is worth noting that an SMC based approach in which particles are carried forward over time using a combination of Sequential Importance Sampling (SIS) and resampling ideas is completely different from parallel MCMC/tempering

mechanisms, where one runs an MCMC chain on an extended space E^N. When carrying out parallel MCMC/tempering, one specifies a joint invariant distribution on E^N for the particles [49], whereas the use of SMC samplers requires only the specification of a distribution on E.

In the development of SMC Samplers one would like to be able to maintain the benefits of standard SMC methodology. This was achieved by effectively transforming the problem posed above into the framework familiar to standard SMC techniques. The important concept developed is that in order to use the methodology of SMC, which involves Sequential Importance Sampling (SIS), resampling or resample and move techniques, one would need to come up with a means of transforming the idea of sampling from a sequence of distributions, which are each defined on E, to that of sampling from a sequence in which each distribution, $\widetilde{\pi}_t$, is defined on the product space E^t, $t \in \mathbb{N}$. Hence the idea was to construct a sequence of distributions, $(\widetilde{\pi}_t)$, which are defined on the product space, E^t, required by the standard SMC methodology. The important consideration is however that in order for this construction to be used as a method to sample sequentially from the sequence (π_t) where each distribution, π_t, is defined on E, one needs to construct $\widetilde{\pi}_t$ in such a way that it admits as a marginal distribution the required target distribution π_t. This approach has connections with Annealed Importance Sampling (AIS) [69] and the algorithms recently proposed in [28] and [23]. However, it will be demonstrated that the generic framework presented by SMC Samplers is more general and allows one to develop new algorithms to make parallel MCMC runs interact in a simple and principled way, to perform global optimization, solve sequential Bayesian estimation problems or compute the probabilities of rare events. As with MCMC, the performance of these algorithms is highly dependent on the target distributions $(\widetilde{\pi}_t)_{t\in\mathcal{N}}$ and proposal distributions used to explore the space. Throughout this thesis effective guidelines will be presented for the design of efficient algorithms which utilise SMC Sampler methodology.

Consider now the construction of the target distributions, $(\widetilde{\pi}_t)_{t\in\mathcal{N}}$. The construction of the sequence of distributions, as proposed in [31], is carried out as shown in equation

(3.1). For $t = 1$, consider $\widetilde{\pi}_1(x_1) = \pi_1(x_1)$ and for $t > 1$, one has

$$\widetilde{\pi}_t(x_{1:t}) = \pi_t(x_t)\,\widetilde{\pi}_t(x_{1:t-1}|x_t) \tag{3.1}$$

The distribution $\widetilde{\pi}_t(x_{1:t-1}|x_t)$ is designed in such a way that for any $x_t \in E$ the distribution $\widetilde{\pi}_t(x_{1:t-1}|x_t)$ is a probability distribution on the product space E^{t-1}. In order to allow for a recursive evaluation of the importance weights it is wise to use $\widetilde{\pi}_t(x_{1:t-1}|x_t)$ as shown in equation (3.2).

$$\widetilde{\pi}_t(x_{1:t-1}|x_t) = \prod_{s=1}^{t-1} L_s(x_{s+1}, x_s) \tag{3.2}$$

The kernels L_s form a sequence $\{L_t\}_{t\in\mathbb{N}}$ which would ideally be a sequence of auxiliary Markov transition kernels, in which the kernel $L_s(x, x')$ represents the probability or probability density depending on the context, of making a move from state x to state x'. It is now obvious how this construction allows one to transform the problem of sampling from the sequence of distributions (π_t), defined on the space E, to that of sampling from a sequence of distributions $(\widetilde{\pi}_t)$ defined on E^t and then obtaining samples from the required distribution by just taking the marginal distribution as shown in equation (3.3). Note that for ease of exposition, x_t shall be used to represent the state of the system at time t. It is also the case that x_t may be an element of a state space E which in generality can be multi-dimensional, in which case x_t would be a vector of state variates at any given time t.

$$\int \widetilde{\pi}_t(x_{1:t})\,dx_{1:t-1} = \pi_t(x_t) \tag{3.3}$$

This formulation is that of the familiar family of algorithms developed in the SMC literature, as illustrated in Chapter 2, section 4. Thus one may consider the application of this idea to carrying out SIS on a sequence of distributions, coupled with resampling or resample-move steps. After constructing $(\widetilde{\pi}_t)_{t\in\mathcal{N}}$ in order to transform SMC Samplers framework into the form of standard SMC algorithms, it is important to present what

form the incremental importance weight will take for this new framework. The standard incremental importance weight for SMC techniques is presented in (2.5) and this can now be used to develop the new importance weight for this SMC Samplers algorithm to obtain equation (3.4). Hence, the Particle Filtering framework is recovered using these new SMC Sampler ideas as follows. The sequential weighting steps are presented below, with the importance sampling density given by the transition kernel $K_t(x', x)$ and the incremental weight,

$$\begin{aligned} w_t(x_{1:t}) &= \frac{\widetilde{\pi}_t(x_{1:t})}{\widetilde{\pi}_{t-1}(x_{1:t-1}) K_t(x_{t-1}, x_t)} \\ &= \frac{\pi_t(x_t) \widetilde{\pi}_t(x_{1:t-1}|x_t)}{\pi_{t-1}(x_{t-1}) \widetilde{\pi}_{t-1}(x_{1:t-2}|x_{t-1}) K_t(x_{t-1}, x_t)} \\ &= \frac{\pi_t(x_t) \prod_{s=1}^{t-1} L_s(x_{s+1}, x_s)}{\pi_{t-1}(x_{t-1}) \prod_{s=1}^{t-2} L_s(x_{s+1}, x_s) K_t(x_{t-1}, x_t)} \\ &= \frac{\pi_t(x_t) L_{t-1}(x_t, x_{t-1})}{\pi_{t-1}(x_{t-1}) K_t(x_{t-1}, x_t)} = w_t(x_{t-1}, x_t). \end{aligned} \tag{3.4}$$

Where the IS weight is now given by $W_t^{(i)} \propto w_t^{(i)} W_{t-1}^{(i)}$. The particles may then be resampled from these weights in the desired method, several examples of which are mentioned in the chapter on SMC methods. The author always advocates the use of stratified/systematic resampling technique, as it has been shown that this resampling method minimises the variance of the importance weights. It is important to point out that the introduction of the sequence of auxiliary kernels, $\{L_t\}_{t\in\mathcal{N}}$, allows for the use of importance sampling without having to compute the marginal distribution of the particles $\left\{X_t^{(i)}\right\}$ explicitly, which is typically difficult to calculate. This is discussed further in the next section along with discussion of how to choose the parameters of the algorithm which include $\{K_t\}_{t\in\mathcal{N}}$ and $\{L_t\}_{t\in\mathcal{N}}$ in order to minimise the variance of the importance weights obtained.

3.4 SMC Samplers Specifics: Theoretical and Algorithmic Considerations

This section explains how one may deal with the fact that the SMC Samplers methodology introduces additional degrees of freedom to standard SMC. In standard SMC algorithms, one has the sequence of distributions, $\{\widetilde{\pi}_t\}$ which have typically been selected according to the problem being solved and the user must choose a suitable sequence of importance distributions / transition kernels $\{K_t\}_{t\in\mathcal{N}}$. Typically one chooses the transition kernels to have two properties; easy to sample and also as close to the transition kernel that minimizes the variance of the importance weights as possible. In SMC Samplers algorithms there is also the additional freedom in choosing the auxiliary transition kernels $\{L_t\}_{t\in\mathcal{N}}$. The following section provides a theoretical analysis of how to select these kernels in order to minimise the variance of the importance weights, then suggested algorithmic settings are presented.

3.4.1 Asymptotic Analysis of Variance

This section provides an expression for the asymptotic variance of the estimate shown in equation (3.5) , which was obtained using the SMC Samplers algorithm. The expression presented in Proposition 1 was derived in [31] and builds on the work of [62], [32] using similar ideas to [27]. This expression is included as it will be very useful when it comes to understanding how the selection of the auxiliary kernels $\{L_t\}_{t\in\mathcal{N}}$ will affect the variance of estimates obtained using SMC Samplers.

$$\widehat{\mathbb{E}}_{\pi_t}(\varphi) = \sum_{i=1}^{N} W_t^{(i)} \varphi\left(X_t^{(i)}\right) \tag{3.5}$$

Proposition 1 *Under the weak integrability conditions given in (Chopin, 2004; theorem 1) or (Del Moral, 2004, section 9.4, pp. 300-306), one obtains the following results. When no resampling is performed, one has*

$$\sqrt{N}\left(\widehat{\mathbb{E}}_{\pi_t}(\varphi)-\mathbb{E}_{\pi_t}(\varphi)\right)\Rightarrow\mathcal{N}\left(0,\sigma^2_{IS,t}(\varphi)\right)$$

with

$$\sigma^2_{IS,t}(\varphi)=\int\frac{\widetilde{\pi}^2_t(x_{1:t})}{\mu_t(x_{1:t})}\left(\varphi(x_t)-\mathbb{E}_{\pi_t}(\varphi)\right)^2dx_{1:t} \tag{3.6}$$

where the importance distribution μ_t is given by

$$\mu_t(x_{1:t})=\mu_1(x_1)\prod_{s=2}^{t}K_s(x_{s-1},x_s).$$

When multinomial resampling is used at each iteration, one has

$$\sqrt{N}\left(\widehat{\mathbb{E}}_{\pi_t}(\varphi)-\mathbb{E}_{\pi_t}(\varphi)\right)\Rightarrow\mathcal{N}\left(0,\sigma^2_{SMC,t}(\varphi)\right)$$

where, for $n\geqslant 2$,

$$\begin{aligned}&\sigma^2_{SMC,t}(\varphi) \qquad (3.7)\\ =&\int\frac{\widetilde{\pi}^2_t(x_1)}{\mu_1(x_1)}\left(\int\varphi(x_t)\widetilde{\pi}_t(x_t|x_1)dx_t-\mathbb{E}_{\pi_t}(\varphi)\right)^2dx_1\\ &+\sum_{s=2}^{t-1}\int\frac{\left(\widetilde{\pi}_t(x_s)L_{s-1}(x_s,x_{s-1})\right)^2}{\pi_{s-1}(x_{s-1})K_s(x_{s-1},x_s)}\left(\int\varphi(x_t)\widetilde{\pi}_t(x_t|x_s)dx_t-\mathbb{E}_{\pi_t}(\varphi)\right)^2dx_{s-1:s}\\ &+\int\frac{\left(\pi_t(x_t)L_{t-1}(x_t,x_{t-1})\right)^2}{\pi_{t-1}(x_{t-1})K_t(x_{t-1},x_t)}\left(\varphi(x_t)-\mathbb{E}_{\pi_t}(\varphi)\right)^2dx_{t-1:t}.\end{aligned}$$

In these expressions $\int\widetilde{\pi}_t(x_{1:t})dx_{1:s-1}dx_{s+1:t}$ is denoted by $\widetilde{\pi}_t(x_s)$ and $\int\widetilde{\pi}_t(x_{1:t})dx_{1:s-1}dx_{s+1:t-1}/\widetilde{\pi}_t(x_s)$ by $\widetilde{\pi}_t(x_t|\,x_s)$. The proof of Proposition 1 is found in Appendix 1.

What is evident from this theoretical analysis is that careful selection of the auxiliary kernels $\{L_t\}$ is imperative in order to obtain an algorithm that provides sensible answers.

This is made explicit by the fact that, in the expression for the asymptotic variance, it can be seen that the faster the mixing of the $\{L_t\}$ kernels then the faster $\widetilde{\pi}_t(x_t|x_s)$ converges to $\pi_t(x_t)$, as $t-s$ increases. Therefore in the situation that the $\{L_t\}$ kernels are mixing well, variance terms in the summation, with $s \ll t$, will become insignificant since the square difference given by

$$\left(\int \varphi(x_t)\,\widetilde{\pi}_t(x_t|x_s)\,dx_t - \mathbb{E}_{\pi_t}(\varphi)\right)^2$$

will be negligible. This means that in situations in which the $\{L_t\}$ kernels are mixing rapidly, one may obtain a good approximation of the variance by just taking the last few terms in the expression for the variance shown in equation (3.7), since the remaining terms in the sum will be negligible. However, it is worth considering the situation in which the $\{L_t\}$ kernels have their fastest mixing. This will be the situation in which $L_{t-1}(x_t, x_{t-1}) = L_{t-1}(x_{t-1})$. Now, in this situation if one assumes all terms in the variance expression obtained in 3.7 are negligible except for the last term, then one obtains a variance expression given by,

$$\sigma^2_{SMC,t}(\varphi) = \int \frac{(\pi_t(x_t)\,L_{t-1}(x_t, x_{t-1}))^2}{\pi_{t-1}(x_{t-1})\,K_t(x_{t-1}, x_t)}\left(\varphi(x_t) - \mathbb{E}_{\pi_t}(\varphi)\right)^2 dx_{t-1:t}.$$

Given the form of this variance expression it would be tempting to select $L_{t-1}(x_t, x_{t-1}) = \pi_{t-1}(x_{t-1})$ as then cancellation is possible and one obtains a variance expression given by,

$$\sigma^2_{SMC,t}(\varphi) = \int \frac{(\pi_t(x_t))^2\,\pi_{t-1}(x_{t-1})}{K_t(x_{t-1}, x_t)}\left(\varphi(x_t) - \mathbb{E}_{\pi_t}(\varphi)\right)^2 dx_{t-1:t}.$$

As pointed out in [31], the reason why this is not a good idea is a result of the fact that, in order to ensure that the variance expression $\sigma^2_{SMC,t}(\varphi)$ is small for any function φ, this would require the ability to control the importance weight

$$\frac{\pi_t(x_t)}{K_t(x_{t-1}, x_t)} \tag{3.8}$$

over $E \times E$. As mentioned in the paper, this expression would typically prohibit the use of MCMC moves, as the ratio in 3.8 is typically not defined. The example provided to demonstrate this point involves taking $\pi_t(x_t)$ as a probability density on $\mathbb{R}$ and K_t a Metropolis-Hastings kernel. Hence for these reasons it is wise not to use $L_{t-1}(x_t, x_{t-1}) = L_{t-1}(x_{t-1})$ and the trade-off is that a sum or terms will appear in the expression of the variance. However most of these terms in the sum will contain a part which takes the form

$$\frac{\widetilde{\pi}_t(x_s) L_{s-1}(x_s, x_{s-1})}{\pi_{s-1}(x_{s-1}) K_s(x_{s-1}, x_s)}$$

which can be controlled more easily via selection of $L_{s-1}(x_s, x_{s-1})$ as a function of $K_s(x_{s-1}, x_s)$ and $\pi_{s-1}(x_{s-1})$.

With these results in mind the following section will discuss and motivate different choices for the sequence of auxiliary kernels, $\{L_t\}$. In addition to this, a comparison and links will be drawn with algorithms in the literature which can be viewed as special cases of this general SMC Sampler framework.

3.4.2 Auxiliary Kernels $\{L_t\}$

As discussed above, careful selection of the sequence of auxiliary kernels $\{L_t\}$ is important if one is to obtain sensible estimates using the SMC Samplers framework. The expression obtained in (3.7) suggests that one should try to optimise the selection of $\{L_t\}$ with respect to $\{K_t\}$ kernels. There seems to be two approaches to consider when selecting the sequence of kernels $\{L_t\}$. The first involves minimising the asymptotic variance of the estimate (3.5) with respect to the kernels $\{L_t\}$. The second approach would be to look directly at the variance of the importance weights, as found in (3.4), and to try to minimise the variance of the importance weights with respect to the $\{L_t\}$ kernels. As was pointed out in [31] it was found to be a good idea to consider the second option and select $\{L_t\}$ kernels by considering directly the variance of the importance weights since this would make the choice of the $\{L_t\}$ kernels independent of the function φ.

In order to explain how to obtain the solution to minimising the variance of the importance weights with respect to kernels $\{L_t\}$ one must first define the following items. The marginal distribution of particles, $\left\{X_t^{(i)}\right\}$, at time t shall be denoted $\mu_t(x_t)$ and will have one of the following forms depending on whether resampling has taken place. When no resampling has occurred up to time t, one has

$$\mu_t(x_t) = \mu_1 K_{2:t}(x_t) \tag{3.9}$$

and if the last time the particles were resampled was at time l then one has

$$\mu_t(x_t) = \pi_l K_{l+1:t}(x_t). \tag{3.10}$$

Now, assume one has particles distributed according to μ_t, at time t, and it is desired to have them distributed according to the target distribution π_t. One way of obtaining particles distributed according to π_t would be to correct for the discrepancy between the distributions via simple Importance Sampling weights. Hence, one would obtain the following expression for the un-normalised correction weights for the particles

$$\frac{\pi_t(x_t)}{\mu_t(x_t)}. \tag{3.11}$$

The problem with using this simplistic approach is that it could be either too computationally intensive or extremely difficult, if not impossible, to obtain an analytical expression for $\mu_t(x_t)$, at each time t. Therefore a first naive approach would be to look for a solution to this dilemma such as using an approximation of the form

$$\widehat{\mu}_t(x_t) = \frac{1}{N}\sum_{i=1}^{N} K_t\left(X_{t-1}^{(i)}, x_t\right).$$

This is also not ideal since clearly it would produce an algorithm which is $O(N^2)$, and this should be avoided when possible. If one took the approach presented previously and

performed the IS correction, using the weights obtained in (3.4), then clearly one no longer has to compute $\mu_t(x_t)$. However, this comes at a price since now the importance weights are defined on domain E^t, as opposed to E, and so will ultimately produce larger variance in the importance weights. This problem is rectified by the fact that one may choose the sequence of $\{L_t\}$ kernels, and intuitively the optimal choice minimising the variance of the importance weights will be the one that takes us from evaluating the weights on E^t back to evaluating the weights simply on E. The following proposition from [31] provides a solution to this problem. The version of the solution presented will be for the situation shown in (3.9) where no resampling has occurred.

Proposition 2 *The conditional distribution $\widetilde{\pi}_t^{opt}$ on E_{t-1} which minimises the variance of the importance weights, $w_t(x_{1:t})$, is given by*

$$\widetilde{\pi}_t^{opt}(x_{1:t-1}|x_t) = \mu_t(x_{1:t-1}|x_t) \tag{3.12}$$

and this conditional distribution takes the form provided in equation 3.2, with for any s,

$$L_{s-1}^{opt}(x_s, x_{s-1}) = \frac{\mu_{s-1}(x_{s-1}) K_s(x_{s-1}, x_s)}{\mu_s(x_s)}. \tag{3.13}$$

The proof of Proposition 2 may be found in Appendix 2. Now, it is obvious that although this is the optimal solution in terms of minimising the variance of the importance weights, with respect to the $\{L_t\}$ kernels, this will not be of use in practice since one still can not easily calculate $\mu_t(x_t)$, as explained previously. One can either choose to approximate L_{t-1}^{opt} or choose kernels $\{L_t\}$ so that the importance weights are easily calculated or have a familiar form. It is in this second approach that parallels are found with existing literature on the subject. The connections to other works are explained in [31]. It is however useful for completeness to outline the connections found between the SMC Samplers methodology and existing work, in more detail.

The first connection that will be mentioned is of importance to the simulation section to be presented next. This connection relates to the work of Jarzynski, [56] and also

to the Annealed Importance Sampling (AIS) algorithm of Neal, [69]. The problem Neal discusses involves moving from a tractable distribution to a distribution of interest via a sequence of intermediate annealed distributions. Neal discusses the fact that the annealed sequence of distributions is typically used as an inexact means of handling isolated modes in Markov chain samplers. He then demonstrates how the Markov chain transitions used for the annealing sequence can be developed to define an importance sampler. He argues that the combination of importance sampling and Markov chain samplers has two advantages. The first advantage comes from the fact that the Markov chain aspects allow for acceptable performance in high dimensional spaces where it may be difficult to design effective importance sampling proposal distributions. The second advantage is that using importance sampling allows for a correction to be made to the Markov chain samples, to make these techniques asymptotically exact, that is as the number of runs is increased the estimates will converge to their correct values. Hence, Neal combined positive aspects of both methods to provide a means of assigning weights to the states which are obtained by making multiple Simulated Annealing (SA) [57] runs. When one is looking at the problem of moving from a distribution which is easy to sample $\mu_1(x)$, to a distribution of interest $\pi(x)$, through a sequence of intermediate distributions. One way of doing this could be to consider the sequence of distributions suggested in [47], where they consider a geometric path such as the one given by equation (3.14) below.

$$\pi_t(x) \propto [\pi(x)]^{\gamma_t} [\mu_1(x)]^{1-\gamma_t} \tag{3.14}$$

If one considers such a geometric sequence and uses a transition kernel, K_t, which is an MCMC transition kernel of invariant distribution π_t, then the form of the L_{t-1} kernel which recovers the AIS algorithm as presented in [69] (with the time index reversed) is given by the following auxiliary kernel.

$$L_{t-1}(x_t, x_{t-1}) = \frac{\pi_t(x_{t-1}) K_t(x_{t-1}, x_t)}{\pi_t(x_t)} \tag{3.15}$$

This selection of kernels for $\{K_t\}$ and $\{L_t\}$ will produce an incremental importance weight given by equation (3.16) below

$$\frac{\pi_t(x_{t-1})}{\pi_{t-1}(x_{t-1})}. \tag{3.16}$$

Clearly this will produce importance weights which are fairly uniform in situations for which $\pi_{t-1} \approx \pi_t$. However, when this is not the case one can expect poor performance from such an algorithm. The weight update presented in 3.16 also allows one to perform auxiliary particle filter concepts to help boost the particles in regions of the state space which will be of importance in the next iteration, prior to mutation by the transition kernel. This can be seen to be the case since the particle weights only depend on the position at the previous iteration and not the new time t. It is also important to mention that the AIS algorithm does not use resampling, the effect of this will be demonstrated in the examples at the end of this chapter.

The second example that demonstrates how SMC Samplers relates to existing work is given by analysing [28]. This paper presents the Importance Sub-sampling Iterative Scheme, (ISIS). This algorithm allows one to obtain samples from a static posterior, $\pi(\theta|y_{1:N})$, by carrying out initial exploration of partial distributions, $\pi(\theta|y_{1:n})$ $(n < N)$. The ISIS algorithm considers the sequence of partial posterior distributions $(\pi_t(\theta|y_{1:n_t}))$, with $n_1 < ... < n_t < ... < n_T = N$. It operates by obtaining a set of particles distributed as $\pi(\theta|y_{1:n_1})$, then this inference is "updated" in a consistent manner, recursively taking the next p observations into account according to the weight given in equation (3.17).

$$w_{n,p}(\theta) = \frac{\pi(\theta|y_{1:n+p})}{\pi(\theta|y_{1:n})} \propto \frac{p(y_{1:n+p}|\theta)}{p(y_{1:n}|\theta)} = p(y_{n+1:n+p}|\theta, y_{1:n}) \tag{3.17}$$

The number of observations p used in this update step is adaptively determined according to a criterion presented in the paper and shown in equation (3.18), where one resamples the particles when $D_{n,p} > d$, where for example $d = 10^{-2}$. Basically the need for this criterion is due to the fact that each update step adds more variability to the initial estimates, which leads to a progressive degeneracy of the particle weights. This can

be stated another way; as p increases the support of π_{n+p} will continue to shrink, relative to the support of π_n and hence the particles will become increasingly degenerate. Hence in the same vein as the standard effective sample size criterion presented in Chapter 2, one uses this criterion to decide when the particles need to be resampled.

$$D_{n,p} = \frac{1}{2}\frac{V\left(\widehat{\mathbb{E}}_{n+p}\right)}{V_{\pi_{n+p}}(\theta)} + \frac{1}{4}\frac{V\left(\widehat{V}_{n+p}\right)}{V_{\pi_{n+p}}(\theta)^2} \tag{3.18}$$

$$\begin{aligned}
\widehat{\mathbb{E}}_{n+p} &= \frac{\sum_{i=1}^{N} w_i\theta_i}{\sum_{i=1}^{N} w_i} \\
\widehat{V}_{n+p} &= \frac{\sum_{i=1}^{N} w_i \left\{\theta_i - \widehat{\mathbb{E}}_{n+p}\right\}\left\{\theta_i - \widehat{\mathbb{E}}_{n+p}\right\}^T}{\sum_{i=1}^{N} w_i}
\end{aligned}$$

Once resampling is carried out there is likely to be sample impoverishment and this must be combated if one wants a sample which is a good representation of the posterior $\pi_{n+p}(\theta)$. To combat the problem of sample impoverishment and the fact that the posterior π_{n+p} is likely to place most of its mass either in different regions or in a reduced region of the state space when compared to π_n, one can use the idea of Gilks *et al* [45] where the particles are "moved" according to a Metropolis-Hastings transition kernel with invariant distribution $\pi_{n+p}(\theta)$.

It is the view of the author that developing effective moves which will place the particles in high mass regions of the support of $\pi_{n+p}(\theta)$, is in general a non-trivial task. This is especially true when the mass of the posterior is moving rapidly around the state space as each observation is added, which is the case when one has informative observations and also in the initial stages when one only has a small number of observations, n small, and the mass of the posterior is still concentrating itself.

The suggested method of [28] is to use an Independent Metropolis-Hastings transition kernel which depends weakly on the previous value of the moved particle. The suggested transition kernel is given by a Gaussian with mean $\widehat{E}_{n+p}$ and covariance $\widehat{V}_{n+p}$ shown in

(3.18). Thinking in the framework of SMC Samplers, one can see that the algorithm presented in [28] is obtained as a special case within the SMC Samplers framework when one has K_t as an MCMC transition kernel of invariant distribution π_t and L_{t-1} is given by (3.15). Hence, ISIS can be seen to be very similar to AIS, except that the ISIS algorithm allows for resampling and the sequence of distributions one wishes to sample is different.

The third example to be presented relates the work of [23] to the SMC Samplers framework. The Population Monte Carlo algorithm presented in [23] can be viewed as a special case of SMC Samplers framework in which one considers the homogeneous situation in which $\pi_t = \pi, K_t = K$ and $L_t = L$. As mentioned in [31] the Population Monte Carlo algorithm considers the case in which the transition kernel $K_t(x, x') = K(x')$ is an MCMC kernel of invariant distribution π such as a Gibbs sampler or the situation in which K depends on the statistics of the entire population of particles at the previous iteration. The auxiliary kernel they consider corresponds to $L_t(x, x') = L(x') = \pi(x')$.

It was explained in [31] that if one was considering the special situation of the homogeneouse set up presented above then in the case that one used an auxiliary kernel given by the same form as [69] and [28] then after resampling once, the particles would be distributed approximately according to π. Then the L kernel becomes the optimal L kernel and the importance weights become unity so that each particle evolves independently according to K and it is not necessary to make them interact anymore. It is then argued that if any other choice is used for the L kernel in this homogeneous situation then resampling would need to be performed periodically and it is pointed out that this approach is not really justified since resampling is not carried out to modify the marginal distribution of the particles but only to modify the correlation between surviving particles at two successive time instants. This would limit the diversity in the set of particles and in general one would not expect the variance of estimates formed using the particles from such a scheme to be any better than that obtained by using non-interacting MCMC chains.

There are many other choices one may consider and some of these are highlighted in detail in [31]. This thesis explores the choice presented in (3.15) and also investigates another approach which was to approximate the expression for the optimal kernel L_{t-1}^{opt}. The approximation used is presented in equation (3.19). This approximation involves substituting the distribution π_{t-1}, which is defined by the problem being solved and assumed known up to a normalising constant, for the distribution μ_{t-1}, which is difficult if not impossible to evaluate analytically.

$$L_{t-1}(x_t, x_{t-1}) = \frac{\pi_{t-1}(x_{t-1}) K_t(x_{t-1}, x_t)}{\pi_{t-1} K_t(x_t)} \tag{3.19}$$

Now following the definition in equation (3.10), it is clear that if the particles are resampled at time $t-1$, then in this case μ_{t-1} will (approximately) equal π_{t-1} and the expression for L_{t-1}, given in (3.19), will equal (3.13). This approximation can be solved in several interesting situations, some of which will be presented next. It should however be mentioned that it may in general be difficult to solve the integral given by $\pi_{t-1} K_t(x_t)$ and hence other approaches should be considered. There will be more discussion on this in the chapter on Trans-Dimensional Sequential Monte Carlo. Using the approximation found in (3.19), the following section presents two detailed examples which demonstrate the performance of the SMC Samplers methodology

3.5 Applications of SMC Samplers

This section provides two detailed examples complete with comparisons between existing algorithms and SMC Samplers methodology. These examples are the joint work of the author and Doucet and can be found in [31]. The problem to be studied is that of variable selection in a Bayesian context. The first example presented will use a sequence of intermediate distributions to move from an initial distribution, μ_1, to a target distribution, π_t, using the SMC Samplers methodology. Comparison with Annealed Importance Sampling [69] and MCMC algorithms will be provided to demonstrate the performance

of SMC Samplers relative to other algorithms in the literature.

The second example is an optimization problem in which one would like to find the mode of the distribution π_t. This allows SMC Sampler methodology to be compared to a long chain annealed MCMC algorithm and also a parallel annealing MCMC algorithm. Before going into the details of these two examples, the Bayesian variable selection problem shall be presented so that one can understand what sequence of distributions will be used in each example.

3.5.1 Bayesian Variable Selection

For any $(X, Y) \in \mathcal{X} \times \mathbb{R}$, we consider the following regression model [60]

$$Y = \sum_{k=1}^{M} I_k \beta_k \Psi_k (X) + V;\ V \sim \mathcal{N}\left(0, \sigma^2\right) \tag{3.20}$$

where the indicator variable, $I_k \in \{0, 1\}$, is such that $\beta_k = 0$ if $I_k = 0$ and $\beta_k \neq 0$ if $I_k = 1$. In this situation there will be 2^M possible models for the regression function. Now, if one assumes there are T independent identically distributed data points, denoted $(X_{1:T}, Y_{1:T})$, then the following vector-matrix notation can be used

$$Y_{1:T} = D\left(I_{1:M}\right) \beta\left(I_{1:M}\right) + V_{1:T},$$

where $D\left(I_{1:M}\right)$ is a $T \times l\left(I_{1:M}\right)$ matrix and $l\left(I_{1:M}\right) = \sum_{k=1}^{M} I_k$ is the number of basis terms included in the model. The jth column of $D\left(I_{1:M}\right)$ corresponds to $\left(\Psi_{\alpha(I_{1:M}, j)}\left(X_1\right), \ldots, \Psi_{\alpha(I_{1:M}, j)}\left(X_T\right)\right)^{\mathrm{T}}$ where $\alpha\left(I_{1:M}, j\right)$ is the index of the jth non-null coefficient of the sequence $I_{1:M}$ and $\beta\left(I_{1:M}\right)$ is the associated $l\left(I_{1:M}\right)$-dimensional vector of non-null regression coefficients. To complete the model in a Bayesian framework the following priors shall be used for the amplitudes of the basis functions and the variance of the observation

noise.

$$\begin{aligned}\beta\left(I_{1:M}\right)|\left(\sigma^2, I_{1:M}\right) &\sim \mathcal{N}\left(0, \delta^2\sigma^2\left(D^{\mathrm{T}}\left(I_{1:M}\right)D\left(I_{1:M}\right)\right)^{-1}\right), \\ \sigma^2 &\sim \mathcal{IG}\left(\frac{\gamma_0}{2}, \frac{\nu_0}{2}\right),\end{aligned}$$

This choice of prior was made as it allows one to perform Rao-Blackwellisation on the parameters for the amplitudes of the basis functions and the observation noise variance. That is, they can be integrated out of the posterior since they have the form of a g-prior [34], which coupled with the linear form (with non-linear basis functions) of expression (3.20) and the Gaussian observation noise, allows for this integration to be made. The details of such an integration are omitted, but a reference in which they are carried out in detail may be found in [79], appendix A, page 202. Secondly, the choice of a g-prior as opposed to, for example, a ridge prior allows one to remove the assumption of prior independence between the amplitude coefficients, hence one does not need to imply that one is using orthogonal basis functions. The properties which make the g-prior favourable are that when one has basis functions along similar projections, then the coefficients of these basis functions will be highly correlated, *a priori.* A lucid discussion of the attributes of the g-prior and the ridge prior may be found in [34], page 80.

Finally, the following specifications were made, $\Pr\left(I_k = 1|\,\lambda\right) = \lambda$ where λ is uniformly distributed on $[0,1]$ and γ_0, ν_0 and δ are fixed hyperparameters. After integrating out those parameters discussed previously and given a realization $(x_{1:T}, y_{1:T})$, one obtains the following marginal posterior distribution for the indicator variables

$$p\left(i_{1:M}|\,x_{1:T}, y_{1:T}\right) \propto \left(\nu_0 + y_{1:T}^{\mathrm{T}} P\left(i_{1:M}\right) y_{1:T}\right)^{T/2+\frac{\gamma_0}{2}} \left(1+\delta^2\right)^{-l(i_{1:M})/2} l\left(i_{1:M}\right)!\left(T - l\left(i_{1:M}\right)\right)!$$

where

$$P\left(i_{1:M}\right) = I_{l(i_{1:M})} - \left(1+\delta^{-2}\right)^{-1} D\left(i_{1:M}\right)\left(D^{\mathrm{T}}\left(i_{1:M}\right)D\left(i_{1:M}\right)\right)^{-1} D^{\mathrm{T}}\left(i_{1:M}\right)$$

with $I_{l(i_{1:M})}$ the identity matrix of dimension $l(i_{1:M})$.

The first example of interest will be to sample this marginal posterior distribution and in the second example it will be the aim to carry out optimisation in order to determine the maximal mode of this distribution. The data for both examples is taken from the sinc function, i.e. $\text{sinc}(x) = \sin(x)/x$, corrupted by additive Gaussian noise, with $\sigma = 0.1$ for $T = 50$ evenly spaced points in the interval $[-10, 10]$. It is assumed there are $M = T$ basis functions of the form

$$\Psi_k(x) = \frac{1}{\sqrt{2\pi}\phi} \exp\left(-\frac{(x - x_k)^2}{2\phi^2}\right)$$

where $\phi = 1.6$.

3.5.2 Application 1: Sampling from $p(i_{1:M} | y_{1:T}, x_{1:T})$

The aim of this first example will be to consider a sequence of distributions given by the expression

$$\pi_t(i_{1:M}) \propto [p(i_{1:M} | x_{1:T}, y_{1:T})]^{\gamma_t}$$

where the "annealing" schedule $\gamma_t \in [0, 1]$ and $t \in \{1, \ldots, p\}$ is monotonically increasing. This produces the example where one would like to move from a tractable distribution, for example where $\pi_1 = \mu_1$ is the uniform distribution which corresponds to $\gamma_1 = 0$, to the distribution of interest $\pi_p(i_{1:M}) = p(i_{1:M} | x_{1:T}, y_{1:T})$ which corresponds to $\gamma_p = 1$. In this example the kernels $\{K_t\}$ were selected as deterministic scan Gibbs samplers of invariant distributions (π_t), where one variable was updated per iteration

$$K_t\left(i_{1:M,(t-1)}, i_{1:M,t}\right) = \frac{\pi_t(i_{1:M,t})}{\pi_t(i_{1:M,t}) + \pi_t\left(i_{1:M,(t-1)}\right)}$$

where $i_{1:M,(t-1)} = \left(i_{j,(t-1)}, i_{1:M\setminus j,(t-1)}\right)$ and $i_{1:M,t} = \left(i^*_{j,(t-1)}, i_{1:M\setminus j,(t-1)}\right)$.

Hence the total number of steps, p, required in the "annealing" schedule, γ_t, was set such that $p \gg 1$. For the selection of the L_t kernels both equation (3.19) and the AIS

choice (3.15) were considered. Now, for the jth particle $I^{(j)}_{1:M,(t-1)}$ (resp. $I^{(j)}_{1:M,t}$) at time $t-1$ (resp. t) these algorithmic choices produce the following incremental importance weights.

For the AIS choice of the L_t kernel one obtains the incremental importance weight as follows

$$\begin{aligned} w_t\left(I^{(j)}_{1:M,(t-1)}, I^{(j)}_{1:M,t}\right) &= \frac{\pi_t\left(I^{(j)}_{1:M,t}\right) L_{t-1}\left(I^{(j)}_{1:M,t}, I^{(j)}_{1:M,(t-1)}\right)}{\pi_{t-1}\left(I^{(j)}_{1:M,(t-1)}\right) K_t\left(I^{(j)}_{1:M,(t-1)}, I^{(j)}_{1:M,t}\right)} \\ &= \frac{\pi_t\left(I^{(j)}_{1:M,t}\right) \frac{\pi_t\left(I^{(j)}_{1:M,(t-1)}\right) K_t\left(I^{(j)}_{1:M,(t-1)}, I^{(j)}_{1:M,t}\right)}{\pi_t\left(I^{(j)}_{1:M,t}\right)}}{\pi_{t-1}\left(I^{(j)}_{1:M,(t-1)}\right) K_t\left(I^{(j)}_{1:M,(t-1)}, I^{(j)}_{1:M,t}\right)} \\ &= \frac{\pi_t\left(I^{(j)}_{1:M,(t-1)}\right)}{\pi_{t-1}\left(I^{(j)}_{1:M,(t-1)}\right)}. \end{aligned}$$

For the second choice of the L_t kernel given by (3.19) one obtains the incremental importance weight

$$w_t\left(I^{(j)}_{1:M,(t-1)}, I^{(j)}_{1:M,t}\right) = \frac{\pi_t\left(I^{(j)}_{1:M,t}\right) L_{t-1}\left(I^{(j)}_{1:M,t}, I^{(j)}_{1:M,(t-1)}\right)}{\pi_{t-1}\left(I^{(j)}_{1:M,(t-1)}\right) K_t\left(I^{(j)}_{1:M,(t-1)}, I^{(j)}_{1:M,t}\right)}$$

$$\begin{aligned} &= \frac{\pi_t\left(I^{(j)}_{1:M,t}\right) \frac{\pi_{t-1}\left(I^{(j)}_{1:M,(t-1)}\right) K_t\left(I^{(j)}_{1:M,(t-1)}, I^{(j)}_{1:M,t}\right)}{\pi_{t-1} K_t\left(I^{(j)}_{1:M,(t-1)}, I^{(j)}_{1:M,t}\right)}}{\pi_{t-1}\left(I^{(j)}_{1:M,(t-1)}\right) K_t\left(I^{(j)}_{1:M,(t-1)}, I^{(j)}_{1:M,t}\right)} \\ &= \frac{\pi_t\left(I^{(j)}_{1:M,t}\right)}{\pi_{t-1} K_t\left(I^{(j)}_{1:M,t}\right)} \end{aligned}$$

now the integral $\pi_{t-1}K_t\left(i_{1:M,t}\right)$ has the following form

$$\begin{aligned}\pi_{t-1}K_t\left(i_{1:M,t}\right) &= \pi_{t-1}\left(i_{1:M,(t-1)}\right)K_t\left(i_{1:M,(t-1)},i_{1:M,t}\right)+\pi_{t-1}\left(i_{1:M,t}\right)K_t\left(i_{1:M,(t-1)},i_{1:M,t}\right)\\ &= \left[\pi_{t-1}\left(i_{1:M,(t-1)}\right)+\pi_{t-1}\left(i_{1:M,t}\right)\right]\frac{\pi_t\left(i_{1:M,t}\right)}{\pi_t\left(i_{1:M,t}\right)+\pi_t\left(i_{1:M,(t-1)}\right)}\end{aligned}$$

hence this produces an incremental importance weight with the following form

$$\begin{aligned}w_t\left(I^{(j)}_{1:M,(t-1)},I^{(j)}_{1:M,t}\right) &= \frac{\pi_t\left(I^{(j)}_{1:M,t}\right)}{\pi_{t-1}K_t\left(I^{(j)}_{1:M,t}\right)}\\ &= \frac{\pi_t\left(I^{(j)}_{1:M,(t-1)}\right)+\pi_t\left(I^{(j)}_{1:M,t}\right)}{\pi_{t-1}\left(I^{(j)}_{1:M,(t-1)}\right)+\pi_{t-1}\left(I^{(j)}_{1:M,t}\right)}\end{aligned}$$

In this case, and more generally in any discrete state-space problems with local exploration, it is usually possible to compute (3.19) exactly. It is worth mentioning that when one considers the importance weights for each of the choices of L_t kernel, clearly one can not expect much difference between using (3.19) or the AIS choice, when one is in the situation that $\pi_t \approx \pi_{t-1}$. It should also be pointed out that the computational complexity of AIS and of the alternative method proposed are similar.

The following set of experiments were carried out for different values of $p \in \{250, 500, 1250, 2500, 5000\}$. The schedule used for this example had $\gamma_1 = 0$ with the sequence $\{\gamma_t\}$ initially increasing linearly for $\left\lfloor \frac{p}{5} \right\rfloor$ steps and then according to $a\log(t)+b$ with $\gamma_p = 1$.

The number of particles, used for all simulations in this example, was $N = 1000$. Additionally, an adaptive resampling scheme was used for the SMC Samplers algorithm where resampling was performed when the E_{ff} was below $N/2$. The SMC Samplers algorithm was also compared to sampling from π with a Gibbs sampler, using pN iterations for the computational complexity of both methods to be approximately similar.

The results of the simulations are presented in table 1, which displays the average Mean Square Error (MSE) and the standard deviation of the MSE estimate of the re-

gression function over 50 simulations using the same data set. The average and standard deviation of the mean of the log-posterior of the last population of particles is also presented. For the MCMC results, the average and standard deviation of the mean of the log-posterior of the samples obtained after burn-in are presented. Two sets of results are presented for AIS with no resampling, one using (3.19) and the other using (3.15). The results for the new SMC Samplers algorithm were produced using (3.19) for the choice of the L_{t-1} kernel and the Gibbs sampler results were produced by discarding the first 40% of samples as burn-in period. For each simulation, the same N random initial starting points are used for AIS and SMC and one of those N points was used to initialize the Gibbs sampler. The results demonstrate that there is almost no difference between AIS using (3.19) or (3.15).

The results demonstrate that in all simulations, the resampling step used in the SMC algorithm produces a reduction in the variance cheaply. The reduction of the variance is most prominent when the number of updates per site is small, hence p is small. Intuitively this makes sense, since in these situations the difference between π_{t-1} and π_t can be significant when compared to the situations in which p is very large and $\pi_{t-1} \approx \pi_t$. An additional point is that, as would be expected, the number of times resampling is carried out increases as p decreases. The results also demonstrate that for large p where $\pi_{t-1} \approx \pi_t$, the SMC algorithm and AIS give almost similar results with regard to the average MSE and its standard deviation. However, the average log-posterior for the final population of samples is clearly higher for SMC compared to AIS, this trend is demonstrated graphically in figure 1 below. The plot in figure 1 demonstrates the average and the standard deviation of the mean log posterior for all of the simulations versus the number of updates per site for AIS and SMC Samplers. In each of the 50 simulations this average was computed using the last set of particles at time p. The average and standard deviation results for MCMC simulations are also presented, except the mean log posterior for each simulation was calculated for the samples remaining after discarding the burn in period. It can be seen, from the plot in figure 1, that for small values of p SMC Samplers

yields samples with higher log-posterior values than the AIS algorithm and the MCMC Gibbs sampler. As p increases, the MCMC algorithm outperformed SMC Samplers and AIS in terms of the average mean log posterior, however the standard deviation of the mean log posterior was significantly larger for MCMC compared to both AIS and SMC Samplers. In fact the SMC Samplers algorithm results are contained within $\pm 1\sigma$ of the MCMC average mean log posterior as demonstrated in figure 1.

Table 1 also demonstrates that compared to SMC, MCMC algorithms yield a lower average MSE. Nevertheless, only one realization of observations has been used so this is not significant. This argument is supported by the fact that the average MSE appeared to be unchanged at approximately around 2.2 even for low values of the average mean log posterior. Additionally, it is worth mentioning that the average effective sample size for the last population of the particles for the SMC Samplers algorithm is significantly better than the results obtained for the AIS algorithm, as shown below in Table 1. When this result is coupled with the results of the average mean log posterior for the last population of particles it demonstrates that the particles simulated in the SMC Samplers algorithm are exploring important regions of the state space with respect to where the target posterior places most of its mass. However the AIS algorithm was not exploring regions of the state space which were as significant and very few particles were located in these regions of interest with respect to the target posterior.

	Updates per site for MCMC to be computationally equivalent (pN/50)				
	$5N$	$10N$	$25N$	$50N$	$100N$
MCMC					
avg. MSE	2.29	2.32	2.49	2.48	2.21
std. MSE	0.93	0.83	0.91	0.81	0.71
avg. mean log posterior	-0.33	3.57	5.81	6.31	6.26
std. mean log posterior	3.61	4.54	4.75	2.92	2.22
	Updates per site (p/50)				
	5	10	25	50	100
AIS with (3.15)					
avg. MSE	4.79	3.26	3.50	3.29	3.44
std. MSE	2.83	1.25	1.60	1.08	1.06
avg.mean log posterior last population	-7.17	-3.21	-0.76	0.90	2.12
std. mean log posterior last population	0.24	0.17	0.23	0.20	0.26
avg. E_{ff} last population	3.50	4.50	13.35	79.21	85.21
AIS with (3.19)					
avg. MSE	4.78	3.26	3.50	3.28	3.44
std. MSE	2.83	1.25	1.60	1.08	1.06
avg. E_{ff} last population	3.52	4.51	13.39	81.27	87.89
SMC with (3.19)					
avg. MSE	3.05	3.04	3.34	3.19	3.65
std. MSE	1.45	1.22	1.17	0.93	1.01
avg. mean log posterior last population	2.53	3.84	4.09	4.63	4.51
std. mean log posterior last population	2.21	0.67	1.56	0.42	0.69
avg. number of resampling steps	7.86	7.82	6.46	4.98	3.22
avg. E_{ff} last population	820.97	925.10	756.96	880.23	802.22

Table 1: Performance of MCMC, AIS and SMC over 50 simulations

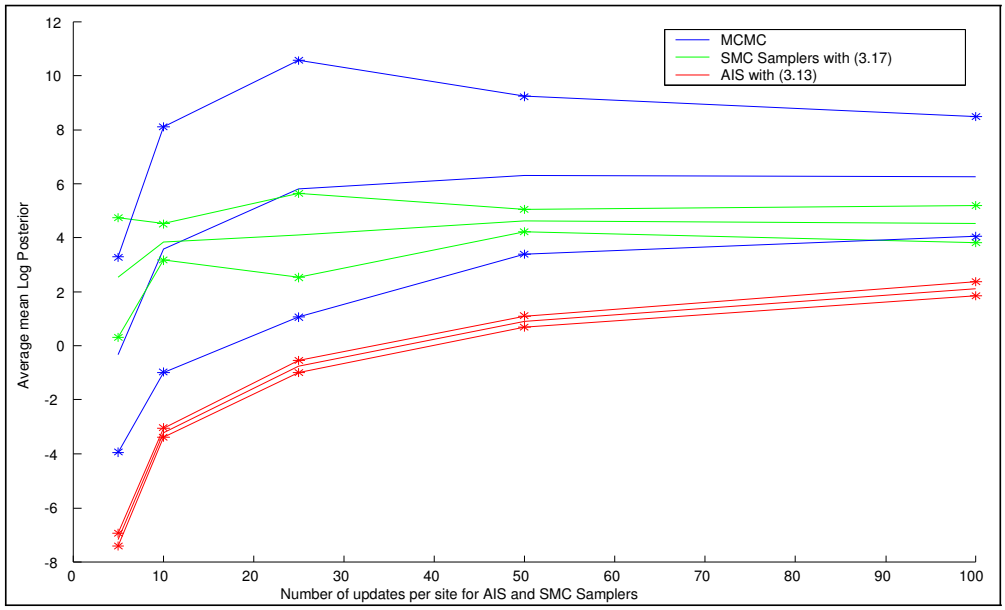

Figure 1: Blue: Average and σ of mean log posterior for MCMC samples after burn in, Green: Average and σ for mean log posterior for SMC Samplers (with 3.17) last set of particles, Red: Average and σ for mean log posterior for AIS (with 3.13) last set of particles.

3.5.3 Application 2 : Optimization of $p\left(i_{1:M}\middle| y_{1:T}, x_{1:T}\right)$ to find the Mode

The aim of this second example will again be to consider a sequence of distributions given by the following expression

$$\pi_t\left(i_{1:M}\right) \propto \left[p\left(i_{1:M}\middle| x_{1:T}, y_{1:T}\right)\right]^{\gamma_t}$$

where now the "annealing" schedule has $\gamma_t \in [0, 10]$ with $t \in \{1, \ldots, p\}$ for the same values of p that were used in Example 1. The annealing schedule in this example was given by $\gamma_1 = 0$ with the sequence $\{\gamma_t\}$ initially increasing linearly for $\left\lfloor \frac{p}{3} \right\rfloor$ steps and then according to $a \log(t) + b$ with $\gamma_p = 10$. This produces an example where one would like

to explore the modes of the distribution $\pi_p(i_{1:M}) = p(i_{1:M} | x_{1:T}, y_{1:T})$ culminating at the end of the annealing schedule, when $\gamma_p = 10$, in an estimate of the most probable mode for this multimodal distribution. Again in this example, as was the case in Example 1, the $\{K_t\}$ were selected as deterministic scan Gibbs samplers of invariant distributions (π_t), where one variable was updated per iteration. The settings of the SMC algorithm will be the same as presented in Example 1.

Two simulated annealing versions of the Gibbs sampler presented in Example 1 were used for comparison with the SMC Samplers algorithm. Long run annealing with $\gamma_1 = 0$ and $\gamma_{pN} = 10$ was used as one of the comparative algorithms and the other was N parallel (non-interacting) annealing runs in which $\gamma_1 = 0$ and $\gamma_p = 10$ were used. As a note for each iteration of the annealing schedule, one of the M variables was updated using the Gibbs deterministic scan sampler. Hence for the long run annealing algorithm it is obvious that each of the M sites were updated pN/M times and for the parallel annealing example in which there were N parallel non-interacting chains, then each site was updated a total of p/M times.

In table 2, we display the average log-posterior values of the estimated mode and its standard deviations over 50 simulations. Again we use the same data set and the same initialization procedure. The posterior mode estimate used to obtain the results for each algorithm was chosen as the sample generated during the simulation which maximized the posterior distribution.

The results of the simulations demonstrated that the SMC algorithm outperforms both these techniques and again this is especially evident when p is small. This is best demonstrated by the simulations where there are 5 updates per site, which corresponds to $p = 250$. In all the simulations, the best estimated mode had a log posterior value of 14.31. In this case the number of times the SMC algorithm reaches a maximum log-posterior value equal to 14.31 is more than twice as often as parallel annealing and the long run annealing never obtains a maximum log-posterior value even close to it, this is also graphically demonstrated in figure 2. The plot in figure 2 clearly demonstrates

that SMC Samplers has the best performance in terms of the number of times a mode greater that 14.3 is obtained, out of the 50 simulations performed for each experiment. Each experiment in the plot corresponds to a different number of updates per site with experiment 1 being the simulations performed for the number of updates per site as 5. The plot demonstrates that most significant difference in performance in which SMC Samplers is clearly out performing both Long run Annealing and Parallel Annealing is when the number of updates per site is small, and hence when the difference between π_{t-1} and π_t is large. This type of performance was also observed in application 1 and further emphasises the gains that can be made through the introduction of resampling which allows the simulated annealing chains to interact in a principled manner.

	Updates per site for Long run				
	Annealing to be equivalent (pN/50)				
	$5N$	$10N$	$25N$	$50N$	$100N$
Long run Annealing					
avg. max log posterior mode	5.07	7.59	8.90	11.12	12.13
std. max log posterior mode	3.24	3.21	3.22	2.74	2.40
number of times reached mode14.31	0	1	8	17	24
	Updates per site				
	5	10	25	50	100
SMC optimization with (3.19)					
avg. max log posterior mode	11.67	13.15	14.26	14.31	14.31
std. max log posterior mode	2.35	1.47	0.34	0.00	0.00
number of times reached mode 14.31	14	30	49	50	50
Parallel Annealing runs					
avg. max log posterior mode	11.51	12.40	14.15	14.26	14.31
std. max log posterior mode	1.41	1.43	0.66	0.34	0.00
number of times reached mode 14.31	6	16	46	49	50

Table 2: Performance of simulated annealing and SMC over 50 simulations

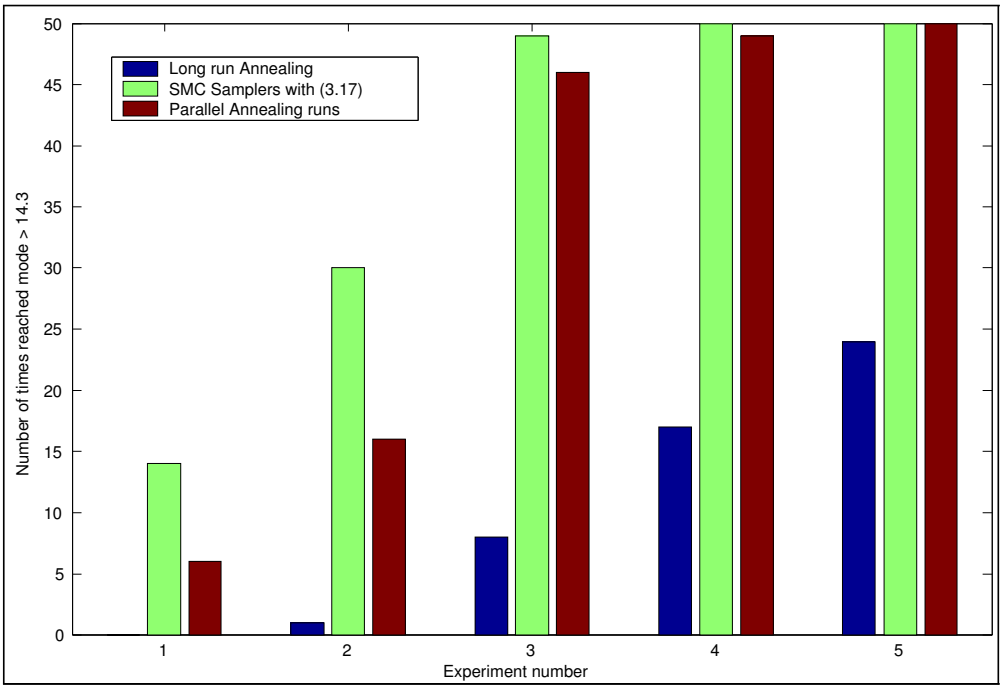

Figure 2: Number of times out of 50 simulations each algorithm reached a mode greater than 14.3. Blue: Long run Annealing, Green: SMC Samplers (with 3.19), Brown: Parallel Annealing runs

It is interesting to mention that Figure 2 demonstrates that for all the experiments the Parallel Annealing simulations out performed the Long run Annealing simulations when it comes to the number of times a mode greater than 14.3 is obtained. This is an interesting result which can be considered for future research since generally one would expect the results of these two types of simulation to be approximately the same and not demonstrate such stark contrasts as was obtained in the simulations carried out.

3.6 Summary

This chapter has provided an introduction to new methodology termed SMC Samplers. SMC Samplers has been examined both from a theoretical perspective and also from an algorithmic perspective, and connections with existing work were detailed. Finally, simulation examples were provided with a detailed comparison between existing algorithms to demonstrate the performance of the new SMC Samplers algorithms developed using the new methodology. The next chapter will present an extension to the SMC Samplers methodology which has been termed Trans-Dimensional Sequential Monte Carlo (TDSMC). Just as SMC Samplers were presented as an SMC analogue of MCMC methods, this methodology has been developed to be an SMC analogue of RJMCMC methods.

Chapter 4

Trans-Dimensional Sequential Monte Carlo (TDSMC)

4.1 Introduction

This chapter introduces work which is a collaboration between the author of the thesis, Dr. Arnaud Doucet and Dr. Jaco Vermaak of Cambridge University. This chapter builds on the previous chapter on SMC Samplers methodology by providing a general framework in which one may carry out joint model order determination and parameter estimation for the analysis of data sequentially. The TDSMC framework to be presented can be used in either the sequential analysis of batch data or in the analysis of sequential data for truly "on-line" situations. This chapter opens with the reasons for developing TDSMC then the link with SMC Samplers methodology will be made explicit. The remainder of the chapter will subsequently consist of constructing a set of principled "moves" in the same vein as RJMCMC methodology. After developing these moves, a generic TDSMC algorithm will be presented and finally applied in two examples. The first example in this chapter will be sequential linear interpolation and the second will involve sequential Kernel Regression.

4.2 Motivation of TDSMC

This section motivates the development of Trans-Dimensional Sequential Monte Carlo. The primary interest in developing this methodology is to construct an algorithm to perform joint estimation of model order and parameters using a framework based on SMC methods. This is an interesting problem to solve as it has wide ranging applicability which stems from the fact that it is a task that is important in a multitude of disciplines including statistics, engineering, finance and bioinformatics. Additionally the development of TDSMC methods is interesting because the approach taken in developing TDSMC can be viewed as the sequential analogue of the Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm [52],[51] which has been well received in the statistics community. However, unlike RJMCMC, the TDSMC algorithm does not require the moves to be reversible, is non-iterative, and requires only a single pass over the data set. As mentioned in the introduction, TDSMC can also be applied to either batch data sets, where the ordering of the data is unimportant, or to the analysis of sequential data.

To further motivate the idea behind TDSMC one may consider what is common to all problems that the RJMCMC algorithm of Green [52] is used to solve. When viewed in a probabilistic framework the common feature of these applications is a posterior distribution defined on a countable union of sub-spaces, one for each of the candidate models. These sub-spaces are generally of different dimension. So basically the RJMCMC algorithm, reviewed in Chapter 2, is essentially an extension of the Metropolis-Hastings (MH) [66] algorithm to accommodate moves between the model sub-spaces as well as those within a single sub-space. Here it will be shown that in the same sense that RJMCMC extends MCMC one can view the TDSMC algorithm as an analogous extension of the SMC Samplers methodology.

Another important property that enables one to draw parallels between RJMCMC and TDSMC is that both methodologies allow many different moves to be combined to facilitate an efficient exploration of the space that the posterior of interest takes support on. These may include birth moves to add new parameters, death moves to remove

redundant or erroneous parameters, update moves to adjust parameter values, split and merge moves to partition and combine parameters, and many more.

A key difference between the RJMCMC algorithm of Green and the TDSMC framework being proposed in this thesis is that TDSMC is a non-iterative algorithm which only requires a single complete pass over the entire data set, whereas due to its batch nature the RJMCMC algorithm makes use of all the available data for each sample generated, this has several consequences. The first consequence of such an approach is that for very large data sets it can become computationally cumbersome, or even infeasible to have to store and access the entire data set for each iteration, this is a significant problem which is discussed in [73]. Another consequence is that a batch, iterative approach also presents the algorithm with the entire probability surface in all its complexity right from the onset which can make efficient localisation and exploration of the modes a potentially difficult problem. There have been approaches suggested which attempt to mitigate this problem. One such approach is to combine RJMCMC with Simulated Annealing (SA) [7] with the aim being to be able to move from a simple probability surface to an increasingly more complex one. However, the construction of an annealing schedule that allows an efficient exploration and guarantees convergence, is a difficult problem.

Hence, these difficulties highlighted lend support to the argument that it is often also beneficial to treat batch data sequentially, as it has been shown to provide a useful tempering effect [28]. This is especially important, as stated before, in the situations where one has very large data sets for which the chosen models exhibit complex probability surfaces. In these situations when the data is treated sequentially the probability surface exhibits a natural tempering effect, which results in the ability to move from a simple to an increasingly more complex surface as more data points are included. Thus, with a sequential strategy, an efficient exploration of the probability surface would be possible without the need to construct complex annealing schedules. The final argument for looking at data sequentially is that either the problem is such that the data arrives sequentially and one would like to perform "on-line" analysis, in which case RJMCMC

is not a viable option or alternatively one has a massive data set and drawing on the arguments made by [73], a sequential analysis of the data may lead to significant computational savings.

4.3 TDSMC Methodology

In this section a general framework for joint model order determination and parameter estimation for sequential analysis of data will be developed. The strategy presented is applicable to truly sequential data, as well as batch data. In the batch case the time index has no relation to physical time, and should be interpreted as a counter ranging over the data. In this TDSMC framework the order in which the batch data points are presented to the algorithm is unimportant, and could be random. The best way to think about TDSMC methodology, presented in this section, is as a generalisation of Importance Sampling (IS) to spaces of variable dimension. Then in the same manner in which SMC Samplers recursively updates a sample, or particle, based approximation of the posterior distribution as more data points become available, so too does the TDSMC algorithm. The TDSMC algorithm provides a means of moving "particles" between different dimensional spaces so that one may obtain an empirical particle estimate of the true posterior defined on the space $\mathbf{\Theta}$. Where the space $\mathbf{\Theta}$ of interest in this chapter is of the form $\mathbf{\Theta} = \bigcup_k \{k\} \times \Theta_k$, where k represents the unknown model order and Θ_k represents the space on which the model parameters exist for the kth model. As was the case in the SMC samplers methodology, the interest will be to sample a sequence of distributions $(\pi_t(k, \theta_{1:k}))$, except now the support of each distribution is on space $\mathbf{\Theta}$.

When one has a space of the form $\mathbf{\Theta}$ it is difficult to design efficient proposal distributions that are capable of generating samples directly in the target parameter space. This is largely a result of the fact that the dimension of the parameter space is generally high and variable. Other factors such as non-linearities in the model and multi-modality of the probability surface also create difficulties. To circumvent these problems the target

parameter space is augmented with an auxiliary parameter space, which will be associated with the parameters at the previous time step before the new data point becomes available. The distribution over this joint space is defined in such a way that it admits the target posterior as one of its marginals. This is where the connection between TDSMC and SMC Samplers will become evident. The advantage of doing this is that Importance Sampling in the joint space can now be formulated in terms of moves from the auxiliary parameter space to the target parameter space. This essentially embodies the sequential nature of the algorithm, since the auxiliary parameter space is associated with the parameters at the previous time step before the new data point at the current time step becomes available. Since the parameter spaces are of variable dimension these moves are frequently trans-dimensional. In this sense the algorithm is the sequential Monte Carlo analogue of RJMCMC. The objective of the TDSMC algorithm will be to recursively estimate the joint posterior distribution of the model order and parameters as more data points become available. This produces the sequence of distributions given by $\pi_t(k, \theta_{1:k}) = p_t(k, \theta_{1:k}|\mathbf{y}_{1:t})$, which can be rewritten using Bayes' rule as

$$p_t(k, \theta_{1:k}|\mathbf{y}_{1:t}) \propto p_t(\mathbf{y}_{1:t}|k, \theta_{1:k})p_t(k, \theta_{1:k}), \tag{4.1}$$

and it will be assumed that the target posterior can be evaluated up to a normalising constant. The prior is assumed to factorise as

$$p_t(k, \theta_{1:k}) = p_t(k)p_t(\theta_1)\prod_{l=2}^{k} p_t(\theta_l|\theta_{1:l-1}). \tag{4.2}$$

The interpretation one gives to such a factorisation is that when new parameters are added to the existing parameters they may depend on previous parameter values. Now the following definition presents the target distribution which is defined on the augmented parameter space which clearly admits as a marginal distribution the distribution

of interest $\pi_t(k, \theta_{1:k})$.

$$\pi_t(k, \theta_{1:k}; k', \theta'_{1:k'}) = p_t(k, \theta_{1:k}|\mathbf{y}_{1:t}) L_t(k', \theta'_{1:k'}|k, \theta_{1:k}). \tag{4.3}$$

From a theoretical perspective the choice of the augmenting distribution L_t is arbitrary, however in practice the choice for L_t does affect the performance of the algorithm, for the same reasons that were presented in the chapter on SMC Samplers. It is now possible to use the same L_t kernels presented in Chapter 4, including the approximation (3.19) to the auxiliary kernel which minimised the variance of the particle weights, given by L_t^{opt}. It should be mentioned that k and k' are not constrained in any sense, so that the target and auxiliary parameter spaces may be of different dimension. Now one must define the distribution on the augmented space from which the particles will be generated, this proposal is given by equation (4.4).

$$Q_t(k, \theta_{1:k}; k', \theta'_{1:k'}) = p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1}) K_t(k, \theta_{1:k}|k', \theta'_{1:k'}) \tag{4.4}$$

Hence samples for the parameters at time t are generated by the proposal as a result of incrementally refining the posterior at time $t-1$ via the kernel K_t. One may now employ an Importance Sampling correction to compensate for the discrepancy between the proposal in (4.4) and the joint posterior in (4.3) which is given by the following incremental weight (4.5).

$$w_t(k, \theta_{1:k}; k', \theta'_{1:k'}) = \frac{p_t(k, \theta_{1:k}|\mathbf{y}_{1:t}) L_t(k', \theta'_{1:k'}|k, \theta_{1:k})}{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1}) K_t(k, \theta_{1:k}|k', \theta'_{1:k'})} \tag{4.5}$$

As a result of the construction of the joint distribution in (4.3), marginal samples in the target parameter space associated with this weighting will be distributed according to the desired target posterior $p_t(k, \theta_{1:k}|\mathbf{y}_{1:t})$. Importantly, this convenient result is achieved without the need to marginalise the proposal in (4.4) over the auxiliary space.

The algorithm presented in this form is very general and allows for many potential choices for the kernel K_t which can include the possibility of dependence on individual particle parameter values at time $t-1$ or for examples such as those presented by [92], where the kernel K_t could depend on sufficient statistics drawn from the entire set of particles at time $t-1$. Several choices will be presented for the design of K_t which each represent a different type of "move" and the corresponding approximation (3.19) for the optimal L_t kernel for each "move" will be presented. An underlying characteristic that must be common to any type of move kernel K_t is that it should be constructed in a way that facilitates an efficient exploration of the model spaces.

For a given K_t the choice of L_t is arbitrary, as long as the importance weight in (4.5) is well-defined over the support of the participating distributions. However, a poor selection of L_t may lead to poor performance in practice, this was explained in depth in Chapter 3. So following the arguments presented there it can be shown that the approximation for the optimal L_t kernel that will minimise the variance of the importance weights for a given K_t, takes the form

$$\begin{aligned} L_t^{\text{opt}}(k', \theta'_{1:k'}|k, \theta_{1:k}) &= \frac{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1})K_t(k, \theta_{1:k}|k', \theta'_{1:k'})}{p_{t-1}K_t(k, \theta_{1:k})} \\ &= \frac{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1})K_t(k, \theta_{1:k}|k', \theta'_{1:k'})}{\sum_{l\in\mathcal{K}}\int_{\Theta_l} p_{t-1}(l, \theta_{1:l}|\mathbf{y}_{1:t-1})K_t(k, \theta_{1:k}|l, \theta_{1:l})d\theta_{1:l}}. \end{aligned} \tag{4.6}$$

In many cases the marginalisation in the denominator of the above expression is analytically intractable, so that L_t^{opt} cannot be computed in closed form. The author has developed generic analytical solutions to this problem which have partially been demonstrated in [31] and shall be detailed in this chapter and the next chapter on applications.

4.4 TDSMC Specifics: Theoretical and Algorithmic Considerations

This section will present the main framework for the algorithmic development of TDSMC, which will include the selection of the K_t kernels to accommodate different moves around and between model sub-spaces, Θ_k, as well as the corresponding approximations of the auxiliary kernel L_{t-1}^{opt} for each choice of K_t. One may also choose to use a mixture transition kernel to select from several possible moves at each time instant. In many situations it will be beneficial to consider a mixture of moves in order to efficiently explore the model space.

4.4.1 Multiple Moves

The types of moves that one may be interested in incorporating, in a mixture transition kernel in order to efficiently explore the posterior of interest include the following. Adjustment moves to adjust existing model parameters to incorporate new data, update moves which adjust the weight of a given set of parameters in light of new observations, birth moves to add new parameters to better explain the data and death moves to remove redundant or erroneous parameters. The form of a mixture transition kernel is presented in equation (4.7), where M candidate moves are used in the proposal kernel K_t.

$$K_t(k, \theta_{1:k}|k', \theta'_{1:k'}) = \sum_{m=1}^{M} \alpha_{m,t}(k', \theta'_{1:k'}) K_{m,t}(k, \theta_{1:k}|k', \theta'_{1:k'}), \quad \sum_{m=1}^{M} \alpha_{m,t}(k', \theta'_{1:k'}) = 1. \tag{4.7}$$

Note that the mixture weights corresponding to the different moves may depend on the previous set of parameters. This gives the algorithm an adaptive flavour, since samples can individually choose the most appropriate moves based on their state values.

One may then define the augmenting kernel L_t as a mixture kernel given by

$$L_t(k', \theta'_{1:k'}|k, \theta_{1:k}) = \sum_{m=1}^{M} \beta_{m,t}(k, \theta_{1:k}) L_{m,t}(k', \theta'_{1:k'}|k, \theta_{1:k}), \quad \sum_{m=1}^{M} \beta_{m,t}(k, \theta_{1:k}) = 1,$$

where the mixture weights do not necessarily correspond to those of the proposal kernel in (4.7).

The incremental importance weights may then be calculated by direct substitution of K_t and L_t into (4.5). Another approach which is computationally more efficient is to involve the additional discrete random variable M_t such that $\Pr(M_t = m) = \alpha_{m,t}(k', \theta'_{1:k'})$. Now for each sample, at each time step, new state values are obtained by first randomly sampling M_t to obtain a choice of proposal kernel according to the proposal kernel weights, and then sampling the new state values from the chosen kernel. The corresponding incremental importance weights can then be calculated as

$$w_{m,t}(k, \theta_{1:k}; k', \theta'_{1:k'}) = \frac{p_t(k, \theta_{1:k}|\mathbf{y}_{1:t}) \beta_{m,t}(k, \theta_{1:k}) L_{m,t}(k', \theta'_{1:k'}|k, \theta_{1:k})}{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1}) \alpha_{m,t}(k', \theta'_{1:k'}) K_{m,t}(k, \theta_{1:k}|k', \theta'_{1:k'})}. \tag{4.8}$$

Using a similar argument to that used to obtain the optimal L_t^{opt} kernel in Chapter 4, one may obtain an expression for the optimal weight $\beta_{m,t}^{\text{opt}}$ and auxiliary kernel $L_{m,t}^{\text{opt}}$ which minimises the variance of the weights.

$$\begin{aligned} & \beta_{m,t}^{\text{opt}}(k, \theta_{1:k}) L_{m,t}^{\text{opt}}(k', \theta'_{1:k'}|k, \theta_{1:k}) \\ = & \frac{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1}) \alpha_{m,t}(k', \theta'_{1:k'}) K_{m,t}(k, \theta_{1:k}|k', \theta'_{1:k'})}{\sum_{n=1}^{M} \sum_{l \in \mathcal{K}} \int_{\Theta_{l,t}} p_{t-1}(l, \theta_{1:l}|\mathbf{y}_{1:t-1}) \alpha_{n,t}(l, \theta_{1:l}) K_{n,t}(k, \theta_{1:k}|l, \theta_{1:l}) d\theta_{1:l}}. \end{aligned} \tag{4.9}$$

The computation of the optimal mixture weights, $\beta_{m,t}^{opt}$, for the augmenting kernel may be computationally cumbersome and very difficult to solve. Hence, in practice one often sets these to be equal to the corresponding weights for the proposal kernel, that is one will use $\beta_{m,t}(k, \theta_{1:k}) = \alpha_{m,t}(k', \theta'_{1:k'})$, $m = 1 \cdots M$. This choice has worked well, which illustrates that it is more important to approximate the optimal augmenting kernel than

it is to approximate the optimal mixture weight, at least for all the examples that have been examined using this methodology to date.

To accommodate multiple moves the generalised importance sampling step is presented as follows.

Generalised Importance Sampling Step

- For $i = 1, ..., N$, sample a move index $M_t^{(i)} \sim \{\alpha_{n,t}(k'^{(i)}, \theta'^{(i)}_{1:k'^{(i)}})\}_{n=1}^{M}$.
- For $i = 1, ..., N$, sample $(k^{(i)}, \theta^{(i)}_{1:k^{(i)}}) \sim K_{M_t^{(i)},t}(\cdot|k'^{(i)}, \theta'^{(i)}_{1:k'^{(i)}})$.
- For $i = 1, ..., N$, set the importance weights to $W_t^{(i)} \propto W_{t-1}^{(i)} w_{M_t^{(i)},t}(k^{(i)}, \theta^{(i)}_{1:k^{(i)}}; k'^{(i)}, \theta'^{(i)}_{1:k'^{(i)}})$, and normalise such that $\sum_{i=1}^{N} W_t^{(i)} = 1$.

The next section will present the details for some generic moves which allow for efficient exploration of the state space, and which will be used in simulations to follow.

4.4.2 Construction with Auxiliary Random Variables

The following move is discussed in [52], and its purpose is to allow one to construct the new state at the current time step as a deterministic function of the old state at the previous time step and some auxiliary random variables as shown below in equation (4.10). This type of move will be useful when one would like to perform tasks such as split and merge moves. An application where one could envision this being useful would be for example a sequential basis function regression, where new information favours one basis function as opposed to two. In this setting it may be beneficial to construct the parameters of the new single basis function as a deterministic function of the parameters of the two basis functions.

$$\theta_{1:k} = \psi(\theta'_{1:k'}, \mathbf{u}), \quad \mathbf{u} \sim K_t(\cdot). \tag{4.10}$$

The incremental importance weight is now given by

$$w_t(k, \theta_{1:k}; k', \theta'_{1:k'}) = \frac{p_t(k, \theta_{1:k}|\mathbf{y}_{1:t})L_t(\mathbf{u}')}{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1})K_t(\mathbf{u})} \left| \frac{\partial(\theta_{1:k}, \mathbf{u}')}{\partial(\theta'_{1:k'}, \mathbf{u})} \right|, \tag{4.11}$$

where the Jacobian appears as a result of the random variable transformation. In this case the augmented parameter space is assumed to be constructed through the mapping

$$\theta'_{1:k'} = \varphi(\theta_{1:k}, \mathbf{u}'), \quad \mathbf{u}' \sim L_t(\cdot).$$

From a theoretical perspective the choice of φ and L_t is, as before, arbitrary, as long as the importance weight in (4.11) is well-defined over the support of the participating distributions. Where possible, however, attempts should be made to construct these so that the variance of the importance weights is minimised.

4.4.3 Update Move

Unless very informative measurements are received, the posterior is not expected to change much between the arrival of consecutive measurements. This is especially true in batch settings when the number of measurements received adequately reflects the meaningful statistical variations in the data. Under these circumstances the existing settings for the model parameters will often fit the new data sufficiently well. It therefore makes sense to include a move that leaves the model parameters unchanged, i.e. $\left\{k'^{(i)}, \theta'^{(i)}_{1:k'^{(i)}}\right\} = \left\{k^{(i)}, \theta^{(i)}_{1:k^{(i)}}\right\}$. For such a move both the proposal and augmenting kernels are delta functions as shown below

$$\begin{aligned} K_t\left(k, \theta_{1:k}|k', \theta'_{1:k'}\right) &= \delta_{\left(k', \theta'_{1:k'}\right)}\left(k, \theta_{1:k}\right) \\ L_t\left(k', \theta'_{1:k'}|k, \theta_{1:k}\right) &= \delta_{(k, \theta_{1:k})}\left(k', \theta'_{1:k'}\right) \end{aligned}$$

and hence the incremental importance weight is given by

$$w_t(k, \theta_{1:k}; k', \theta'_{1:k'}) = \frac{p_t(k', \theta'_{1:k'}|\mathbf{y}_{1:t})}{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1})}. \tag{4.12}$$

4.4.4 Birth Move

For data that arrives sequentially a birth move is of paramount importance. Such a move is required to add new parameters to the model to better explain the increasing data. Here it will be assume that only a single new parameter is added, and that the existing parameters remain unaltered. For such a birth move the proposal and augmenting kernels take the form given by

$$\begin{aligned} K_t\left(k, \theta_{1:k}|k', \theta'_{1:k'}\right) &= \delta_{k'+1}\left(k\right)\delta_{\theta'_{1:k'}}\left(\theta_{1:k-1}\right)K_t\left(\theta_k|k', \theta'_{1:k'}\right) \\ L_t\left(k', \theta'_{1:k'}|k, \theta_{1:k}\right) &= \delta_{k-1}\left(k'\right)\delta_{\theta_{1:k-1}}\left(\theta'_{1:k'}\right) \end{aligned}$$

from which it is clear that $k = k' + 1$. In the above, the new parameter is generated from the kernel $K_t\left(\theta_k|k', \theta'_{1:k'}\right)$, which will be specified shortly. For these kernels the incremental importance weight becomes

$$\begin{aligned} w_t(k, \theta_{1:k}; k', \theta'_{1:k'}) &= \frac{p_t(k'+1, \theta'_{1:k'}, \theta_k|\mathbf{y}_{1:t})}{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1})K_t\left(\theta_k|k', \theta'_{1:k'}\right)} \\ &= \frac{p_t(k', \theta'_{1:k'}|\mathbf{y}_{1:t})p_t\left(\theta_k|\theta'_{1:k'}, k', y_{1:t}\right)}{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1})K_t\left(\theta_k|k', \theta'_{1:k'}\right)} \end{aligned}$$

Hence if one would like to minimise the conditional variance of the importance weights it is easy to see that the kernel from which the new parameter is generated should be set to

$$K_t^{\text{opt}}(\theta_k|k', \theta'_{1:k'}) = p_t(\theta_k|\theta'_{1:k'}, k', \mathbf{y}_{1:t}). \tag{4.13}$$

In this case the incremental importance weight is again given by the expression in (4.12). The advantage of this optimal birth weight expression is that it only depends on the parameters obtained in the previous iteration. Hence one may carry out resampling

prior to mutation and correction. This is similar in idea to auxiliary particle filtering and has the aim of boosting particles in regions of high posterior mass before mutation, as first suggested by [70] and then improved in [4]. Note that it is generally not possible to use the optimal proposal in practice, since its normalising constant can normally not be obtained in closed form. However, it is possible to use this strategy for parameters with a finite discrete support. In other cases suitable approximations can often be found.

4.4.5 Death Move

A death move is required to remove parameters that have become redundant, or that have been erroneously added at an earlier time. Here it will be assumed that only a single parameter is removed, and that the remaining parameters are left unaltered. For such a death move there shall be two possible representations of the proposal kernel presented.

Technique 1 This death kernel has been included as it provides a simple method of applying a death move which is computationally cheaper than the second technique to be presented and in many cases provides an effective kernel, which explores the parameter space sufficiently well. For such a death move the proposal kernel can be written as

$$K_t(k, \theta_{1:k}|k', \theta'_{1:k'}) = \delta_{k'-1}(k)\delta_{\theta'_{1:k'\setminus d}}(\theta_{1:k})K_t(d|k', \theta'_{1:k'}), \tag{4.14}$$

from which it is clear that $k = k' - 1$. In the above '$\setminus$' denotes the set difference operator, d is the index of the parameter to be removed, and $K_t(d|k', \theta'_{1:k'})$ is the probability of picking this parameter to be removed. A common approach is to select the parameter to be removed uniformly randomly from the existing parameters, in which case $K_t(d|k', \theta'_{1:k'}) = \mathcal{U}_{\{1\cdots k'\}}(d)$, where $\mathcal{U}_{\mathcal{A}}(\cdot)$ denotes the uniform distribution for the set $\mathcal{A}$. Using the expression in (4.6), it is straightforward to show that the optimal augmenting

kernel for the proposal kernel in (4.14) is given by

$$L_t^{\text{opt}}(k', \theta'_{1:k'}|k, \theta_{1:k}) = \frac{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1})}{p_{t-1}(k'-1, \theta'_{1:k'\setminus d}|\mathbf{y}_{1:t-1})}.$$

For these kernels the incremental importance weight becomes

$$w_t(k, \theta_{1:k}; k', \theta'_{1:k'}) = \frac{p_t(k'-1, \theta'_{1:k'\setminus d}|\mathbf{y}_{1:t})}{p_{t-1}(k'-1, \theta'_{1:k'\setminus d}|\mathbf{y}_{1:t-1})K_t(d|k', \theta'_{1:k'})}. \tag{4.15}$$

Technique 2 This version of the death transition kernel and its associated weight has been provided as an alternative method of carrying out death moves as it may be more efficient at exploring the state space in question. In this case the death move proposal kernel can be written as

$$K_t(k, \theta_{1:k}|k', \theta'_{1:k'}) = \sum_{j=1}^{M} \delta_{k'-1}(k)\delta_{\theta'_{1:k'\setminus j}}(\theta_{1:k})K_t(\theta_j|k', \theta'_{1:k'}), \tag{4.16}$$

from which it is clear that $k = k' - 1$ and again '\' denotes the set difference operator, M is the set of possible parameters which may be removed, θ_j is the parameter to be removed, and $K_t(\theta_j|k', \theta'_{1:k'})$ is the probability of picking this parameter to be removed. Where one way of selecting these probabilities which has been found to be highly effective is to use $K_t(\theta_j|k', \theta'_{1:k'}) \propto p_t\left(k, \theta'_{1:k'\setminus j}|y_{1:t}\right)$.

It is important to note that in truly sequential settings, it makes sense to only allow death moves to occur for the M parameters which have occurred up to Δt time steps in the past. The justification for this is that in some sequential examples it may be assumed that data observed at the present time t has little or no effect on the parameters estimated at distant times in the past, this has the added advantage of reducing computations.

Using the expression in (4.6), it is straightforward to show that the optimal augment-

ing kernel for the proposal kernel in (4.16) is given by

$$
\begin{aligned}
L_t^{\mathrm{opt}}(k', \theta'_{1:k'}|k, \theta_{1:k}) &= \frac{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1}) K_t(k, \theta_{1:k}|k', \theta'_{1:k'})}{p_{t-1} K_t(k, \theta_{1:k})} \qquad (4.17) \\
&= \frac{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1}) \left(\sum_{j=1}^{M} K_t(\theta_j|k', \theta'_{1:k'}) \delta_{k'-1}(k) \delta_{\theta'_{1:k'\setminus j}}(\theta_{1:k}) \right)}{\sum_{j=1}^{M} K_t(\theta_j|k', \theta'_{1:k'}) p_{t-1}(k'-1, \theta'_{1:k'\setminus j}|\mathbf{y}_{1:t-1})}.
\end{aligned}
$$

For this kernel the incremental importance weight becomes

$$
w_t(k, \theta_{1:k}; k', \theta'_{1:k'}) = \frac{p_t(k'-1, \theta'_{1:k'\setminus J}|\mathbf{y}_{1:t})}{\sum_{j=1}^{M} K_t(\theta_j|k', \theta'_{1:k'}) p_{t-1}(k'-1, \theta'_{1:k'\setminus j}|\mathbf{y}_{1:t-1})}.
$$

4.4.6 Adjustment Move

Very often the arrival of new measurements requires only small modifications to the existing parameters to improve the modelling fit, another way of understanding when this will be the case is to consider the situation in which one has a sharply peaked distribution. In this situation one can explore the distribution well by small perturbations of the parameters. Here it will be assumed that only a single parameter is chosen to be updated, and that the remaining parameters are left unaltered. The corresponding optimal augmenting kernel can be computed from the expression shown below

$$
L_t^{opt}(k', \theta'_{1:k'}|k, \theta_{1:k}) = \frac{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1}) K_t(\theta_a|k', \theta'_{1:k'})}{\int p_{t-1}(k', \theta'_{1:k'\setminus a}, \theta'_a|\mathbf{y}_{1:t-1}) K_t(\theta_a|k', \theta'_{1:k'\setminus a}, \theta'_a) d\theta'_a}.
$$

If one sets the proposal for the parameter to be updated to its posterior given all the available data as shown below

$$
K_t^{opt}(\theta_a|k', \theta'_{1:k'}) = p_t(\theta_a|\theta'_{1:k'\setminus a}, \mathbf{y}_{1:t})
$$

then the optimal augmenting kernel can be further simplified as

$$
L_t^{opt}(k', \theta'_{1:k'}|k, \theta_{1:k}) = \frac{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1})}{\int p_{t-1}(k', \theta'_{1:k'\setminus a}, \theta'_a|\mathbf{y}_{1:t-1}) d\theta'_a} \qquad (4.18)
$$

which in some cases will provide an analytical solution.

However for many cases of interest the integral in the denominator of (4.18) will be intractable hence one needs to resort to using a different proposal kernel. The next version of the adjustment move has been presented, as it provides an analytical solution to L_t^{opt} which is straightforward and has been found to be very effective in all simulations carried out. It is assumed that only a single parameter θ'_a at time t is chosen to be updated, and that the remaining parameters are left unaltered. The proposal kernel for such an update move may be expressed as

$$K_t(k, \theta_{1:k}|k', \theta'_{1:k'}) = \delta_{k'}(k)\delta_{\theta'_{1:k'\setminus a}}(\theta_{1:k\setminus a})K_t(a|k', \theta'_{1:k'})K_t(\theta_a|k', \theta'_{1:k'}),$$

where a is the index of the parameter to be adjusted, $K_t(a|k', \theta'_{1:k'})$ is the probability of picking this parameter to be updated, and $K_t(\theta_a|k', \theta'_{1:k'})$ is the kernel from which the updated parameter value is generated.

As for technique 1 in the death move, it is common to pick the parameter to be updated uniformly randomly from the set of M existing parameters. This can be further extended in a sequential "on-line" setting, in which parameters are temporally ordered so that one only selects a parameter to be adjusted which is within a window of time given by the range $[t-\Delta t, t]$. Some examples of such a situation shall be presented in the applications chapter. Now if one selects the kernel $K_t(\theta_a|k', \theta'_{1:k'})$ to be a set of weighted, randomly or deterministically spaced grid points $\{\theta'_a - s\delta_1, ..., \theta'_a + s\delta_{2s+1}\}$ such that

$$K_t(\theta_a|k', \theta'_{1:k'}) = \sum_{j=1}^{2s+1} w_j \delta_{(\theta'_a-(s-j+1)\delta_j)}(\theta_a)$$

where $w_j \propto p_t\left(k', \theta'_{1:k'\setminus a}, \theta_a = (\theta'_a - (s-j+1)\,\delta_j)\,|\mathbf{y}_{1:t}\right)$, then the corresponding optimal

augmenting kernel can again be computed, and is given by

$$
\begin{aligned}
& L_t^{\mathrm{opt}}(k', \theta'_{1:k'}|k, \theta_{1:k}) \\
= & \frac{p_{t-1}(k', \theta'_{1:k'}|\mathbf{y}_{1:t-1}) \left(\frac{1}{M} \sum_{j=1}^{2s+1} w_j \delta_{(\theta'_a - (s-j+1)\delta_j)} (\theta_a)\, \delta_{k'}(k) \delta_{\theta'_{1:k'\setminus a}}(\theta_{1:k\setminus a}) \right)}{\frac{1}{M} \sum_{j=1}^{2s+1} w_j p_{t-1}(k', \theta'_{1:k'\setminus a}, \theta'_a - (s-j+1)\, \delta_j|\mathbf{y}_{1:t-1})}.
\end{aligned}
$$

This will then produce an incremental importance weight for the adjustment move,

$$
w_t(k, \theta_{1:k}; k', \theta'_{1:k'}) = \frac{p_t(k', \theta'_{1:k'\setminus a}, \theta_a|\mathbf{y}_{1:t})}{\frac{1}{M} \sum_{j=1}^{2s+1} w_j p_{t-1}(k', \theta'_{1:k'\setminus a}, \theta'_a - (s-j+1)\, \delta_j|\mathbf{y}_{1:t-1})},
$$

which is again similar to the expression obtained for the death move using technique 2.

The moves presented are just examples and many others could be designed and used. Now that the basic settings for the transition kernel and auxiliary kernels have been established for several move types, one may construct the following generic TDSMC algorithm.

4.4.7 TDSMC Algorithm

At time $t-1$ assume that one has a set of weighted samples $\{W_{t-1}^{(i)}, k'^{(i)}, \theta'^{(i)}_{1:k'^{(i)}}\}_{i=1}^N$ that are approximately distributed according to the posterior distribution $p_{t-1}(k, \theta_{1:k}|\mathbf{y}_{1:t-1})$ which is given by the particle approximation,

$$
p_{t-1}(k, \theta_{1:k}|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} W_{t-1}^{(i)} \delta_{(k'^{(i)}, \theta'^{(i)}_{1:k'^{(i)}})}(k, \theta_{1:k}),
$$

in which $\delta_x(\cdot)$ denotes the Dirac delta function with mass at x. The TDSMC algorithm then proceeds as follows at time t.

TDSMC Generic Algorithm

- For $i = 1 \cdots N$, sample $(k^{(i)}, \theta^{(i)}_{1:k^{(i)}}) \sim K_t(\cdot|k'^{(i)}, \theta'^{(i)}_{1:k'^{(i)}})$.
- For $i = 1 \cdots N$, set the importance weights to $W_t^{(i)} \propto W_{t-1}^{(i)} w_t(k^{(i)}, \theta^{(i)}_{1:k^{(i)}}; k'^{(i)}, \theta'^{(i)}_{1:k'^{(i)}})$, and normalise such that $\sum_{i=1}^N W_t^{(i)} = 1$.

<u>*Resampling Step*</u>

- If resampling is required then for $i = 1 \cdots N$, sample an index $j(i) \sim \{w_t^{(l)}\}_{l=1}^N$, and replace $\{w_t^{(i)}, k^{(i)}, \theta^{(i)}_{1:k^{(i)}}\} \leftarrow \{N^{-1}, k^{j(i)}, \theta^{j(i)}_{1:k^{j(i)}}\}$.

The resulting set of weighted samples $\{W_t^{(i)}, k^{(i)}, \theta^{(i)}_{1:k^{(i)}}\}_{i=1}^N$ is then approximately distributed according to the posterior distribution $p_t(k, \theta_{1:k}|\mathbf{y}_{1:t})$ given next.

$$p_t(k, \theta_{1:k}|\mathbf{y}_{1:t}) \approx \sum_{i=1}^N W_t^{(i)} \delta_{(k^{(i)}, \theta^{(i)}_{1:k^{(i)}})}(k, \theta_{1:k}).$$

Hence, summarising what has been presented. An algorithm has been developed which sequentially allows one to obtain weighted samples from the sequence of distributions $(\pi_t(k, \theta_{1:k}) = p_t(k, \theta_{1:k}|y_{1:t}))$ where each distribution $p_t(k, \theta_{1:k}|y_{1:t})$ is defined on a space of the form $\Theta_t = \cup_{k \in \mathcal{K}} \{k\} \times \Theta_{k,t}$. This has been achieved by augmenting the target parameter space with an auxiliary parameter space which is associated with the samples obtained in the previous iteration. This augmented posterior was designed to admit as a marginal distribution, the target distribution of interest at iteration t, which is given by $p_t(k, \theta_{1:k}|y_{1:t})$. An algorithm was presented in which the transition kernel comprised a mixture of transition kernels, with each component transition kernel representing a different type of move within or between dimensions, such as birth, death, update and adjustment. In addition to this a computationally efficient incremental weight estimate

(4.8) which involved sampling an auxiliary random variable at each iteration t was presented. This auxiliary random variable M_t was sampled from the discrete distribution of mixture weights $\alpha_{m,t}$ presented in (4.7) and then it was associated with the type of move/transition kernel that would be used at time t for the mutation step.

Following this algorithmic section, in which the TDSMC framework has been detailed, is a section which involves an asymptotic analysis of the TDSMC algorithm, followed by a section in which these ideas have been successfully applied in two different situations.

4.4.8 Asymptotic Variance for TDSMC Algorithm

The weighted particle estimates obtained from the TDSMC algorithm are typically used to form estimates of integrals, as shown in the example below

$$\widehat{E}_{p_t}(\varphi) = \sum_{i=1}^{P} W_t^{(i)} \varphi\left(\{k, \theta_{1:k}\}_t^{(i)}\right). \tag{4.19}$$

To obtain such an estimate at time t, one first determines the mode for the marginal posterior of model order k. Then conditional on this modal or MAP model order, which shall be labelled k^*, the weights of the particles associated with k^* are renormalised. This allows one to obtain a weighted estimate, such as the one presented above in equation (4.19), where P is the number of particles associated with model order k^*.

In the same manner as [31] and the results presented in Chapter 3, an asymptotic variance expression for the estimate (4.19) has been developed. This follows the results presented for the Central Limit Theorem of ([27] and [32], section 9.4, pp. 300-306). The asymptotic variance expressions obtained are for the two extreme cases, when no resampling is used and when multinomial resampling is used at every iteration. It is important to mention that by introducing the auxiliary random variables $M_{1:t}$ in order to create an efficient means of computing the incremental weights, the parameter space has changed from $\Theta_t = \cup_{k\in\mathcal{K}}\{k\} \times \theta_{k,t}$ to $\Theta_t = \cup_{k\in\mathcal{K}}\{k\} \times \theta_{k,t} \times \mathcal{X}^t$, where $\mathcal{X} = \{1, 2, 3, ...M\}$ is the support of the auxiliary random variable M_t. Recall that this represents the set of

indexes associated with the M possible move types/mixture components in the mixture transition kernel presented in (4.7). Therefore defining the new posterior on this extended product space as $\pi_t(k_t, \theta_{1:k,t}, m_t|y_{1:t}) = p_t(k_t, \theta_{1:k,t}|m_t, y_{1:t})\, p_t(m_t|y_{1:t})$ which as one can see, still clearly admits the desired target posterior $p_t(k_t, \theta_{1:k,t}|m_t, y_{1:t})$ as a marginal. The distribution $p_t(m_t|y_{1:t})$ is just a discrete set of probabilities $\alpha_{m,t}$ over the set $\mathcal{X}$ at each iteration t.

Before stating some results which are a direct result of proposition 1 and derived from the work of (Del Moral and Guionnet,1999; Del Moral and Miclo, 2000; Kunsch, 2001; Chopin, 2004), it will prove useful to define the following notation, where the initial set of particles at time $t = 1$ are given by $\{k_1, \theta_{1:k,1}, m_1\}_{i=1:N}$ and they are assumed to be sampled from the distribution $\mu_1(k_1, \theta_{1:k,1}, m_1)$. Then at time t, if the particles have not been resampled they will be distributed according to

$$\begin{aligned} & \mu_t(k_{1:t}, \theta_{1:k,1:t}, m_{1:t}) \\ = \; & \mu_1(k_1, \theta_{1:k,1}, m_1) \textstyle\prod_{n=1}^{t} \alpha_{m,n}(k_{n-1}, \theta_{1:k,n-1})\, K_{m,n}(k_{n-1}, \theta_{1:k,n-1}; k_n, \theta_{1:k,n}) \end{aligned}$$

and if they have been resampled at time l then they will be distributed as

$$\begin{aligned} & \widehat{\pi}_t(k_{1:t}, \theta_{1:k,1:t}, m_{1:t}|y_{1:t}) \\ = \; & \pi_l(k_l, \theta_{1:k,l}, m_l) \textstyle\prod_{n=l}^{t} \alpha_{m,n}(k_{n-1}, \theta_{1:k,n-1})\, K_{m,n}(k_{n-1}, \theta_{1:k,n-1}; k_n, \theta_{1:k,n}). \end{aligned}$$

However, we would like the particles to be weighted samples from the target posterior

$$\begin{aligned} & \widetilde{\pi}_t(k_{1:t}, \theta_{1:k,1:t}, m_{1:t}|y_{1:t}) \\ = \; & \pi_t(k_t, \theta_{1:k,t}, m_t|y_{1:t}) \textstyle\prod_{n=1}^{t} \beta_{m,n-1}(k_n, \theta_{1:k,n})\, L_{m,n}(k_n, \theta_{1:k,n}; k_{n-1}, \theta_{1:k,n-1}) \end{aligned}$$

so an importance sampling correction is made. We shall also use the following notation

for the marginal posterior of the set of variables at time a

$$\widetilde{\pi}_t\left(k_a,\theta_{1:k,a},m_a|y_{1:t}\right)=\sum_{j=1:t\backslash a}\sum_{k_j=1}^{k_{\max,j}}\sum_{m_j=1}^{M}\int\widetilde{\pi}_t\left(k_{1:t},\theta_{1:k,1:t},m_{1:t}|y_{1:t}\right)d\theta_{1:k,1:t\backslash a}$$

and we shall denote the following conditional distribution $\widetilde{\pi}_t\left(k_t,\theta_{1:k,t},m_t|y_{1:t},k_a,\theta_{1:k,a},m_a\right)$ as follows

$$\begin{aligned}&\widetilde{\pi}_t\left(k_t,\theta_{1:k,t},m_t|y_{1:t},k_a,\theta_{1:k,a},m_a\right)\\=&\left[\sum_{j=1}^{t-1}\sum_{k_j=1}^{k_{\max}}\sum_{m_j=1}^{M}\int\widetilde{\pi}_t\left(k_{1:t},\theta_{1:k,1:t},m_{1:t}|y_{1:t}\right)d\theta_{1:k,1:t-1}\right]/\widetilde{\pi}_t\left(k_a,\theta_{1:k,a},m_a|y_{1:t}\right).\end{aligned}$$

Now by using Proposition 1 of this thesis which uses a combination of equations $(3,4,9)$ and Theorem 1 from [27] and the Delta method, one is able to make the following remark which is of fundamentally the same form as Proposition 1, presented in [31]. The proof of this remark is identical to that found in appendix 1.

Remark 3 *Utilising the Central Limit Theorem and associated weak integrability results presented by (Chopin, 2004) or (Del Moral, 2004, section 9.4, pp. 300-306), we are able to state the following results:*

In the case in which no resampling is used the following convergence in distribution is obtained :

$$\sqrt{N}\left(\widehat{E}_{\pi_t}\left(\varphi\right)-E_{\pi_t}\left(\varphi\right)\right)\Rightarrow\mathcal{N}\left\{0,\sigma^2_{GIS,t}\left(\varphi\right)\right\}$$

where

$$\sigma^2_{GIS,t}\left(\varphi\right)=\sum_{j=1}^{t}\sum_{k_j=1}^{k_{\max}}\sum_{m_j=1}^{M}\int\frac{\widetilde{\pi}_t\left(k_{1:t},\theta_{1:k,1:t},m_{1:t}|y_{1:t}\right)^2}{\mu_t\left(k_{1:t},\theta_{1:k,1:t},m_{1:t}\right)}\left(\varphi\left(\theta_{1:k,t}\right)-E_{\pi_t}\left(\varphi\right)\right)d\theta_{1:k,1:t}$$

and in the case in which multinomial resampling is used at every iteration, one has

$$\sqrt{N}\left(\widehat{E}_{\pi_t}(\varphi) - E_{\pi_t}(\varphi)\right) \Rightarrow \mathcal{N}\left\{0, \sigma^2_{TDSMC,t}(\varphi)\right\}$$

where, for $n \geqslant 2$ one has

$$\begin{aligned}
&\sigma^2_{TDSMC,t}(\varphi) \\
= &\sum_{k_1=1}^{k_{\max}} \sum_{m_1=1}^{M} \int A_1 \left(\int \varphi(\theta_{1:k,t}) \widetilde{\pi}_t(k_t, \theta_{1:k,t}, m_t | y_{1:t}, k_1, \theta_{1:k,1}, m_1) - E_{\pi_t}(\varphi) \right) d\theta_{1:k,1} \\
&+ \sum_{j=2}^{t-1} \sum_{k_j=1}^{k_{\max}} \sum_{m_j=1}^{M} \int A_2 \left(\int \varphi(\theta_{1:k,t}) \widetilde{\pi}_t(k_t, \theta_{1:k,t}, m_t | y_{1:t}, k_j, \theta_{1:k,j}, m_j) - E_{\pi_t}(\varphi) \right) d\theta_{j-1:j} \\
&+ \sum_{k_j=1}^{k_{\max}} \sum_{m_j=1}^{M} \int A_3 \left(\varphi(\theta_{1:k,t}) - E_{\pi_t}(\varphi) \right) d\theta_{t-1:t}.
\end{aligned}$$

with

$$\begin{aligned}
A_1 &= \frac{\widetilde{\pi}_t(k_1, \theta_{1:k,1}, m_1 | y_{1:t})^2}{\mu_1(k_1, \theta_{1:k,1}, m_1)} \\
A_2 &= \frac{\left[\widetilde{\pi}_t(k_j, \theta_{1:k,j}, m_j | y_{1:t}) \beta_{m,j-1}(k_j, \theta_{1:k,j}) L_{m,j}(k_j, \theta_{1:k,j}; k_{j-1}, \theta_{1:k,j-1})\right]^2}{\pi_{j-1}(k_{j-1}, \theta_{1:k,j-1}, m_{-1}) \alpha_{m,j}(k_{j-1}, \theta_{1:k,j-1}) K_{m,j}(k_{j-1}, \theta_{1:k,j-1}; k_j, \theta_{1:k,j})} \\
A_3 &= \frac{\left[\widetilde{\pi}_t(k_t, \theta_{1:k,t}, m_t | y_{1:t}) \beta_{m,t-1}(k_t, \theta_{1:k,t}) L_{m,t}(k_t, \theta_{1:k,t}; k_{t-1}, \theta_{1:k,t-1})\right]^2}{\pi_{j-1}(k_{t-1}, \theta_{1:k,t-1}, m_{t-1}) \alpha_{m,t}(k_{t-1}, \theta_{1:k,t-1}) K_{m,t}(k_{t-1}, \theta_{1:k,t-1}; k_t, \theta_{1:k,t})}
\end{aligned}$$

4.5 Application of TDSMC Algorithm

This section will now demonstrate an application of the TDSMC algorithm which has been published in [89] and [90]. The problem to be investigated is sequential kernel regression which will be applied in a simulated example and then to a real data set. The sequential kernel regression example presented in this chapter is adapted from earlier work of Dr. Jaco Vermaak of Cambridge University. The author must also thank Dr. Doucet for suggesting the radial basis function regression example as a good means of

testing the TDSMC methodology and for guidance in the development of this problem. The input of the author came from developing and implementing the framework for the approximation of the optimal L_t^{opt} kernel and then in this example demonstrating and developing its use. This involved carrying out more extensive simulations, which include the use of approximation of the optimal L_t^{opt} kernel, than those found in [89] and [90] .

The author has used this example to demonstrate how one may improve the choice of the kernel L_t, that was used in the two papers just cited, by using the approximation of the optimal L_t^{opt} kernel which minimises the variance of the importance weights. Hence the author of this thesis has used this example and introduced the moves developed throughout this chapter, coupled with the associated approximations for the optimal auxiliary kernels to improve the performance of the algorithm used in the two papers. This improvement is compared with the results obtained in the two papers to help demonstrate why one should attempt an approximation of the optimal L_t^{opt} kernel and when this is important. It will also give some indication of how much difference there is between a poor choice of kernel L_t and an approximation of the optimal kernel for this application.

4.5.1 Application 1: Sequential Kernel Regression

The objective of the radial basis function regression is to fit a mixture of local kernels to some unknown function of which one only has noisy samples. This example is built upon an example presented in [89].

Model Description

The aim of this section is to develop a strategy that estimates the number of kernels, k, and the parameter values, $\theta_{1:k}$, of the kernels sequentially and in a single pass over the data. Kernel regression [16] is a well studied problem and many batch strategies have been developed to estimate both the number of kernels and the parameters of the kernels [5], [60], [34]. This example provides a set up in which a sequential approach is being

used to solve what is typically a classical batch estimation problem. Hence there will be no temporal ordering to the data points, this does not affect the TDSMC algorithm to be presented.

The model for the kernel regression is defined as follows

$$y_t = \beta_0 + \sum_{i=1}^{k} \beta_i K(\mathbf{x}_t, \boldsymbol{\mu}_i) + v_t$$

with the following specifications :

- $\mathbf{x}_t \in \mathbb{R}^d$ is the input variables
- $y_t \in \mathbb{R}$is the corrupted output
- $v_t \in \mathbb{R} \quad v_t \sim \mathcal{N}(0, \sigma_y^2)$ is the *i.i.d.* Gaussian noise
- σ_y^2 is the observation noise variance
- $\beta = (\beta_0 \cdots \beta_k) \in \mathbb{R}^{k+1}$ is the regression coefficients
- $K(\cdot; \boldsymbol{\mu})$ is a local Gaussian kernel function with centre $\boldsymbol{\mu} \in \mathbb{R}^d$ and known width.

In this example it is assumed that the batch data $\{\mathbf{x}_t, y_t\}_{t=1}^T$ is available and one would like to estimate the number of kernels and the corresponding unknown regression coefficients and kernel centres which will be denoted by $\theta_{1:k}$ where $\theta_i = (\beta_i, \boldsymbol{\mu}_i)$. The kernel centres shall have an evolving support which is given by the available data points at time t. Hence the support of the unknown parameters can be written as $\Theta_{k,t} = (\mathbb{R} \times \{\mathbf{x}_1 \cdots \mathbf{x}_t\})^k$. The prior structure used in this analysis is given by the following

$$p_t(k, \theta_{1:k}) = p(k)p(\beta_0) \prod_{i=1}^{k} p(\beta_i)p_t(\boldsymbol{\mu}_i), \tag{4.20}$$

with

- $p(k) \propto \lambda^k \exp(-\lambda)/k! \quad k \in \{1 \cdots k_{\max}\}$

- $p(\beta_i) = \mathcal{N}(\beta_i|0, \sigma_\beta^2) \quad i = 0 \cdots k$
- $p_t(\boldsymbol{\mu}_i) = \mathcal{U}_{\{\mathbf{x}_1 \cdots \mathbf{x}_t\}}(\boldsymbol{\mu}_i) \quad i = 1 \cdots k.$

where σ_β^2 is a common variance and the maximum number of kernels is typically set to some value less than the total number of data points T. Additionally it will be assumed that the prior parameters $(\lambda, \sigma_\beta, \sigma_y)$ are known. The conditional independence assumption on data points coupled with the conjugate prior structure used allows the full posterior to be marginalised over the regression coefficients to obtain the following

$$p_t(k, \boldsymbol{\mu}_{1:k}|\mathbf{y}_{1:t}) \propto \frac{|\mathbf{B}|^{1/2} \exp(-\mathbf{y}^{\mathrm{T}}\mathbf{P}\mathbf{y}/2\sigma_y^2) p(k) p_t(\boldsymbol{\mu}_{1:k})}{(2\pi\sigma_y^2)^{t/2}(\sigma_\beta^2)^{k+1/2}}, \tag{4.21}$$

$$\mathbf{B} = (\mathbf{K}^{\mathrm{T}}\mathbf{K}/\sigma_y^2 + \mathbf{I}_{k+1}/\sigma_\beta^2)^{-1}$$
$$\mathbf{P} = \mathbf{I}_t - \mathbf{K}\mathbf{B}\mathbf{K}^{\mathrm{T}}/\sigma_y^2.$$

$$\mathbf{K} = \begin{bmatrix} 1 & K(\mathbf{x}_1, \boldsymbol{\mu}_1) & \cdots & K(\mathbf{x}_1, \boldsymbol{\mu}_k) \\ \vdots & \vdots & & \vdots \\ 1 & K(\mathbf{x}_t, \boldsymbol{\mu}_1) & \cdots & K(\mathbf{x}_t, \boldsymbol{\mu}_k) \end{bmatrix}.$$

where $\mathbf{y} \in \mathbb{R}^t$ is the column vector of t outputs and $\mathbf{K} \in \mathbb{R}^{t \times (k+1)}$ is the kernel regression matrix.

Given an estimate of the number of kernels and the locations of the kernel centres one can then reconstruct an estimate of the noise free data points as $\widehat{\mathbf{z}} = \mathbf{K}\mathbf{B}\mathbf{K}^{\mathrm{T}}/\sigma_y^2$. The following section will outline the algorithmic aspects of the TDSMC algorithm that shall be used in this example.

TDSMC Algorithm

Four moves shall be used in the application of the TDSMC framework to solve this problem; an update move, an adjustment move, a birth move and a death move. The

details of the moves are identical to those presented previously in this chapter, with the incremental importance weights given as follows.

Birth Move : In this model it would be possible to sample from the optimal proposal for the birth move, since the kernel centres have a discrete support, however to reduce computation the birth location is sampled uniformly over the grid of unoccupied data points.

$$K_{\text{birth},t}(\boldsymbol{\mu}_k|k', \mu'_{1:k'}) = \mathcal{U}_{\Gamma'_t}(\boldsymbol{\mu}_k),$$

where $\Gamma'_t = \{\mathbf{x}_1 \cdots \mathbf{x}_t\} \setminus \{\boldsymbol{\mu}'_1 \cdots \boldsymbol{\mu}'_{k'}\}$ is the set of unoccupied data points. At time $t-1$ one has particles given by $\{k', \boldsymbol{\mu}'_{1:k'}\}$ then at time t one has $\{k, \boldsymbol{\mu}_{1:k}\}$ where $k = k'+1$ and $\boldsymbol{\mu}_{1:k} = \boldsymbol{\mu}'_{1:k'} \cup \boldsymbol{\mu}_k$ which produces an incremental importance weight given by :

$$\begin{aligned} W_{birth} &\propto \frac{p_t\left(k, \boldsymbol{\mu}_{1:k}|y_{1:t}\right)}{p_{t-1}\left(k', \boldsymbol{\mu}'_{1:k'}|y_{1:t-1}\right) K_{\text{birth},t}\left(\boldsymbol{\mu}_k|k', \boldsymbol{\mu}'_{1:k'}\right)} \\ &= \frac{|B|^{1/2} \exp\left(-\left(y^T P y - y'^T P' y'\right)/2\sigma_y^2\right)}{\sigma_\beta |B'|^{1/2} \left(2\pi\sigma_y^2\right)^{1/2}} . \frac{\lambda (t-1)^{k'}}{t^k (k'+1)} (t-k') \end{aligned}$$

Death move : For the death move one discards a kernel selected uniformly randomly from the set of existing kernels. At time $t-1$ one has particles given by $\{k', \boldsymbol{\mu}'_{1:k'}\}$ then at time t one has $\{k, \boldsymbol{\mu}_{1:k}\}$ where $k = k'-1$ and $\boldsymbol{\mu}_{1:k} = \boldsymbol{\mu}'_{1:k'} \setminus \boldsymbol{\mu}'_d$ which produces an incremental importance weight given by :

$$\begin{aligned} W_{death} &\propto \frac{p_t\left(k'-1, \boldsymbol{\mu}_{1:k'\setminus d}|y_{1:t}\right)}{p_{t-1}\left(k'-1, \boldsymbol{\mu}'_{1:k'\setminus d}|y_{1:t-1}\right) K_t(d|k', \boldsymbol{\mu}'_{1:k'})} \\ &= \frac{\sigma_\beta |B|^{1/2} \exp\left(-\left(y^T P y - y'^T P' y'\right)/2\sigma_y^2\right)}{|B'|^{1/2} \left(2\pi\sigma_y^2\right)^{1/2}} . \frac{(t-1)^{k'} (k')^2}{\lambda t^k} \end{aligned}$$

Update move : At time $t-1$ one has particles given by $\{k', \boldsymbol{\mu}'_{1:k'}\}$ then at time t one has $\{k, \boldsymbol{\mu}_{1:k}\}$ where $k' = k$ and $\boldsymbol{\mu}'_{1:k'} = \boldsymbol{\mu}_{1:k}$ which produces an incremental importance weight given by :

$$\begin{aligned} W_{update} &\propto \frac{p_t\left(k, \boldsymbol{\mu}_{1:k} | y_{1:t}\right)}{p_{t-1}\left(k', \boldsymbol{\mu}_{1:k} | y_{1:t-1}\right)} \\ &= \frac{|B|^{1/2} \exp\left(-\left(y^T P y - y'^T P' y'\right)/2\sigma_y^2\right)}{|B'|^{1/2}\left(2\pi\sigma_y^2\right)^{1/2}} . \frac{(t-1)^{k'}}{t^k} \end{aligned}$$

Adjustment move : At time $t-1$ one has particles given by $\{k', \boldsymbol{\mu}'_{1:k'}\}$ then at time t one has $\{k, \boldsymbol{\mu}_{1:k}\}$ where $k = k'$ and $\boldsymbol{\mu}_{1:k} = \boldsymbol{\mu}'_{1:k'} \backslash \boldsymbol{\mu}'_a \cup \boldsymbol{\mu}_a = \left(\boldsymbol{\mu}'_{1:a-1}, \boldsymbol{\mu}_a, \boldsymbol{\mu}'_{a+1:k'}\right)$. In this type of move it shall be assumed that there are s or less possible grid points $\left\{\boldsymbol{\mu}_j\right\}_{j=1:s}$, to be used as the potential new adjusted kernel centre, which were selected randomly from the set $\Gamma'_t = \{\mathbf{x}_1 \cdots \mathbf{x}_t\} \setminus \{\boldsymbol{\mu}'_1 \cdots \boldsymbol{\mu}'_{k'}\} \cup \{\boldsymbol{\mu}'_a\}$. This will produce an incremental importance weight given by,

$$\begin{aligned} W_{adjustment} &\propto \frac{p_t(k', \boldsymbol{\mu}'_{1:k'\backslash a}, \boldsymbol{\mu}_a | \mathbf{y}_{1:t})}{\frac{1}{k'}\sum_{j=1}^{s} w_j p_{t-1}(k', \boldsymbol{\mu}'_{1:k'\backslash a}, \boldsymbol{\mu}_j | \mathbf{y}_{1:t-1})} \\ &= \frac{(t-1)^{k'} |B|^{1/2} \exp\left(-\left(y^T P y\right)/2\sigma_y^2\right) / \left(\left(2\pi\sigma_y^2\right)^{t/2}\left(\sigma_\beta^2\right)^{(k+1)/2}\right)}{\frac{t^k}{k'}\sum_{j=1}^{s} w_j \left|B'_j\right|^{1/2} \exp\left(-\left(y'^T P'_j y'\right)/2\sigma_y^2\right) / \left(\left(2\pi\sigma_y^2\right)^{(t-1)/2}\left(\sigma_\beta^2\right)^{(k'+1)/2}\right)} \end{aligned}$$

with the probability of a given new kernel centre position given by $w_j \propto p_t(k', \boldsymbol{\mu}'_{1:k'\backslash a}, \boldsymbol{\mu}_j | \mathbf{y}_{1:t})$.

Note that the assumption on factorisation of the likelihood no longer applies since integration over the regression coefficients has caused the data points to become dependent in the marginal posterior. Thus, the addition and deletion of kernels has an impact over all data points, such that the entire likelihood has to be evaluated for each move. As was carried out in [89], the inverse calculation required to obtain matrix $\mathbf{B}$ can be determined incrementally from the inverse obtained at the previous time step, [85].

The mixture weights for each of the moves were set in a similar fashion to [52], and for the following constraints were also imposed; for $k = 0$ only a birth move is possible and when $k = \min\{k_{\max}, t\}$ a birth move is impossible.

$$\alpha_{\text{birth}} = c \min\{1, p(k+1)/p(k)\}$$
$$\alpha_{\text{death}} = c \min\{1, p(k-1)/p(k)\}$$
$$\alpha_{\text{adjustment}} = [1 - (\alpha_{\text{birth}} + \alpha_{\text{death}})]/4$$
$$\alpha_{\text{update}} = 1 - \alpha_{\text{birth}} - \alpha_{\text{death}} - \alpha_{\text{adjustment}}$$

In the above $c \in (0, 1)$ is a parameter that tunes the relative frequency of the dimension changing moves to the adjustment and update moves. For simplicity and computational savings the probabilities for the corresponding augmenting kernels were set to the same values as the transition kernel equivalent moves. This was found to work well in the experimental evaluation. To initialise the algorithm $k = 0$ was used for all of the samples. The performance of the TDSMC algorithm was then tested on the two data sets found in [89] and [90].

TDSMC: Sinc Data set

This data set has been used in [17] and has proven to be a popular bench-mark which allows for direct comparison between TDSMC results and current algorithms in the literature. Just as was the case in [89], the data is taken to be the sinc function, $\text{sinc}(x) = \sin(x)/x$ in the interval $x \in [-10, 10]$ which was corrupted by additive Gaussian noise of standard deviation $\sigma_y = 0.1$. The kernel used is a Gaussian of $\sigma = 1.6$, the training data was taken to be 50 evenly spaced points in this interval and the test data was 1000 points also over this interval. Then in the training stage and the test stage of the TDSMC algorithm data points were presented randomly. Several simulations were carried out which involved varying the λ parameter in $(\lambda, k_{\max})$. The values used for the simulations were given by the following combinations $\{(1, 50), (2, 50), (3, 50), (4, 50), (5, 50),$

$(6, 50), (7, 50), (8, 50)\}$. These λ values were selected as they represent a range of mean values around those obtained by the algorithms being used for comparison, as found in [16]. The fraction of dimension change moves was set to $c = 0.25$.

Initially all four move types were implemented. However, it was found that for this example the adjustment move did not significantly improve the performance of the algorithm. It should be stressed that the adjustment move is an integral part of such a methodology, there are several situations in which results are significantly improved through the use of such moves. This has been demonstrated in the sequential estimation example presented in [31] and the author has applied adjustment moves to other sequential kernel regression problems and found they significantly improve the performance as will be demonstrated in the following chapter.

The results for the sinc data simulations are now presented. Figure 3 shows the average test error, as a function of the number of particles N, for the range of λ values used. These results were obtained by averaging over 50 random generations of the training data for each value of N. As expected, the error decreases with an increase in the number of particles. No significant decrease is obtained beyond $N = 250$. A typical MMSE estimate of the clean data, computed from the particles prior to resampling, is shown in Figure 4. The results are also presented in Appendix 3 with standard deviation errors provided.

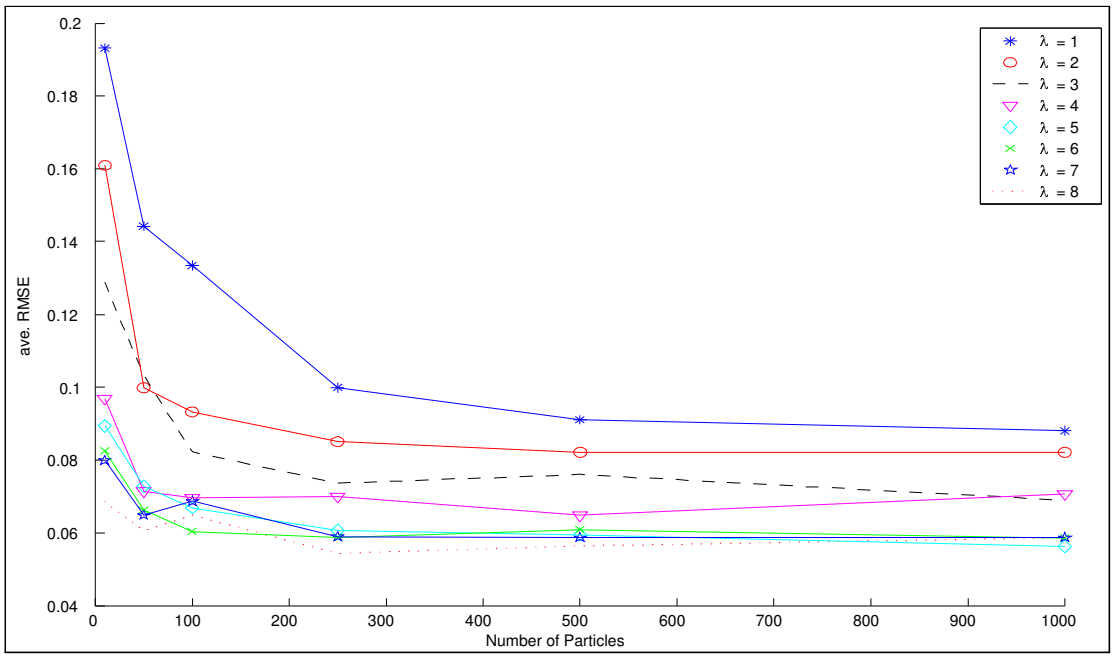

Figure 3: Average RMSE approximation error versus the number of particles for a range of λ values.

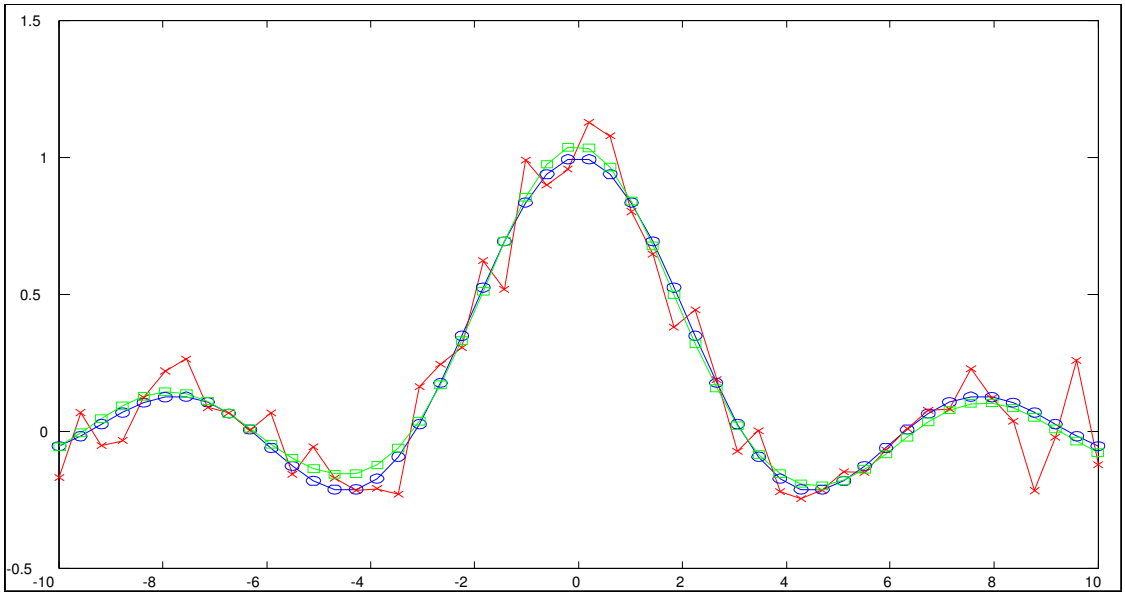

Figure 4: Blue: original uncorrupted data, Red: noisy observations, Green: typical MMSE estimate obtained

A comparison can now be made between published batch algorithms and the sequential TDSMC algorithm. The batch techniques compared are Support Vector Machine (SVM) [88] and Relevance Vector Machine (RVM) [86], [43] and the results for these algorithms are from [90]. Table 3 shows that the error for the sequential algorithm is on average slightly higher than the results of the batch algorithms. This is due to the stochastic nature of the algorithm, and the fact that it uses only very simple moves. The fact that the test error of the TDSMC algorithm is slightly larger than the batch algorithms used in this comparison should be offset against the TDSMC algorithms simplicity and significant gain in computational efficiency, data storage and the fact that it is an 'on-line' algorithm. The value of $\lambda = 5$ was used to generate the results in the following table, with $N = 1000$ particles.

Method	**Test Error**	**# Kernels**	**Noise Estimate**
Figueiredo	0.0455	7.0	-
SVM	0.0519	28.0	-
RVM	0.0494	6.9	0.0943
Variational RVM	0.0494	7.4	0.0950
MCMC	0.0468	6.5	-
Sequential RVM	0.0591	4.5	0.1136
TDSMC L_t^{opt}	0.0563	7.05	-
TDSMC [90]	0.0591	4.5	

Table 3: Comparative performance results for the sinc data.

The results in Table 3 demonstrate that the average test error for the TDSMC algorithm using the approximation of the optimal auxiliary kernel L_t^{opt} seem to only make a marginal improvement over the results obtained in [90], where the auxiliary kernel for the death move was not optimised and was supposedly selected as a degenerate delta mass. However, this is somewhat misleading since this will only be the case when large

enough numbers of particles are used such as in Table 3 where $N = 1000$. In actual fact this is in stark contrast to the performance obtained when one varies the number of particles used in the simulations for smaller values of N. The plot shown in Figure 5 below was presented in [90]. It demonstrates the performance of the TDSMC algorithm, when one does not use an optimal auxiliary kernel. Clearly in this case for small numbers of particles such as when $N \in [1, 200]$ the performance of the algorithm presented in [90] is significantly worse than the results presented for the new algorithm and shown in Figure 3, which utilises approximations to the optimal auxiliary kernel L. Hence, it can be argued that it is important, when ever possible to make a wise choice for the auxiliary kernels L_t, as this will improve results by minimising the variance of the particle weights with respect to these kernels.

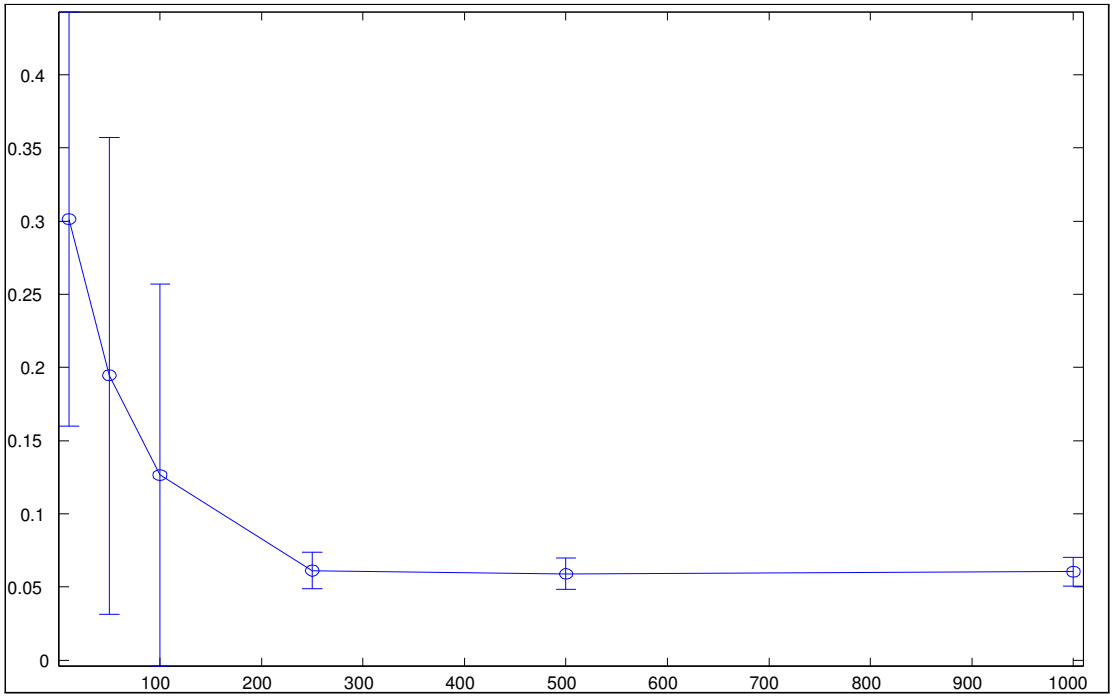

Figure 5: Blue: Average RMSE approximation error versus the number of particles, reproduced from [90].

TDSMC: Boston Housing Data

The algorithm was also applied to the popular Boston housing data set. The Boston Housing problem data can be found at StatLib at Carnegie Mellon University. The Boston house price data was first published by [54]. This is a very famous dataset in the field of statistical analysis; many have used it to prove the validity of alternative statistical techniques. It is a well known data set for testing non-linear regression methods. The data set consists of 506 cases with 14 attributes in which 12 continuous variables and 1 binary variable determine the median house price in a certain area of Boston in thousands of dollars. The prices lie between \$5000 and \$50000 in units of \$1000. There are 14 attributes in each case of the dataset, which are listed below.

x_t- elements

- CRIM - per capita crime rate by town
- ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS - proportion of non-retail business acres per town.
- CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
- NOX - nitric oxides concentration (parts per 10 million)
- RM - average number of rooms per dwelling
- AGE - proportion of owner-occupied units built prior to 1940
- DIS - weighted distances to five Boston employment centres
- RAD - index of accessibility to radial highways
- TAX - full-value property-tax rate per \$10,000
- PTRATIO - pupil-teacher ratio by town
- B - $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town

- LSTAT - % lower status of the population

y_t - value

- MEDV - Median value of owner-occupied homes in $1000's

The algorithms used for comparison were the SVM and RVM with the results taken from [86]. A random train / test partitions of the data of size 300 / 206 was used for this example. Again, a Gaussian kernel with width of 5 was used. Parameter values similar to those for the sinc experiment were used, except for setting $\lambda = 15$ to allow a larger number of kernels. The results are summarised in Table 4. These were obtained by averaging over 10 random partitions of the data, and setting the number of particles to $N = 250$. The results for a range of λ values are presented in appendix 4. The test error is comparable to those for the batch strategies, but far fewer kernels are required. It should be mentioned that this is partially due to the fact that the tests that TDSMC is being compared to used linear spline kernels, which is why they required larger numbers of kernels. However, it is clear that using the TDSMC algorithm makes for a computationally efficient and sequential alternative to these batch strategies. Additionally it is interesting to note that in this example again there is no significant difference between the performance of both forms of the TDSMC algorithm in terms of the average test error. However, use of the optimal auxiliary kernel has clearly reduced the number of kernel basis functions required to achieve approximately the same level of accuracy.

Method	**Test Error**	**# Kernels**
SVM	8.04	142.8
RVM	7.46	39.0
TDSMC L_t^{opt}	7.96	8.6
TDSMC [90]	7.18	25.29

Table 4: Comparative performance results for the Boston housing data.

4.6 Summary

This chapter introduced the TDSMC algorithm to perform joint model order determination and parameter estimation using sequential data. The algorithm is non-iterative, and based on a generalisation of importance sampling to spaces of variable dimension. The methodology underpinning the TDSMC algorithm was developed and several move types described. An asymptotic analysis of the variance of the Importance weights was presented. Two examples were analysed using TDSMC and comparisons to existing batch techniques were made, both for simulated data and for a real data set, each of which can be considered to be a bench mark data set. These examples demonstrated the problem of sequential kernel regression on a batch data set and hence provided an example of sequential analysis being used to solve classical batch estimation problems. The TDSMC algorithm was able to achieve results that compare favourably with a variety of batch algorithms. These results are even more remarkable considering the fact that the TDSMC algorithm is non-iterative, requiring only a single pass over the data. Furthermore, only a small number of Monte Carlo samples were used. This gives the TDSMC algorithm a computational advantage over batch algorithms, since its processing time can be guaranteed, while batch algorithms are inherently iterative.

Chapter 5

Applications

This chapter develops two detailed applications of the TDSMC methodology presented in Chapter 4. The first application is the estimation of an inhomogeneous Poisson process rate using a simple piecewise constant function approximation. The unknown elements will be the rate over a given segment and the number of segments. The algorithm developed was then applied to the popular real data set for coal mine disasters in the UK between 1851 and 1962.

The second example involves sequential basis function regression for the General Linear Model. TDSMC was developed to perform sequential basis function regression using an exponential basis function. In this example the parameters of the exponential and the number of exponentials present are unknowns.

5.1 Inhomogeneous Poisson Processes

Inhomogeneous Poisson processes are used in many fields to model a vast number of different phenomena. For example, they have been used to model claim occurrence epochs in a risk model [18], in applications such as finance. The Bayesian formulation of the model for an inhomogeneous Poisson process was used in [52], to analyse a data set which contained the dates of coal mining disasters in the UK.

5.1.1 Construction and Conditions for an Inhomogeneous Poisson Process

The following provides a brief review of which conditions a point process must obey in order for it to be considered a temporal Poisson process. As stated in [82] where they are rigorously presented, the reason for presenting these conditions is that the degree to which a Poisson process can be considered a reasonable model for a given application can be judged by the degree to which the following conditions are satisfied.

- **Orderliness :** Counting process $\{N(t) : t \geq t_0\}$ is orderly at $t \geq t_0$ if for any given ε, there exists a $\delta = \delta(t, \varepsilon) > 0$ such that,

$$\Pr\left(N\left(t, t+\delta'\right) > 1\right) \leqslant \varepsilon \Pr\left(N\left(t, t+\delta'\right) = 1\right)$$

  This condition can be interpreted as stating that points may not arrive simultaneously.

- **Evolution without after effects :** A point process evolves without after effects if the realisation of points in the interval $[t_j, \infty)$ is independent of the points that occurred in the interval $[t_i, t_j]$. This characterises a notion of independence of time increments.

Now one may define the Poisson process, which is an integer-valued stochastic process $\{N_t\}_{t \geq 0}$ in continuous time with rate or intensity parameter $\lambda(t)$ if

- $N_t \sim \mathcal{P}(\lambda t) : \Pr(N_t = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$,$k = 0, 1, 2....$
- N_t has independent increments on disjoint intervals

To obtain the inhomogeneous Poisson process one extends the above definition of a Poisson process in which the rate parameter is constant over time to that of a Poisson process in which the rate parameter is now time varying. The following construction is

required. Assume that one has a measurable function $\lambda(t) \geq 0$ on $[0, \infty)$, then one can define the following expression :

$$P(N_{t+h} = N_t + k | N_t = x) = \begin{cases} 1 - \lambda(t) h + o(h) & k = 0 \\ \lambda(t) h + o(h) & k = 1 \end{cases}$$

This expression defines an inhomogeneous Poisson process N_t which has a rate function given by $\lambda(t)$. This provides a natural interpretation of the distribution of the number of jumps N_A that have occurred at times from a Borel set $A \subset R$ which is given by $\mathcal{P}(\Lambda(A))$ where $\Lambda(A)$ represents the integrated rate on this Borel set, given by

$$\Lambda(A) = \int_A \lambda(t)\, dt.$$

5.1.2 Bayesian Model for Estimation of the Rate of an Inhomogeneous Poisson Process

A Bayesian model for the simple function piecewise constant non-parametric estimation of the intensity function of an inhomogeneous Poisson process can be presented as follows. One is assumed to have a data set which was generated by a process which satisfies the requirements that were presented earlier. This restriction is necessary for the process to be an inhomogeneous Poisson process. Consider the model, as used by [9], in which at time t, one has access to time occurrences which are assumed to follow an inhomogeneous Poisson process of intensity $\lambda : \mathbb{R}^+ \to \mathbb{R}^+$; that is the likelihood of l_t time occurrences is given in (5.1) by

$$p_t\left(y_{1:l_t} | \{\lambda(u)\}_{u \leq t}\right) = \exp\left(-\int_0^t \lambda(u)\, du\right) \prod_{l=1}^{l_t} \lambda(y_l). \tag{5.1}$$

In order to carry out any form of estimation of the rate function $\lambda(t)$ one must first decide on how the rate function will be represented, either as a parametric approximation

or as a non-parametric estimation. For this analysis the author has followed the representation of $\lambda(t)$ used in [9] which is a non-parametric simple function approximation of the rate function. This simple function approximation takes the form

$$\lambda(t) = \sum_{m=1}^{k} \lambda_m \mathbb{I}_{[\tau_{m-1},\tau_m)}(t) + \lambda_{k+1}\mathbb{I}_{[\tau_k,\infty)}(t)$$

with $\tau_0 = 0$. This corresponds to the left end-point simple function approximation of the rate function. This approximation of the rate function may be parameterised using the variables $\{k, \tau_{1:k}, \lambda_{1:k+1}\}$. Here k represents the model order, that is the number of segments in the approximation and $\tau_{1:k}$ represents the knot points which correspond to the times at which these k intervals begin and $\lambda_{1:k+1}$ represents the rates or amplitudes for the corresponding intervals. All of these variables are assumed unknown. The objective is to estimate the full posterior distribution,

$$p(k, \tau_{1:k}, \lambda_{1:k+1}|y_{1:l_t}) = \frac{p(y_{1:l_t}|k, \tau_{1:k}, \lambda_{1:k+1})\, p(k, \tau_{1:k}, \lambda_{1:k+1})}{p(y_{1:l_t})}.$$

Now the likelihood under this parameterisation takes the form

$$p_t(y_{1:l_t}|k, \tau_{1:k}, \lambda_{1:k+1}) = \prod_{m=1}^{k+1} (\lambda_m)^{L(m)} \exp\left(-\sum_{m=1}^{k} \lambda_m [\tau_m - \tau_{m-1}] - \lambda_{k+1}[t - \tau_k]\right).$$

Where here $L(m)$ represents the number of observations that arrived in the interval $[\tau_{m-1}, \tau_m]$ for $m \leq k$ and for $m = k+1$ then $L(m)$ is the number of observations that arrived in the interval $[\tau_k, t]$. In this sequential analysis the following time-dependent prior distribution on the unknown parameters was used

$$p_t(k, \lambda_{1:k+1}, \tau_{1:k}) = p_t(k)\, p_t(\lambda_{1:k+1}|\, k)\, p_t(\tau_{1:k}|\, k)$$

where $p_t(k)$ is a Poisson distribution of parameter $\lambda_q t$, $p_t(\tau_{1:k}|\, k)$ is the distribution of

a vector of uniform order statistics on $[0, t)$ and

$$p_t\left(\lambda_{1:k+1} \middle| k\right) = p\left(\lambda_1\right) \prod_{m=2}^{k+1} p\left(\lambda_m \middle| \lambda_{m-1}\right)$$

where $\lambda_1 \sim \mathcal{G}a\left(\mu, \upsilon\right)$ and $\lambda_m \middle| \lambda_{m-1} \sim \mathcal{G}a\left(\lambda_{m-1}^2/\chi; \lambda_{m-1}/\chi\right)$; μ, υ, χ are parameters specified by the user. The prior model given to the intensity λ_m of the rate function over an interval $[\tau_{m-1}, \tau_m]$ was defined as being conditional on the previous intensity λ_{m-1} on the interval directly proceeding the mth, as given by $p_t\left(\lambda_m | \lambda_{m-1}\right)$. Now in [52] the prior structure for the intensities was an independent Gamma distribution. However, the author argues that in a sequential setting and for the sake of continuity of the rate function in such a setting, it makes sense to base the current intensity of the rate function at time t on previous intensities, which is why a conditional prior for the intensity was used. The height prior therefore took the form shown above which specifies the mean of the Gamma distribution to be λ_{i-1} and the variance of the Gamma distribution to be χ.

Combining this prior structure with the likelihood presented, allows one to define the sequence of posterior distributions over times $n\Delta T$ that will be of interest in this analysis which are given by

$$\begin{aligned}
\pi_n\left(k, \lambda_{1:k+1}, \tau_{1:k}\right) &= p_{n\Delta T}\left(k, \lambda_{1:k+1}, \tau_{1:k} \middle| y_{1:l_{n\Delta T}}\right) \\
&\propto p_{n\Delta T}\left(y_{1:l_{n\Delta T}} \middle| k, \lambda_{1:k+1}, \tau_{1:k}\right) p_{n\Delta T}\left(k, \lambda_{1:k+1}, \tau_{1:k}\right)
\end{aligned}$$

where ΔT is a time interval defined by the user. These distributions are defined on $\Theta = \cup_{k=0}^{\infty} \{k\} \times \vartheta_k$ where $\vartheta_k = \left\{\tau_{1:k} \in \mathbb{R}^k; 0 < \tau_1 < \ldots < \tau_k\right\} \times \left(\mathbb{R}^+\right)^{k+1}$, with the support of π_n being reduced to the subset $\left\{\tau_{1:k} \in \mathbb{R}^k; 0 < \tau_1 < \ldots < \tau_k < n\Delta T\right\} \times \left(\mathbb{R}^+\right)^{k+1}$.

This is a problem in which the number of unknowns is itself unknown. To sample from one of these distributions, a standard approach would consist of using a Reversible Jump MCMC algorithm [52]. Instead it is proposed here to sample from the sequence of distributions using SMC samplers methodology in the form of the TDSMC algorithms

presented in Chapter 4. At each time step, a mixture of four different moves was considered and the next section explains the construction and selection of the moves.

5.1.3 Moves Used In Sequential Estimation of the Underlying Rate Function in an Inhomogeneous Poisson Process

The estimation of the sequence of distributions $(\pi_n(k, \lambda_{1:k+1}, \tau_{1:k}))$, using the TDSMC framework, used update, birth, death and adjustment moves. The ith particle, at time n, in this analysis will contain a realisation of the random variables denoted by $\{k, \tau_{1:k}, \lambda_{1:k}\}_n^{(i)}$.

Update Move

No change to the parameters is made by this move which has a transition kernel given by

$$K_{n,1}\left((k, \lambda_{1:k+1}, \tau_{1:k}), \left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)\right) = \delta_{k,\lambda_{1:k+1},\tau_{1:k}}\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)$$

and $\{k, \tau_{1:k}, \lambda_{1:k}\}_n^{(i)} = \{k, \tau_{1:k}, \lambda_{1:k}\}_{n-1}^{(i)}$. The TDSMC generalised incremental importance weight for an update move is given by the expression

$$\begin{aligned}
& \frac{\pi_n\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)}{\pi_{n-1}\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)} \\
\propto\ & \frac{p_{n\Delta T}\left(y_{1:l_{n\Delta T}} \middle| k, \lambda_{1:k+1}, \tau_{1:k}\right) p_{n\Delta T}\left(k, \lambda_{1:k+1}, \tau_{1:k}\right)}{p_{(n-1)\Delta T}\left(y_{1:l_{(n-1)\Delta T}} \middle| k, \lambda_{1:k+1}, \tau_{1:k}\right) p_{(n-1)\Delta T}\left(k, \lambda_{1:k+1}, \tau_{1:k}\right)} \\
=\ & (\lambda_{k+1})^{L_n(k+1)} \exp\left(-\lambda_{k+1}\Delta T\right) \exp\left(-\lambda_q \Delta T\right)
\end{aligned}$$

where $L_n(k+1)$ is the number of observations that have occurred in the interval $[(n-1)\Delta T, n\Delta T]$.

Birth Move

As described earlier, when carrying out a birth move it is ideal if one can construct it in such a manner that the optimal importance sampling distribution is used. For the model described in this example it was possible to obtain a good approximation with a simple

form, due to the conjugacy introduced. The approximation of the optimal importance density means that the importance distribution is not blind to the information contained by the data. The optimal importance distribution for the birth move is obtained from minimisation of the variance of the birth step incremental weight, as discussed in Chapter 4. An approximation of this importance distribution is now presented. Assume that at time $(n-1)\,\Delta T$ one has $\{k, \tau_{1:k}, \lambda_{1:k+1}\}$ and that after the birth move, at time $n\Delta T$, one has $\{k+1, \tau_{new}, \tau_{1:k}, \lambda_{new}, \lambda_{1:k+1}\}$. In the birth step constructed, the first k components of the particle undergoing a birth step remain unaltered since it makes the problem simpler computationally and it was found that adjustment moves were adequate if changes with previous parameters were required. The birth transition kernel will be expressed as

$$\begin{aligned} & K_{n,2}\left(\left(k, \lambda_{1:k+1}, \tau_{1:k}\right), \left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)\right) \\ = \; & \delta_{k+1,\lambda_{1:k+1},\tau_{1:k}}\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right) q_n\left(\left(\lambda_{1:k+1}, \tau_{1:k}\right), \left(\lambda'_{k+2}, \tau'_{k+1}\right)\right) \end{aligned}$$

where the appended component $\left(\lambda'_{k+2}, \tau'_{k+1}\right) = \left(\lambda_{new}, \tau_{new}\right)$ is sampled according to a proposal distribution $q_n\left(\left(\lambda_{1:k}, \tau_{1:k}\right), \cdot\right)$. In this case the incremental weight is given by

$$\frac{\pi_n\left(k', \lambda'_{1:k'+2}, \tau'_{1:k'+1}\right)}{\pi_{n-1}\left(k, \lambda_{1:k+1}, \tau_{1:k}\right) q_n\left(\left(\lambda_{1:k+1}, \tau_{1:k}\right), \lambda'_{k+2}, \tau'_{k+1}\right)}.$$

It was established in Chapter 4, that the proposal q_n^{opt} minimizing the variance of this incremental weight, given $(\lambda_{1:k+1}, \tau_{1:k})$, is of this form

$$\begin{aligned} & \pi_n\left(\lambda'_{k+2}, \tau'_{k+1} \middle| k+1, \lambda_{1:k+1}, \tau_{1:k}\right) \\ = \; & p_{n\Delta T}\left(\lambda'_{k+2}, \tau'_{k+1} \middle| y_{1:l_{n\Delta T}}, k+1, \lambda_{1:k+1}, \tau_{1:k}\right) \\ = \; & \frac{p_{n\Delta T}\left(y_{1:l_{n\Delta T}} | \lambda'_{k+2}, \tau'_{k+1}, \lambda_{1:k+1}, \tau_{1:k}\right) p_{n\Delta T}\left(\lambda'_{k+2}, \tau'_{k+1} | \lambda_{1:k+1}, \tau_{1:k}, k+1\right)}{p_{n\Delta T}\left(y_{1:l_{n\Delta T}} | \lambda_{1:k+1}, \tau_{1:k}, k+1\right)} \\ = \; & \frac{p_{n\Delta T}\left(y_{L':l_{n\Delta T}} | \lambda'_{k+2}, \tau'_{k+1}\right) p_{n\Delta T}\left(\lambda'_{k+2} | \lambda_{1:k+1}\right) p_{n\Delta T}\left(\tau'_{k+1} | \tau_{1:k}\right)}{\int\int p_{n\Delta T}\left(y_{L':l_{n\Delta T}} | \lambda'_{k+2}, \tau'_{k+1}\right) p_{n\Delta T}\left(\lambda'_{k+2} | \lambda_{1:k+1}\right) p_{n\Delta T}\left(\tau'_{k+1} | \tau_{1:k}\right) d\lambda'_{k+2} d\tau'_{k+1}} \end{aligned}$$

where L' is the first arrival/observation in the interval $\left[\tau'_{k+1}, n\Delta T\right]$ and the conditioning

on $k+1$ is suppressed for notational convenience. In this model the prior $p_t(\lambda_{k+2}|\lambda_{k+1})$ is known to take the form of a Gamma distribution. Observe now that conditional on knowledge of $\tau_{1:k}$ the occurrence times of the previous segment starting times, the distribution of τ_{k+1} will be an exponential distribution with rate λ_q. This is summarised by

- $p_t(\lambda_{k+2}|\lambda_{k+1}) = \mathcal{G}a\left(\lambda_{k+2}; \alpha = \frac{\lambda_{k+1}^2}{\mathcal{X}}, \beta = \frac{\lambda_{k+1}}{\mathcal{X}}\right)$
- $p_t(\tau_{k+1}|\tau_{1:k}) = \exp(\tau_{k+1}; \lambda_q)$.

Note that it does not make very much sense to sample a new segment time τ_{k+1}, at a time which is greater than $n\Delta T$ as there will not yet be observations to support such a sample. Hence the time τ_{k+1} will instead be sampled from a truncated exponential distribution which will be restricted to the interval $[\tau_k, n\Delta T]$. This results in a truncated exponential distribution, which can be sampled from via rejection sampling.

Now if one were to use this model for the birth kernel it would require two things to be possible. First one would need to be able to sample from $p_{n\Delta T}\left(\lambda'_{k+2}, \tau'_{k+1} \middle| y_{1:l_{n\Delta T}}, k+1, \lambda_{1:k+1}, \tau_{1:k}\right)$ and then one would also need to be able to solve the normalising constant,

$$\begin{aligned}
& \int\int p_{n\Delta T}\left(y_{L':l_{n\Delta T}}|\lambda'_{k+2}, \tau'_{k+1}\right) p_{n\Delta T}\left(\lambda'_{k+2}|\lambda_{1:k+1}\right) p_{n\Delta T}\left(\tau'_{k+1}|\tau_{1:k}\right) d\lambda'_{k+2} d\tau'_{k+1} \\
= & \int\int \left[\left(\lambda'_{k+2}\right)^{\lceil l_{n\Delta T} - L' \rceil} \exp\left(\lambda'_{k+2}\left[n\Delta T - \tau'_{k+1}\right]\right)\right] \\
& \times p_{n\Delta T}\left(\lambda'_{k+2}|\lambda_{1:k+1}\right) p_{n\Delta T}\left(\tau'_{k+1}|\tau_{1:k}\right) d\lambda'_{k+2} d\tau_{k+1} \\
= & \int\int \left[\left(\lambda'_{k+2}\right)^{\lceil l_{n\Delta T} - L' \rceil} \exp\left(\lambda'_{k+2}\left[n\Delta T - \tau'_{k+1}\right]\right)\right] \\
& \times \mathcal{G}a\left(\lambda'_{k+2}; \alpha, \beta\right) \exp\left(\tau'_{k+1}; \lambda_q\right) d\lambda'_{k+2} d\tau'_{k+1}
\end{aligned}$$

It is difficult to sample from this optimal distribution and the normalising constant of this importance distribution is not easily solved. However an approximation may be obtained which has been found to work effectively in practice, as will be demonstrated in the examples section. This approximation is to use the alternative formulation for a

Poisson likelihood which involves numbers of counts in an interval, $[(n-1)\,\Delta T, n\Delta T]$, which will be labelled N_n, as opposed to the times of occurrence of the counts in the above formulation. This formulation contains slightly less information, however has the big advantage of allowing one to calculate the normalising constants, and as will be demonstrated in the simulations, works effectively. Hence the new formulation for the approximation of the optimal importance sampling distribution will use a likelihood of the following form

$$p\left(N_{1:n}|\lambda\right) = \left[\prod_{s=1}^{n} \frac{1}{N_s!}\left(\lambda\right)^{N_s} \exp\left(\lambda\right)\right]$$

where $N_{1:n}$ are the observation counts for the intervals associated with rate λ.

Now the next step to note is that if, as shown in [48], one has a vector $N_{1:n}$ of *i.i.d.* counts of observations from a Poisson process over a segment of time $[t_1, t_n]$, in which the rate function is constant and n is the number of integer segments in this interval. One also has a prior distribution for this rate being a Gamma distribution $\mathcal{G}a\left(\lambda;\alpha,\beta\right)$, then the natural conjugacy allows one to obtain the fact that $\lambda|N_{1:n} \sim \mathcal{G}a\left(\lambda;\alpha^*,\beta^*\right)$, where $\alpha^* = \alpha + n\overline{N}$ and $\beta^* = \beta + n$. Note that N_i represents the number of counts which have occurred in the ith integer time segment $[t_i, t_i+1]$. One may now use the ideas presented to produce an approximation of the optimal importance sampling distribution, however first a few more definitions are required.

Assume a time segment $[\tau_k, n\Delta T]$ in which τ_{k+1} is sampled, and the simple function approximation of the true rate function is λ_{k+1} over $[\tau_k, \tau_{k+1})$ and λ_{k+2} over $[\tau_{k+1}, n\Delta T]$, where λ_{k+1} and τ_k are known. However the value of τ_{k+1} is unknown, so it must be sampled first. In this situation one has $y_{L':l_{n\Delta T}}$ which represents the observations over the interval $[\tau_{k+1}, n\Delta T]$ for the occurrence times. In order to use the new likelihood formulation these occurrence time observations need to be changed into counts over the interval $[\tau_{k+1}, n\Delta T]$. In situations in which τ_{k+1} is not an integer time one should just approximate and use the integer part $\lfloor\tau_{k+1}\rfloor$. Now assuming there are s integer segments in the interval $[\lfloor\tau_{k+1}\rfloor, n\Delta T]$, then one will now obtain the reformulated observation sequence $\widehat{N}_{1:s}$. The count of the arrivals in the ith integer segment is given by $\widehat{N}_i$, the

total number of segments being s. Hence, one has converted the arrival times $y_{L':l_{n\Delta T}}$ into the number of counts of arrivals $\widehat{N}_{1:s}$ in each integer time interval between $[\lfloor \tau_{k+1} \rfloor, n\Delta T]$, where $\sum_{i=1}^{s} \widehat{N}_i \simeq l_{n\Delta T} - L'$.

So now using the new likelihood formulation presented, coupled with the fact that the model structure uses a prior for λ_{k+2} which is $\mathcal{G}a\left(\lambda_{k+2}; \alpha = \frac{\lambda_{k+1}^2}{\mathcal{X}}, \beta = \frac{\lambda_{k+1}}{\mathcal{X}}\right)$ and a modified prior for τ_{k+1} which is the truncated exponential on the interval $[\tau_k, n\Delta T]$, then the following simplifications can be achieved

$$\frac{p_{n\Delta T}\left(y_{L':l_{n\Delta T}}|\lambda'_{k+2}, \tau'_{k+1}\right) p_{n\Delta T}\left(\lambda'_{k+2}|\lambda_{1:k+1}\right) p_{n\Delta T}\left(\tau'_{k+1}|\tau_{1:k}\right)}{\int\int p_{n\Delta T}\left(y_{L':l_{n\Delta T}}|\lambda'_{k+2}, \tau'_{k+1}\right) p_{n\Delta T}\left(\lambda'_{k+2}|\lambda_{1:k+1}\right) p_{n\Delta T}\left(\tau'_{k+1}|\tau_{1:k}\right) d\lambda'_{k+2} d\tau'_{k+1}}$$

$$\approx \frac{\widehat{p}_{n\Delta T}\left(\widehat{N}_{1:s}|\lambda'_{k+2}, \tau'_{k+1}\right) p_{n\Delta T}\left(\lambda'_{k+2}|\lambda_{1:k+1}\right) p_{n\Delta T}\left(\tau'_{k+1}|\tau_{1:k}\right)}{\int\left[\int \widehat{p}_{n\Delta T}\left(\widehat{N}_{1:s}|\lambda'_{k+2}, \tau'_{k+1}\right) p_{n\Delta T}\left(\lambda'_{k+2}|\lambda_{1:k+1}\right) d\lambda'_{k+2}\right] p_{n\Delta T}\left(\tau'_{k+1}|\tau_{1:k}\right) d\tau'_{k+1}}$$

In order to proceed, a two stage process is developed to sample from the approximation of the optimal importance distribution. Stage one involves first sampling the new knot point τ'_{k+1} from the truncated exponential as discussed above, this is unfortunately not based on information from the observations. Then based on the newly sampled value τ'_{k+1} one can use the conjugacy argument presented above to sample the new rate λ'_{k+2} from $\mathcal{G}a\left(\lambda'_{k+2}; \alpha + sE\left[\widehat{N}_{1:s}\right], \beta + s\right)$, where $\alpha = \frac{\lambda_{k+1}^2}{\mathcal{X}}, \beta = \frac{\lambda_{k+1}}{\mathcal{X}}$, and the sampling of this new rate will be based on information from the observations.

Death Move

Given $(k+1, \lambda_{1:k+2}, \tau_{1:k+1})$, the third move is a death move where it is proposed to remove a knot (λ_{J+1}, τ_J) among the D most recent knots to obtain $\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right) = (k, \lambda_{1:k+2} \setminus \{\lambda_{J+1}\}, \tau_{1:k+1} \setminus \{\tau_J\})$. The value of J is sampled according to a uniform distribution $J \sim \mathcal{U}(\{k-D+1, \ldots .k\})$. In this case as was demonstrated in Chapter 4 for the death move, the resulting incremental importance weight is given by

$$\frac{\pi_n\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)}{\pi_{n-1}\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right) D^{-1}}.$$

Height Adjustment Move

Finally the fourth and last move is a height adjustment move, which was constructed in order to introduce some diversity into the sample paths over time and also to allow the particles to adjust the intensity or "height" of one of the segments. The aim of this move was to improve the simple function approximation of the underlying rate function, as more information is received.

In the height adjustment move, the current value of an intensity parameter in the recent past is modified to obtain $\left(k', \lambda'_{1:k'+1}, \tau'_{1:k'}\right)$. Here J is sampled according to a uniform distribution, $J \sim \mathcal{U}\left(\{k-D+1, \ldots .k\}\right)$, and the new intensity λ'_{J+1} according to a discrete probability distribution with support $\{\lambda_{J+1} - s\delta, \lambda_{J+1} - (s-1)\delta, ..., \lambda_{J+1} + s\delta\}$, where s and δ are specified by the user. The discrete proposal distribution for the intensity to be adjusted is given by

$$q_n\left(\lambda_{J+1}, \lambda'_{J+1}\right) \propto \pi_n\left(k, \lambda_{1:k+1} \backslash \{\lambda_{J+1}\}, \lambda'_{J+1}, \tau_{1:k}\right)$$

and the resulting incremental importance weight using the approximation of the optimal L kernel and the framework introduced in Chapter 4, is given by

$$\frac{\pi_n\left(k, \lambda_{1:k+1} \backslash \{\lambda_{J+1}\}, \lambda'_{J+1}, \tau_{1:k}\right)}{\sum_i q\left(\lambda_{J+1}, \lambda_{J+1} - (s-i+1)\delta\right) \pi_{n-1}\left(k, \lambda_{1:k+1} \backslash \{\lambda_{J+1}\}, \lambda_{J+1} - (s-i+1)\delta, \tau_{1:k}\right)}. \tag{5.2}$$

The expressions for the importance weights of each move just presented do not include the move probabilities. The simulations were run using a birth probability $\alpha_{n,2}(k) = c\min\left(1, \frac{\lambda_q t}{k+1}\right)$, a death probability $\alpha_{n,3}(k) = c\min\left(1, \frac{k}{\lambda_q t}\right)$, a height adjustment probability $\alpha_{n,4} = 0.15$ and finally the probability of not moving was selected such that the probabilities sum to 1. The constant c was selected as large as possible under the constraint that $\alpha_{n,2}(k) + \alpha_{n,3}(k) \leqslant 0.85$. These probabilities correspond to the terms $\alpha_{n,m}$ appearing in the mixture proposal (4.7). Using the expression (4.8) with $\beta_{n,m} = \alpha_{n,m}$ in this example provided computational savings and satisfactory results so there was no need to approximate the optimal $\beta_{n,m}$ given in (4.9). The next section

presents application of this algorithm to the analysis of simulated and real data sets. The parameters used for the simulations which were the same for all simulations are $\delta = 0.2$, $s = 4$, $D = 4$.

5.1.4 Simulation Examples

Example 1: Estimation of the Rate of an Exponential Inhomogeneous Poisson Process Rate

Now that all the moves used have been specified and explained, one may now consider some applications of this model. The first application involves data which is a set of arrival times generated from an exponential rate function. The method used to generate this observation data was the thinning method of Lewis and Shedler as described in [35] which is presented below.

Generation of observation sequences :

Initialisation :

$T = 0$

$k = 0$

Repeat

Generate Z, the first event in a Poisson process from which

one can sample with rate function μ occurring after T.

Set $T = Z$

Generate a uniform $[0, 1]$ random variable U.

If $U \leq \frac{\lambda(Z)}{\mu(Z)}$

$k = k + 1$

$y_k = T$.

end

Until reach end of time limit on which want to run simulation

---

The inhomogeneous Poisson Process rate function used in the first simulation was an exponential rate function with the following expression

$$\lambda(x) = -10exp(-(1/100)x) + 11.5 \tag{5.3}$$

The first result to present for this model is a simple analysis which demonstrates that the moves made are sensible. Figure 6 shows the simple function approximation of the underlying inhomogeneous Poisson Process rate function obtained using the MAP estimate for the full posterior distribution at time point $T = 100$. It is important to mention that during the simulation the effective sample size was well behaved with an average effective sample size during this simulation of 39%. The following specifications were used for this simulation; $T = 100$, $N = 100$, $\lambda_q = 1/8$, $\mathcal{X} = 1/4$, $\Delta T = 1$, $\mu = 9/2$, $\upsilon = 3/2$, and $E_{ff} = 30\%$. Figure 6 presents a comparison of the reconstructed simple piecewise constant rate function approximation for the MAP estimate at time $T = 100$ and the true exponential inhomogeneous Poisson Process rate function.

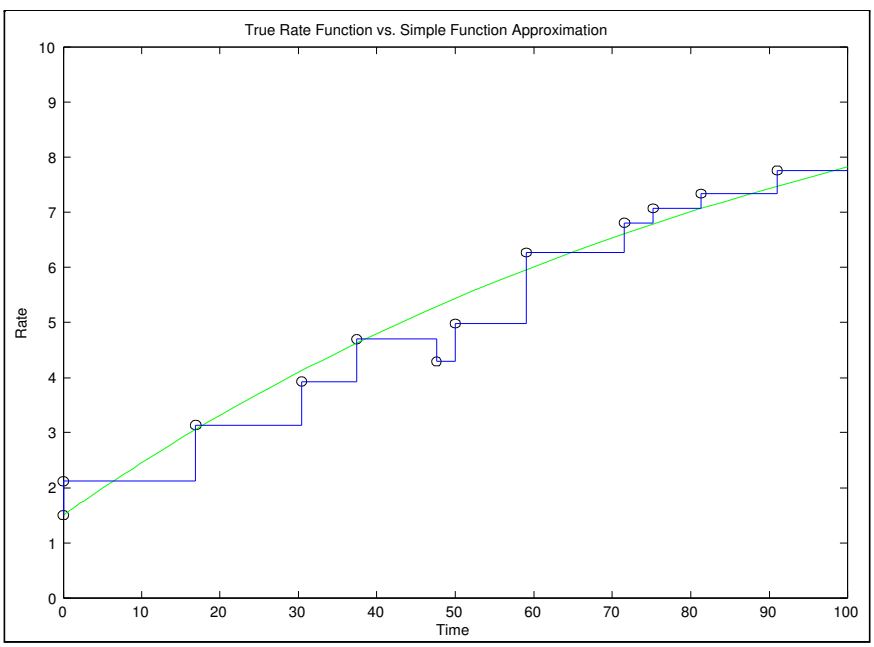

Figure 6: Simple Function Approximation of Inhomogeneous Poisson Process Rate Function vs. True Rate Function

The results to be presented next demonstrate the behaviour as the number of particles is increased. Tables 5, 6 and 7 present results for the same simulation scenario discussed above, using a rate function which is again given by (5.3). There is two ways in which one may calculate the MMSE estimate. The first involves taking the MAP estimate for the model order which is the number of piece-wise linear segments k^* associated with the largest particle weight. Then all the particles which have this model order have their weights renormalised within this batch of particles and the MMSE estimate is calculated, this corresponds to model selection. The second approach to calculating the MMSE rate at a given time t involves taking an average of the rate. This is obtained by taking each particles estimated rate at time t then averaging these rates using the corresponding particle weights, this corresponds to the case in which one performs model averaging. This case was used for this section.

If the algorithm is performing well, one would expect that as the number of particles is increased, the MMSE estimate should become smoother and better approximate the underlying rate function. Additionally, if one looks at the average absolute error between the MMSE estimator at each time step and the true rate function then this error should decrease and stabilise as the number of particles used increases. It should also be mentioned that during the simulations it was noted that depending upon the mixture probabilities used for the moves for each particle, the percentage average effective sample size can be between an average of 30% and 79%, hence this is something to keep in mind when designing algorithms using this methodology.

These runs were carried out in Matlab V6.5 with a seeded random number generator which ensured that the data samples being analysed in each example were the same for each simulation. There were three different seeds (rand('state',1) | rand('state',8) | rand('state',16)) used to obtain three different data sets for the observational data and the choice of seed was not important. This will allow a rigorous comparison between the results as the number of particles is increased.

Table 5 presents the results of these simulations using $\lambda_q = 1/8$ and $\mathcal{X} = 1/4$

# **Particles**	% **Ave.** N_{eff} Seed : 1| 8 | 16	**Ave.** $\left\|E\left[\lambda\left(t\right)\left\|y_{1:l_{n\Delta T}}\right.\right]-\lambda\left(t\right)\right\|$ Seed : 1| 8 | 16	**Ave.**
50	37.6| 37.3| 35.7	0.7819| 0.7555| 0.7371	0.7582
100	34.6| 35.2| 36.8	0.7055| 0.5674| 0.7362	0.6697
250	35.7| 35.0| 34.7	0.6012| 0.6313| 0.**6**191	0.**6**172
500	35.1| 34.9| 33.0	0.**5**779| 0.6860| 0.**6**267	0.**6**302
750	34.9| 34.4| 32.4	0.**5**413| 0.6586| 0.**6**217	0.**6**072
1000	33.9| 33.9| 32.7	0.**5**884| 0.7040| 0.**6**749	0.**6**458
2000	32.8| 32.5| 33.6	0.**5**389| 0.7302| 0.**6**146	0.**6**279
3000	32.5| 32.9| 32.0	0.**5**853| 0.**6**592| 0.**6**608	0.**6**351
5000	33.0| 32.3| 31.3	0.**5**982| 0.**6**841| 0.**6**720	0.**6**514
7500	33.2| 30.1| 32.0	0.**5**785| 0.**6**346| 0.**6**468	0.**6**200
10000	33.5| 31.1| 30.9	0.**5**678| 0.**6**829| 0.**6**767	0.**6**425

Table 5: Estimation of an exponential inhomogeneous Poisson Process Rate function.

The important point to make about these results and those to follow is that as the number of particles is increased it can be seen that the average error is beginning to stabilise and is definitely reducing as the number of particles is increased.

Table 6 presents results using $\lambda_q = 1/4$ and $\mathcal{X} = 1/4$ and Table 7 presents results using $\lambda_q = 1/2$ and $\mathcal{X} = 1/4$.

# **Particles**	% **Ave.** N_{eff}	**Ave.** $\left\|E\left[\lambda\left(t\right)\left\|y_{1:l_{n\Delta T}}\right.\right]-\lambda\left(t\right)\right\|$
100	35.5	0.7529
500	35.4	0.6194
1000	33.2	0.5772
5000	33.5	0.5318

Table 6: Estimation of an exponential inhomogeneous Poisson Process Rate function.

# Particles	% Ave. N_{eff}	Ave. $\left\lvert E\left[\lambda\left(t\right)\left\lvert y_{1:l_{n\Delta T}}\right.\right]-\lambda\left(t\right)\right\rvert$
100	32.0	0.6048
500	32.7	0.6065
1000	32.0	0.5990
5000	31.9	0.5341

Table 7: Estimation of an exponential inhomogeneous Poisson Process Rate function.

These results demonstrate several important points. The first point is that they suggest that the algorithm is robust to the parameters which are set by the user, since the range of parameters used produced similar results and behaviour throughout the simulations. Ideally one would like to use hyperpriors for the parameters which are currently being set by the user, however it is not obvious how to do so in this sequential setting. This is being investigated as future work. The results also demonstrate that as the number of particles is increased the average absolute error between the MMSE estimator and the true rate function is decreasing. This demonstrates that the estimates obtained by the TDSMC estimate of the inhomogeneous Poisson Process rate is improving as the number of particles increases. This behaviour is what is expected from SMC algorithms and suggests that the particle filter density estimate of the posterior is converging to the true posterior.

The next section presents another simulation carried out for a different rate function. The same conclusions are drawn for the results of this simulation as were made for the simulation above. Note in this case the inhomogeneous rate function is more difficult, yet the algorithm still performs well.

Example 2: Estimation of the rate of a Sinusoidal Inhomogeneous Poisson Process rate

The inhomogeneous Poisson Process rate function used in this simulation was a sinusoidal rate function with the form

$$\lambda(t) = 2 + 4(1 + cos(\frac{\pi}{50}t))$$

This is a difficult problem especially in the minima of the sinusoidal function as there are several integer intervals in which there are very few observations. This problem is however still successfully tackled by the algorithm as will be demonstrated. Again the observation data is generated using the same method detailed in Example 1 in this section. This time a realisation of the MMSE estimate versus the true rate function will be provided to demonstrate graphically that the algorithm is providing reasonable estimates of the rate function at each time instant. The plot in Figure 7 presents the MMSE estimate $E[\lambda(t)|y_{1:t}]$ versus the true sinusoidal rate function when only $N = 100$ particles were used and the user set parameters were set to $\Delta T = 1$, $\mu = 50$, $\upsilon = 5$, $\lambda_q = 1/4$ and $\mathcal{X} = 1/2$. This plot was provided to demonstrate that even with a very small number of particles and no smoothing applied, the estimated inhomogeneous Poisson Process rate obtained is still reasonably sensible.

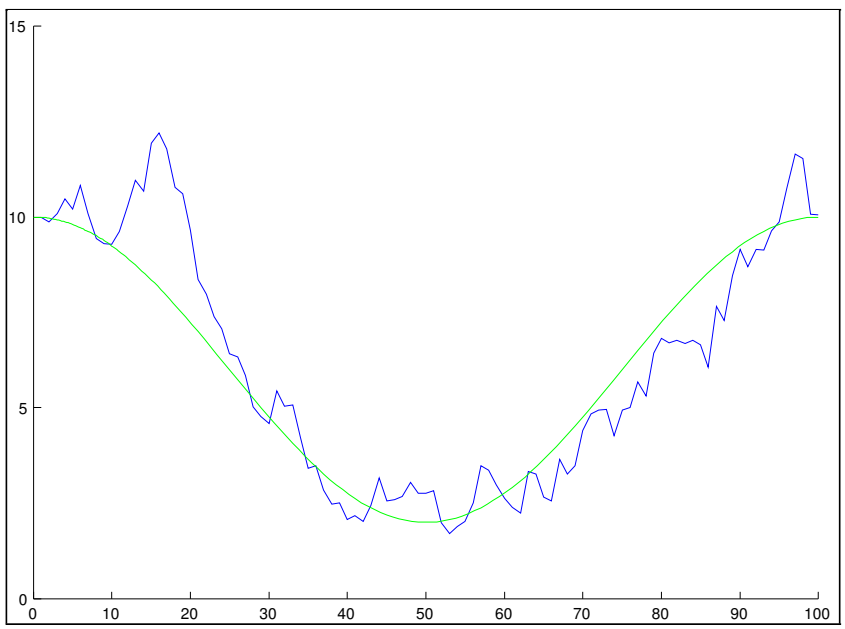

Figure 7: True Rate Function vs. MMSE Simple Function Approximation (100 particles)

Tables 8,9 and 10 present results for the sinusoidal rate function simulations, again for an increasing number of particles and a range of different user set parameter values. Table 8 is for parameter values $\lambda_q = 1/2$ and $\mathcal{X} = 1/2$, while Table 9 had parameter values $\lambda_q = 1/2$ and $\mathcal{X} = 1/4$ and finally Table 10 used $\lambda_q = 1$ and $\mathcal{X} = 1/4$.

# Particles	**% Ave.** N_{eff}	**Ave.** $\left\lvert E\left[\lambda\left(t\right)\left\lvert y_{1:l_{n\Delta T}}\right.\right] - \lambda\left(t\right)\right\rvert$
50	37.4	1.0152
100	36.1	0.9703
250	37.0	0.8863
500	35.6	0.9302
750	35.6	0.8560
1000	34.8	0.8647
2000	35.4	0.8739
3000	35.2	0.8403
5000	34.8	0.8562
7500	34.0	0.8485

Table 8: Estimation of a sinusoidal inhomogeneous Poisson Process Rate function.

# Particles	**% Ave.** N_{eff}	**Ave.** $\left\lvert E\left[\lambda\left(t\right)\left\lvert y_{1:l_{n\Delta T}}\right.\right] - \lambda\left(t\right)\right\rvert$
100	30.5	1.1021
500	29.1	0.9130
1000	29.0	0.9289
5000	29.1	0.9253

Table 9: Estimation of a sinusoidal inhomogeneous Poisson Process Rate function.

# Particles	**% Ave.** N_{eff}	**Ave.** $\left\lvert E\left[\lambda\left(t\right)\left\lvert y_{1:l_{n\Delta T}}\right.\right] - \lambda\left(t\right)\right\rvert$
100	24.3	0.9332
500	23.5	0.9317
1000	23.5	0.9423
5000	24.0	0.8786

Table 10: Estimation of a sinusoidal inhomogeneous Poisson Process Rate function.

The previous tables of results again demonstrated that the algorithm is performing as desired for a range of different user set parameters. Again the performance is seen to improve as the number of particles is increased.

Example 3: Estimation of the rate of an Inhomogeneous Poisson Process rate for Coal Mine Disasters between 1851 and 1962

The final example will involve an analysis of a real data from the Coal mine data set, which represents the coal mine disasters between 1851 and 1962 in the UK. The first plot below presents the coal mining disaster data set to be analysed, presented as a cumulative counting process.

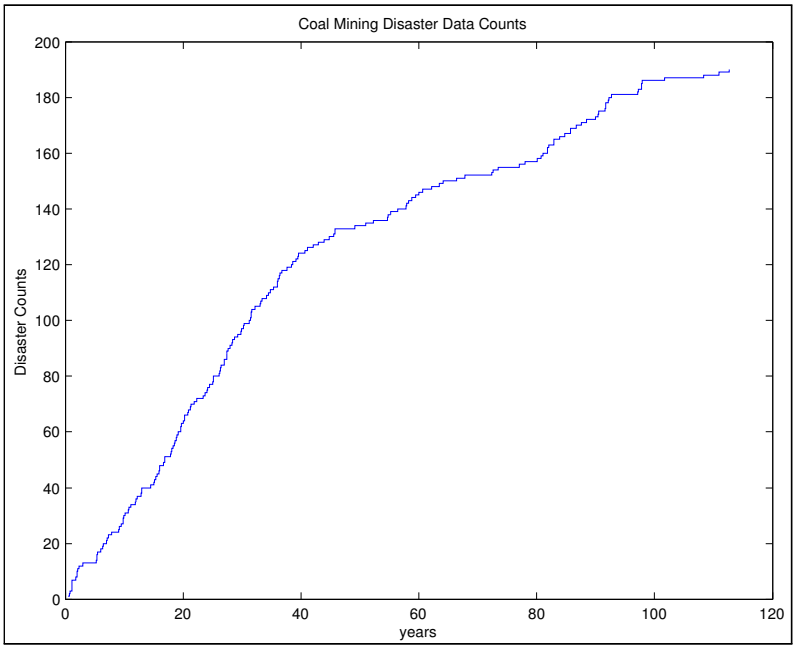

Figure 8: Coal Mining Disasters, 1851 - 1962

The version of the data set in [52] was used. However, the analysis carried out was for disasters renormalised on a year scale. A RJMCMC algorithm was also implemented, using the same statistical model as the TDSMC, to sample from the posterior distribution given the whole data set. The moves utilised in the RJMCMC algorithm were designed

to be similar to those used in [52]. The user specified parameters were $\Delta T = 1$ year, $\lambda_q = 1/4$, $\mu = 9/2$, $\upsilon = 3/2$, $D = 4$, $s = 4$, $\delta = 0.2$ and $\chi = 0.1$. The number of particles used for the SMC simulation was $N = 25000$ while the RJMCMC algorithm used 220000 samples with the first 20000 samples discarded for the "burn in" stage.

Figure 9 displays the smoothed estimate of the inhomogeneous Poisson intensity obtained using the TDSMC algorithm versus the estimated using RJMCMC. The smoothed TDSMC estimate presented is $E\left[\lambda(t)\,|y_{1:l_{n\Delta T+14\Delta T}}\right]$, hence it is different from RJMCMC which uses the whole data set. For both the TDSMC algorithm and the RJMCMC algorithm the estimated $\pm 3\sigma$ error lines are plotted. In the simulations the E_{ff} never went below $0.3N$.

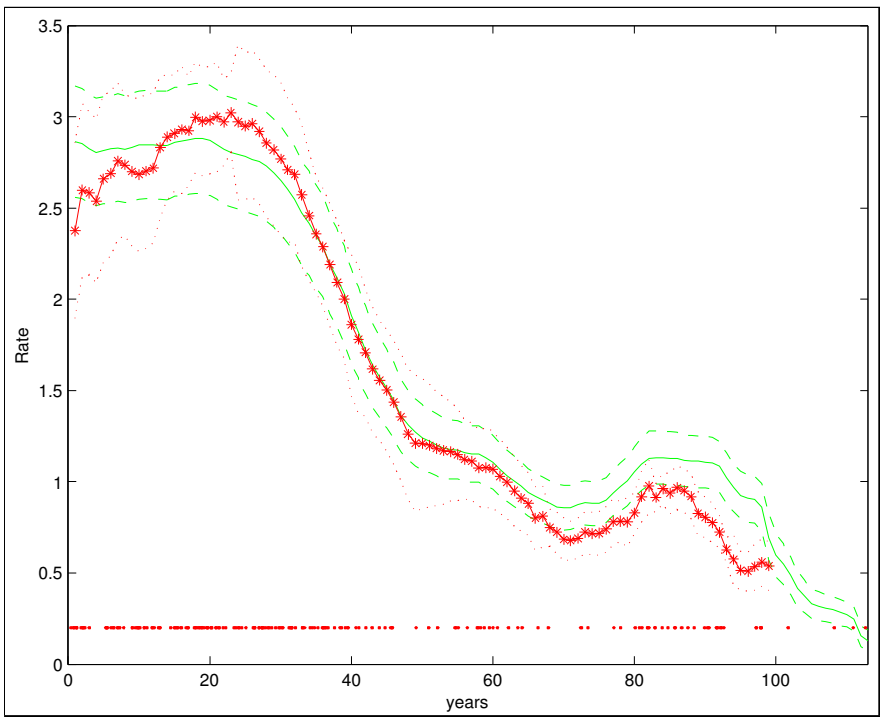

Figure 9: Bottom: Coal mining disaster data, 1851-1962: occurrences of disasters, Solid line: RJMCMC estimate of the intensity, Dashed lines: RJMCMC estimate $\pm$ 3 standard deviation, Star: TDSMC estimate of the intensity, Dotted lines: TDSMC estimate $\pm$ 3 standard deviation.

5.1.5 Summary

This example has presented a TDSMC algorithm in which the observations were analysed in a sequential setting. This constitutes what is effectively estimation of the rate of the inhomogeneous Poisson Process "on-line". The model used for this analysis and the details of the TDSMC methodology employed were presented earlier. The reason this data set was analysed is that it allows a direct comparison between Green's RJMCMC algorithm presented in [52], in which this data is analysed in a batch scenario versus the on-line analysis using TDSMC. It is important to mention that the model implemented for Green's RJMCMC methodology was changed so that it had the same prior structure as was used for the TDSMC methodology. This will facilitate direct comparison. Additionally all the move types utilised in the RJMCMC simulations were the same as those presented by Green. The main differences between the priors used for the TDSMC algorithm and Green's selected priors in [52], is that Green assumes independence between heights of segments and also probabilistically spaces the times between starting points of segments, making it more likely to have longer segments. The author argues that this is logical in a batch scenario in which one may have an idea of the number of segments that can be used to model the rate function as a result of visual inspection of the dates of disasters prior to analysis. Green makes the point that this was used to ensure that the simplest model is used to represent the rate function and that this provides an uninformative prior.

However in a true sequential setting in which the data is being presented to the algorithm as an "on-line" analysis one does not have this foresight. Hence the author feels it is justified to use a prior structure in which a height is sampled from a distribution in which the previous height is used as the mean. This has the advantage of building into the model a notion of continuity of the rate function. Both prior structures may be used for the TDSMC algorithm, however the author prefers the prior presented for the reasons outlined. The results presented therefore used the prior structure presented by the author for both the TDSMC algorithm and the RJMCMC algorithm.

5.2 General Linear Model Basis Function Regression

This section uses the same statistical model as presented in Chapter 4, in the section on radial basis function regression. However, the big difference between this section and the work presented in Chapter 4 and in [90] is that this analysis presented next is a truly sequential analysis which has a temporal ordering associated with the parameters being estimated. It should be mentioned that the next section to be presented was developed by the author independently of work found in [90] which was at the stage of development of this section, unpublished. This section provides a novel truly sequential approach which differs significantly both algorithmically and methodologically from the work presented in chapter 4.

So to summarise the situation, Chapter 4 demonstrated how significant improvement to the results obtained in [90] could be obtained by using the optimal TDSMC version of the auxiliary kernel L_t^{opt} which was developed in this thesis. This example was sequential but was not "truly sequential" in the sense that it analysed batch data in a non-iterative fashion, hence saving on computations. However in this chapter it will be demonstrated that a temporally "truly sequential" analysis can be achieved using a sliding window of observations. This leads to even greater computational savings as will be discussed. This was achieved by using the ideas developed in Chapter 4, which relate to the approximation of the optimal L_t^{opt} kernel. Hence this section demonstrates a different means of performing sequential basis function regression for the General Linear Model (GLM).

As mentioned instead of looking at all the data at time t given by $y_{1:t}$, in this formulation one considers a sliding window of data $y_{t-\Delta:t}$ which produces computational savings, this point will be elaborated upon next. Another difference between the formulation used in both Chapter 4 and [90] and the framework to be presented next relates to the types of transition kernels used. In this method a set of very general generic moves have been developed which work for a range of different models. To avoid unnecessary repetition of what has been presented previously, the full details of the statistical model will not be repeated in the sections which overlap, unless it aids the clarity of the discussion. For

the sake of notational differences, the basic model will be presented here as shown in equation (5.4).

The application of TDSMC which will be considered for this section involves the model shown in equation (5.4).

$$y_t = \sum_{j=1}^{k_t} \left[\alpha_j \exp\left(\beta_j \left(t - \tau_j\right)\right)\right] I\left(t \geqslant \tau_j\right) + w_t \qquad t \in R,\ w_t \sim N\left(0, \sigma_w\right) \tag{5.4}$$

In such a problem one is interested in sequentially inferring model order k_t, amplitudes $\alpha_{1:k,t}$, dilation factors $\beta_{1:k,t}$ and translations $\tau_{1:k,t}$ given the set of noisy observations $y_{1:t}$ which are arriving sequentially in time. It is now clear that this problem is of the form which is relevant for TDSMC methodology because the space on which the relevant posterior distribution will be defined has the product space form $\Theta_t = \cup_{k=0}^{k_{\max}} \{k_t\} \times \Theta_{k,t}$.

5.2.1 Rao-Blackwellised TDSMC: GLM

In the TDSMC formulation one is interested in sequentially obtaining a weighted particle estimate of the density $p\left(k_t, \alpha_{1:k,t}, \beta_{1:k,t}, \tau_{1:k,t} | y_{1:t}\right)$, where the set of particles at time t take the form $\left\{k_t, \alpha_{1:k,t}, \beta_{1:k,t}, \tau_{1:k,t}\right\}_t^{i=1:N}$ and the weights for the particles are obtained utilising the weighting procedure outlined previously in the TDSMC methodology, found in Chapter 4.

The statistical model used for this example is presented in Chapter 4, except for a few model specific differences. In this example the regression matrix $D\left(k_t, \theta_{1:k,t}\right)$ is shown below

$$D\left(k_t, \theta_{1:k,t}\right) = \begin{bmatrix} \exp\left[-\beta_1\left(1-\tau_1\right)\right] \mathbb{I}\left(t \geqslant \tau_1\right) & . & . & \exp\left[-\beta_k\left(1-\tau_k\right)\right] \mathbb{I}\left(t \geqslant \tau_k\right) \\ . & . & & . \\ . & & . & . \\ \exp\left[-\beta_1\left(t-\tau_1\right)\right] \mathbb{I}\left(t \geqslant \tau_1\right) & . & . & \exp\left[-\beta_k\left(t-\tau_k\right)\right] \mathbb{I}\left(t \geqslant \tau_k\right) \end{bmatrix}.$$

The prior model was also assumed to factorise as shown in equation (5.5).

$$\begin{aligned} & p\left(k_t, \alpha_{1:k,t}, \beta_{1:k,t}, \tau_{1:k,t}, \sigma_w\right) \\ = \; & p\left(\alpha_{1:k,t}|k_t, \beta_{1:k,t}, \tau_{1:k,t}, \sigma_w\right) p\left(\beta_{1:k,t}|k_t\right) p(\tau_{1:k,t}|k_t) p\left(k_t\right) p\left(\sigma_w\right) \end{aligned} \tag{5.5}$$

Where the following distributions were used for the relevant elements of the prior factors above.

- $\sigma_w^2 \sim \mathcal{IG}\left(\frac{\nu_0}{2}, \frac{\gamma_0}{2}\right)$ where $\nu_0 = \gamma_0 = 0.01$. Note, in the limit $\nu_0 \to 0, \gamma_0 \to 0$ one obtains a distribution for the noise variance which is Jeffreys' uninformative prior for scale parameters.

- $k_t \sim \mathcal{P}\left(\lambda_q t\right)$ where this is a truncated Poisson distribution with $k_t \leqslant (\lambda_q t + n)$ and it is assumed λ_q is known.

- $\tau_{1:k,t}|k_t$ now conditional on the model order at time t the translations $\tau_{1:k,t}$ are distributed as the uniform order statistics on the interval $[0, t]$

- It is assumed that the dilations $\beta_{1:k,t}|k_t$ are independent where each dilation $\beta_j|k_t \sim \mathcal{U}\left[a, b\right]$ where the user defined values are $a = 0.2$, $b = 0.7$;

- $\alpha_{1:k,t}|k_t, \beta_{1:k,t}, \tau_{1:k,t}, \sigma_w \sim \mathcal{N}\left(0, \sigma_w^2 \Sigma_{k_t}\right)$ where the mean is zero to reflect ignorance about the sign of the amplitude and $\Sigma_{k_t}^{-1} = \delta^{-2} D\left(k_t, \theta'_{1:k,t}\right)^T D\left(k_t, \theta'_{1:k,t}\right)$, [6].

Now that the prior structure has been established as before it is possible to integrate out the nuisance parameters which are the amplitudes $\alpha_{1:k,t}$ and the noise variance σ_w^2 from the full posterior. It is noted that given the prior structure established above the joint density $p\left(\alpha_{1:k,t}, \sigma_w^2|k_t, \beta_{1:k,t}, \tau_{1:k,t}\right)$ is the well known g-prior, the interested reader is referred to either Chapter 3 or [40], [34] and ([79] Appendix A) for discussion of the g-prior properties and the details of the standard integration of the posterior to remove these nuisance parameters. The posterior that is obtained after this analytic integration

takes the form presented in Chapter 3, which is reproduced here for the specifics of this model as shown in equation (5.6).

$$\begin{aligned} & p\left(k_t, \beta_{1:k,t}, \tau_{1:k,t} | y_{1:t}\right) \\ \propto \quad & \left(\gamma_0 + y_{1:t}^T P_{k_t} y_{1:t}\right)^{-\frac{(t+\nu_0)}{2}} \left[\frac{\lambda}{\left[\delta^2+1\right](b-a)}\right]^{k_t} \mathbb{I}\left(k_t, \beta_{1:k,t}, \tau_{1:k,t}\right) \end{aligned} \tag{5.6}$$

where

$$\begin{aligned} M_{k_t} &= D\left(k_t, \theta_{1:k,t}\right)^T D\left(k_t, \theta_{1:k,t}\right) + \Sigma_{k_t}^{-1} \\ P_{k_t} &= I_t - D\left(k_t, \theta_{1:k,t}\right) M_{k_t} D\left(k_t, \theta_{1:k,t}\right)^T \end{aligned}$$

It should be noted that the amplitudes integrated out of the posterior may be estimated using Least Squares or in sequential situations, Recursive Least Squares.

5.2.2 Move Details

The moves used in this application were birth, death, update and adjustment of the dilation factors. Each particle contained the following random variables $k_t, \beta_{1:k,t}, \tau_{1:k,t}$ at time t. An important point to make is that the moves were only carried out on the parameters of each particle $k_t, \beta_{k_{t-\Delta}:k,t}, \tau_{k_{t-\Delta}:k,t}$ which were located in time, within a window of $[t-\Delta, t]$, which shifts forward with the time index. The justification of this characteristic is that it is assumed that the observation obtained at time t will not contain significant information about any parameters obtained before $t-\Delta$, where Δ is the window length which may be set by the user. This allows for substantial computational savings to be made. The details of these moves are now presented below.

Update Move

In this move the parameters which make up the particle undergoing an update move obey $\left\{k_t, \beta_{1:k,t}, \tau_{1:k,t}\right\}_t^{(i)} = \left\{k_{t-1}, \beta_{1:k,(t-1)}, \tau_{1:k,(t-1)}\right\}_{t-1}^{(i)}$ and have a TDSMC generalised

importance weight which takes the following form :

$$w_t^{(i)} \propto \frac{\left(\gamma_0 + y_{t-\Delta:t}^T P_{k_{t-\Delta}:k,t} y_{t-\Delta:t}\right)^{-\frac{(t-\Delta+\nu_0)}{2}}}{\left(\gamma_0 + y_{t-1-\Delta:t-1}^T P_{k_{t-1-\Delta}:k,(t-1)} y_{t-1-\Delta:t-1}\right)^{-\frac{((t-1-\Delta)+\nu_0)}{2}}} \mathbb{I}\left(k_t, \beta_{k_{t-\Delta}:k,t}, \tau_{k_{t-\Delta}:k,t}\right)$$

Birth Move

The birth move used in this application is again based on the results presented in Chapter 4. It is written in such a way that it serves as a model for any TDSMC formulation of GLM basis regression. The birth move will be presented with respect to this example but as mentioned is very general in nature and what is more important is the fact that it is adapted to the observations and the previous particle information. The birth move is one where $k_t = k_{t-1} + 1$, and the first part of the birth step involves sampling a new dilation factor β^*. The next part of the birth step involves creating a grid of time points in the window $[t - \Delta, t]$ which will be labelled $s_{1:\Delta}$. The grid is obtained by sampling a time uniformly from each integer time segment in $[t - \Delta, t]$. Finally using the same β^* for each grid point, one obtains a multinomial distribution for the grid times $s_{1:\Delta}$, which has weights denoted by w_{birth}^j and each weight is determined using the posterior as shown in equation (5.7).

$$\begin{aligned} w_{birth}^j \quad \propto \quad & \left(\gamma_0 + y'_{t-\Delta:t}\left[P_{k_{t-\Delta}:k,t}\left(s_j\right)\right] y_{t-\Delta:t}\right)^{-\frac{(t-\Delta+\nu_0)}{2}} \\ & \times \left[\frac{\lambda}{\left[\delta^2+1\right](b-a)}\right]^{(k_t-k_{t-\Delta})} \mathbb{I}\left(k_t, \beta_{k_{t-\Delta}:k,t}, \tau_{k_{t-\Delta}:k,t}, s_j, \beta^*\right) \end{aligned} \tag{5.7}$$

These weights are then normalised and the new proposed birth translation parameter τ_j and dilation parameter β^* are sampled from this multinomial distribution. This is

summarised by the following birth kernel shown in equation (5.8).

$$
\begin{aligned}
& K_t\left(k_t, \beta_{1:k,t}, \tau_{1:k,t} | k_{t-1}, \beta_{1:k,(t-1)}, \tau_{1:k,(t-1)}\right) \qquad (5.8) \\
= \; & K_t\left(\tau_j, \beta^* | k_{t-1}, \beta_{1:k,(t-1)}, \tau_{1:k,(t-1)}\right) \delta_{\left(\left\{\tau_{1:k,(t-1)}, \beta_{1:k,(t-1)}\right\}^{(i)}\right)} \left(\tau_{1:k,(t-1)}, \beta_{1:k,(t-1)}\right) \delta_{(k_{t-1}+1)}(k) \\
= \; & \left[\sum_{j=1}^{\Delta} w_{prop}^j \delta_{\{s_j\}}(s)\right] \left[\frac{1}{(b-a)}\right] \delta_{\left(\left\{\tau_{1:k,(t-1)}, \beta_{1:k,(t-1)}\right\}^{(i)}\right)} \left(\tau_{1:k,(t-1)}, \beta_{1:k,(t-1)}\right) \delta_{(k_{t-1}+1)}(k)
\end{aligned}
$$

The particle $\left\{k_t, \beta_{1:k,(t-1)}, \beta^*, \tau_{1:k,(t-1)}, \tau_j\right\}$ then has its weight calculated as explained in Chapter 4.

Death Move

The death move has $k_t = k_{t-1} - 1$ and was constructed using the same idea as shown in the birth step. All of the parameters $\tau_{k_{t-\Delta}:k,t}, \beta_{k_{t-\Delta}:k,t}$ which are associated to the time window $[t-\Delta, t]$ are used to create a discrete distribution for which a set of parameters $\left\{\tau_j, \beta_j\right\} \in \left\{\tau_{k_{t-\Delta}:k,t}, \beta_{k_{t-\Delta}:k,t}\right\}$ will be sampled from this discrete distribution to be removed in the death step. The un-normalised multinomial weights are calculated as shown in equation (5.9).

$$
\begin{aligned}
w_{death}^j \propto & \left(\gamma_0 + y'_{t-\Delta:t}\left[P_{k_{t-\Delta}:k,t}\left(\left\{\tau_{k_{t-\Delta}:k,t}, \beta_{k_{t-\Delta}:k,t}\right\} \setminus \left\{\tau_j, \beta_j\right\}\right)\right] y_{t-\Delta:t}\right)^{-\frac{(t-\Delta+\nu_0)}{2}} \\
& \times \left[\frac{\lambda}{\left[\delta^2+1\right](b-a)}\right]^{(k_t-k_{t-\Delta})} I\left(k_t, \beta_{k_{t-\Delta}:k,t}, \tau_{k_{t-\Delta}:k,t}\right) \qquad (5.9)
\end{aligned}
$$

Hence the death move kernel may be summarised as shown in equation (5.10).

$$
\begin{aligned}
& K_t\left(k_t, \beta_{1:k,t}, \tau_{1:k,t} | k_{t-1}, \beta_{1:k,(t-1)}, \tau_{1:k,(t-1)}\right) = K_t\left(\tau_j, \beta_j | k_{t-1}, \beta_{k_{t-\Delta}:k,(t-1)}, \tau_{k_{t-\Delta}:k,(t-1)}\right) \\
& \times \delta_{\left(\left\{\tau_{k_{t-\Delta}:k,(t-1)}, \beta_{k_{t-\Delta}:k,(t-1)}\right\}^{(i)} \setminus \left\{\tau_j, \beta_j\right\}^{(i)}\right)} \left(\tau_{k_{t-\Delta}:k,t}, \beta_{k_{t-\Delta}:k,t}\right) \delta_{(k_{t-1}-1)}(k) \\
= \; & \left[\sum_{j=k_{t-\Delta}}^{k_{t-1}} w_{death}^j \mathbb{I}_{[\tau_j, \beta_j]}\right] \delta_{\left(\left\{\tau_{k_{t-\Delta}:k,(t-1)}, \beta_{k_{t-\Delta}:k,(t-1)}\right\}^{(i)} \setminus \left\{\tau_j, \beta_j\right\}^{(i)}\right)} \left(\tau_{k_{t-\Delta}:k,t}, \beta_{k_{t-\Delta}:k,t}\right) \\
& \times \delta_{(k_{t-1}-1)}(k) \qquad (5.10)
\end{aligned}
$$

The optimal L kernel now takes the form shown in equation (5.11).

$$\begin{aligned} & L_t\left(k_{t-1}, \beta_{k_{t-\Delta}:k,(t-1)}, \tau_{k_{t-\Delta}:k,(t-1)} | k_t, \beta_{k_{t-\Delta}:k,t}, \tau_{k_{t-\Delta}:k,t}\right) \\ = & \frac{\pi_{t-1}\left(k_{t-1}, \beta_{k_{t-\Delta}:k,(t-1)}, \tau_{k_{t-\Delta}:k,(t-1)}\right)}{\pi_{t-1}\left(k_{t-1}-1, \left\{\tau_{k_{t-\Delta}:k,(t-1)}, \beta_{k_{t-\Delta}:k,(t-1)}\right\} \setminus \{\tau_j, \beta_j\}\right)} \end{aligned} \tag{5.11}$$

Dilation Adjustment Move (adapted to observations and previous particle parameters)

The adjustment move is designed to randomly select a dilation factor $\{\beta_j\} \in \left\{\beta_{k_{t-\Delta}:k,t}\right\}$ from those in the window of time $[t-\Delta, t]$. As with the other moves create a grid of possible new dilations $\beta_{1:g}$ within the range $[a, b]$. Then in the same manner as shown previously create a multinomial distribution for the possible dilation factors with unnormalised weight w^j_{adjust} shown in equation (5.12).

$$\begin{aligned} w^j_{adjust} \propto & \left(\gamma_0 + y'_{t-\Delta:t}\left[P_{k_{t-\Delta}:k,t}\left(\left\{\tau_{k_{t-\Delta}:k,t}, \beta_{k_{t-\Delta}:k,t}\right\} \cup \{\beta^*_j\}\right)\right] y_{t-\Delta:t}\right)^{-\frac{(t-\Delta+\nu_0)}{2}} \\ & \times \left[\frac{\lambda}{\left[\delta^2+1\right](b-a)}\right]^{(k_t-k_{t-\Delta})} I\left(k_t, \beta_{1:k_{t-\Delta}:k,t}, \tau_{1:k_{t-\Delta}:k,t}\right) \end{aligned} \tag{5.12}$$

This adjustment kernel shown in equation (5.13) has the advantage of allowing one to easily calculate the optimal L kernel for a dilation adjustment, which will ultimately reduce the variance of the weights of the particles.

$$\begin{aligned} & K_t\left(k_t, \beta_{1:k,t}, \tau_{1:k,t} | k_{t-1}, \beta_{1:k,(t-1)}, \tau_{1:k,(t-1)}\right) = K_t\left(\beta_j | k_{t-1}, \beta_{1:k,(t-1)}, \tau_{1:k,(t-1)}\right) \\ & \times \delta_{\left(\left\{\tau_{1:k,(t-1)}, \beta_{1:k,(t-1)}\right\}^{(i)} \setminus \{\beta_j\}^{(i)}\right)}\left(\tau_{1:k,(t-1)}, \beta_{1:k\setminus j,(t-1)}\right) \delta_{(k_{t-1})}(k) \\ = & \left[\sum_{j=1}^{k_t-k_{t-\Delta}} w^j_{adjust}\delta_{(\beta_j)}(\beta)\right]\left(\frac{1}{k_t-k_{t-\Delta}}\right) \\ & \times \delta_{\left(\left\{\tau_{1:k,(t-1)}, \beta_{1:k,(t-1)}\right\}^{(i)} \setminus \{\beta_j\}^{(i)}\right)}\left(\tau_{1:k,(t-1)}, \beta_{1:k-1,(t-1)}\right) \delta_{(k_{t-1})}(k) \end{aligned} \tag{5.13}$$

Then sample a dilation factor from this multinomial distribution and calculate the

adjusted particles update weight as shown previously, where now the approximate optimal L kernel takes the form shown in equation (5.14).

$$\begin{aligned} & \widehat{L_t^{opt}}(k_t, \beta_{1:k,t}, \tau_{1:k,t}; k_{t-1}, \beta_{1:k,(t-1)}, \tau_{1:k,(t-1)}) \\ = & \frac{\pi_{t-1}\left(k_{t-1}, \beta_{1:k,(t-1)}, \tau_{1:k,(t-1)}\right) w^j_{adjust}}{\sum_{j=1}^{k_t - k_{t-\Delta}} w^j_{adjust} \pi_{t-1}\left(k_{t-1}, \beta_{1:k\backslash j,(t-1)}, \tau_{1:k,(t-1)}, \beta_j\right)} \end{aligned} \tag{5.14}$$

5.2.3 Simulation Results

In all of the simulations, that will be discussed in the next section, the results are presented for one simulation as well as summaries of 20 repeated simulations, using the same data set and algorithmic parameters. Then in appendix 5, some of the 20 multiple simulations run on the same data set can be found in more detail. The following simulations were carried out to demonstrate how effective the TDSMC algorithm can be when one uses the moves presented in the previous section.

Simulation 1

The first plot, in figure 10, shows the original data un-corrupted by noise, then the corrupted data which formed the observations. The additive Gaussian noise had variance $\sigma_w^2 = 1$ and the exponentials were generated with translation times which were simulated from a Poisson distribution with rate $\lambda_{data} = 1/20$.

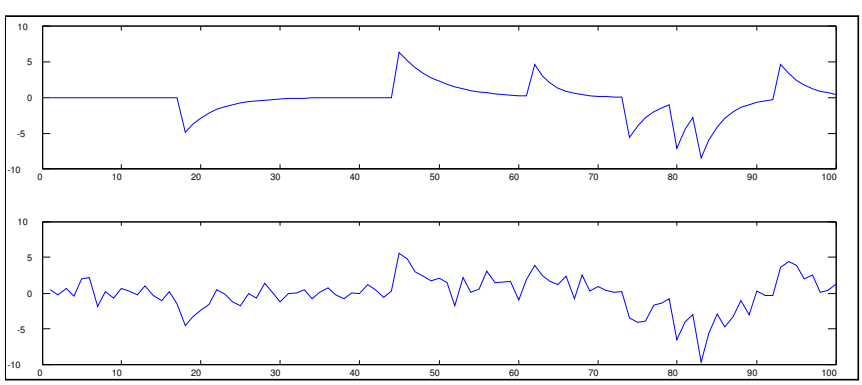

Figure 10: Top: True sequence over time; Bottom: Noisy observations over time

The results, in Table 11, used the following algorithm parameter values; $N = 5000$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/20$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$. One way of selecting the probabilities for the different types of moves is to use the fact that the problem is formulated so that birth, death and adjustment moves are only made within a window of length Δ which is shifting with the time index. Hence, the probabilities may be selected such that if there are no basis functions within the window $[t - \Delta, t]$ then only carry out birth and zero moves with probabilities set by the user, otherwise the full range of moves is available. Then, when it is possible to perform any type of move, a standard approach, as presented by Green in [52], which sets the probability of death $P_d = \min\left(1, \frac{\mathcal{P}(k-1;\lambda_q t)}{\mathcal{P}(k;\lambda_q t)}\right) \times c$, probability of birth $P_b = \min\left(1, \frac{\mathcal{P}(k+1;\lambda_q t)}{\mathcal{P}(k;\lambda_q t)}\right) \times c$, probability of time adjustment $P_{adj} = 0.1$ and the probability of doing nothing $P_z = 1 - P_{adj} - P_b - P_d$ where c is selected as large as possible under the constraint that $P_d + P_b \leqslant 0.7$, may be used.

The table demonstrates the estimate $E\left(\tau_{1:k_{(T,MAP)}}, \lambda_{1:k_{(T,MAP)}} | k_{(T,MAP)}, y_{1:T}\right)$ which are obtained by first finding the MAP model order which is obtained by finding the mode of the marginal posterior, $p\left(k_T | y_{1:T}\right)$, using the particle estimate. This mode will be labelled $k_{(T,MAP)}$. Then for all the particles of model order $k_{(T,MAP)}$, renormalise the weights and obtain the weighted average for the translations and dilations for the given subset of particles. The lower half of Table 11, presents the mean RMSE over time for a simulation, averaged over 20 simulations and the standard deviation of the mean RMSE for 20 simulations. The simulations used to calculate these quantities all used the same parameters and the same algorithmic settings. In order to obtain the mean RMSE, the amplitudes were estimated using Least Squares at time T using the MAP estimate for the model order and the MMSE estimates conditional on this MAP model order for the translation and dilation parameters. The results for the average MAP model order, k, are also presented for the 20 simulations.

True α values	-5.3227	7.0514	5.3895	-5.9342	-9.3055	-9.0064	6.3609
True τ values	17.6266	44.4703	61.5432	73.8207	79.1937	82.1407	92.1813
True β values	0.2596	0.2046	0.4240	0.3335	0.4795	0.3398	0.3256
Estimated α	-4.9942	6.2242	3.7116	-4.9292	-6.3888	-8.6567	5.5966
Estimated τ	17.9152	44.2625	61.0624	73.9212	79.6820	82.5011	92.0874
Estimated β	0.4483	0.2164	0.2069	0.2891	0.4124	0.3486	0.2128
Ave. $\overline{\text{RMSE}}$	0.6953						
Std. $\overline{\text{RMSE}}$	0.1250						
Ave. MAP k	7.3						

Table 11: True Parameter Values versus Parameter Values Estimated

The next plot in figure 11, demonstrates the reconstructed signal using the estimated parameters versus the true signal without noise and the noisy observation sequence used for the data set. It can be seen from this reconstruction that visually the estimated parameters provide a good estimate of the underlying signal given the high presence of noise.

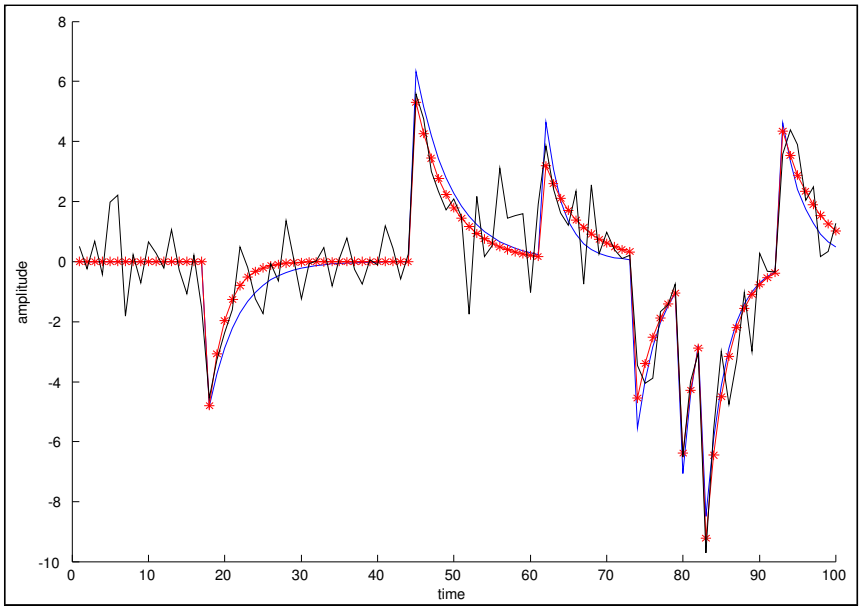

Figure 11: True noise free signal: blue; Noisy observations: black; Reconstructed estimate: red

The plot in figure 12 is a histogram of the particle estimate of the marginal posterior for the model order $p(k_T|y_{1:T})$. It can be seen that in this simulation the particle estimate has assigned a large mass to the correct model order. Obviously, this will not always be the case and these results have been presented here to demonstrate just how well the TDSMC framework can perform.

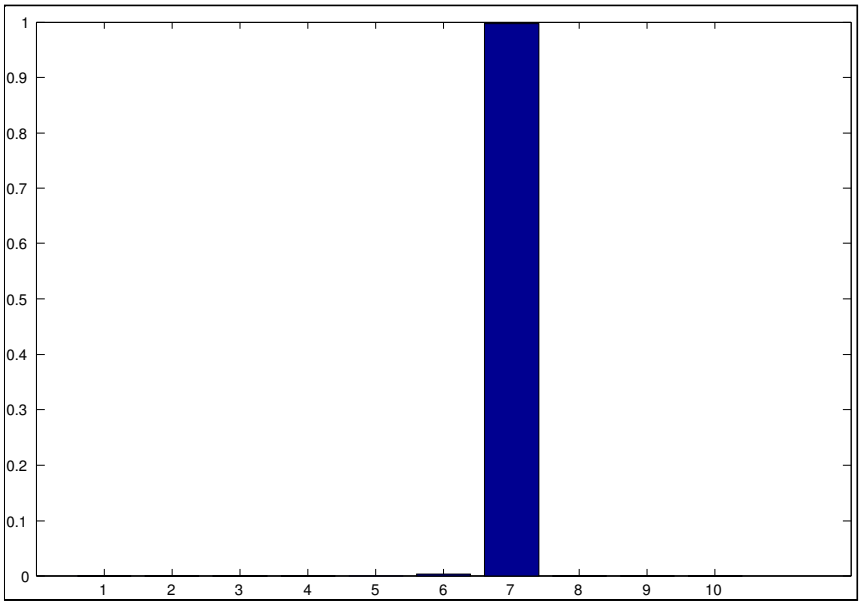

Figure 12: Histogram of estimated model order.

The next plot, in figure 13, shows a histogram of the marginal posterior of the translations and dilations, $p(\tau_{1:k_{(T,MAP)}}, \beta_{1:k_{(T,MAP)}}|k_{(T,MAP)}, y_{1:T})$, at time T given the mode estimate of the model order, $k_{(T,MAP)}$.This is obtained by resampling ten thousand times from the renormalised weights associated to the model order $k_{(T,MAP)}$, then plotting a histogram of the results.

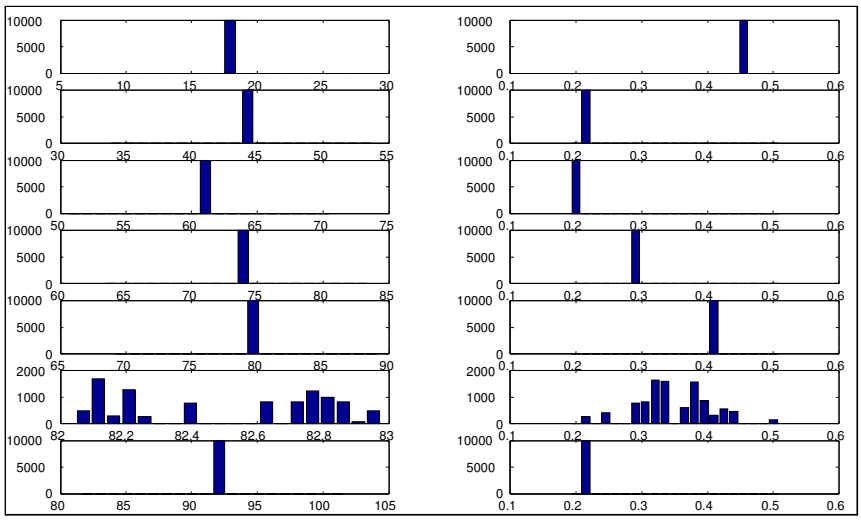

Figure 13: Histogram of estimated parameters conditioned on MAP estimate of model order. Left: translations, Right: dilations

It is clear from these results that at time T the path history of the particles has coalesced for the parameter values which were obtained early in the path history. However, the path history close to time T is clearly not degenerate since the plot above demonstrates that there are a range of values expressed by the particle estimate, conditioned on the MAP model order, for the dilations and translations. One would expect the coalescence of the path histories of the particles to depend on the length of the window Δ used in the simulation. The reason being that parameters associated with exponentials which occurred temporally before $t - \Delta$ can no longer be adjusted in the algorithm, as explained in the details of each move. One can see that there is a trade-off between making computational savings where one would like Δ to be small compared with other factors such as having enough information in the observations used to form accurate estimates and avoiding as much as possible coalescence in the path history. Hence it will be important that the TDSMC framework developed here will be fairly robust to choice of Δ, within reason.

Additionally, if one would like less coalescence in the path history, then schemes such as Fixed-Lag SMC,[38], could be attempted in this framework. This is currently being

investigated by the author. For the same data set more simulations were carried out using different number of particles, these may be found in appendix 5.

Simulation 2

The next example demonstrates the performance for a different number of particles and a different data set. The exponentials were generated with translation times which were simulated from a Poisson distribution with rate $\lambda_{data} = 1/30$. The other parameters used for this simulation were; $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/20$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$. The original data set and the noisy observations are presented in figure 14. Again the true parameters used to generate the data set are presented and then compared to the MMSE estimates of the particle estimate of $p(\tau_{1:k_{(T,MAP)}}, \beta_{1:k_{(T,MAP)}} | k_{(T,MAP)}, y_{1:T})$ when one conditions on the model order which is most probable, as reflected in the particle weights at time T.

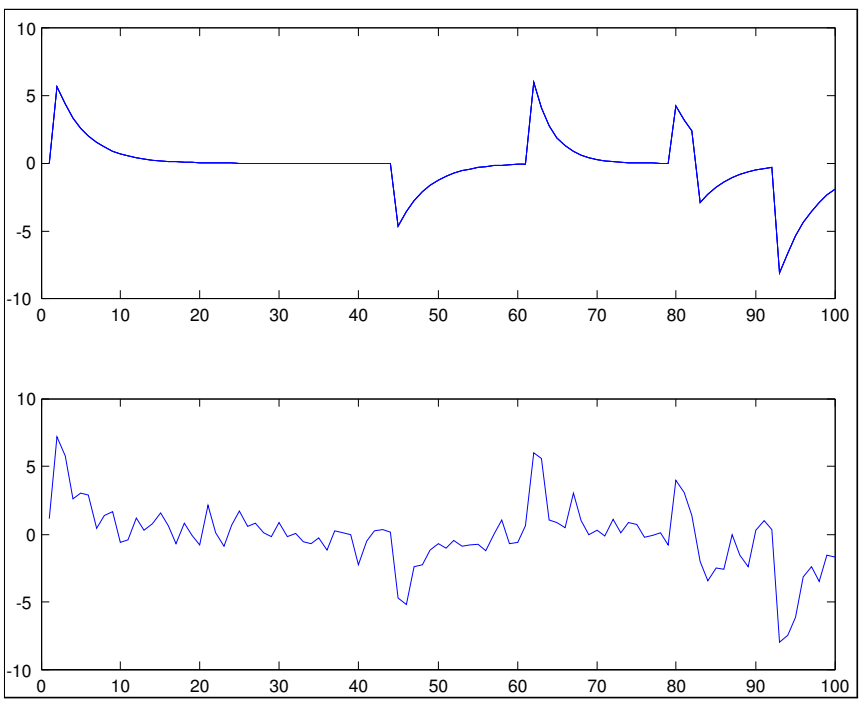

Figure 14: Top: True sequence over time; Bottom: Noisy observations over time

The comparison in table 12, again demonstrates for a randomly selected simulation, how well the TDSMC algorithm performs. It is clear that the moves developed in this chapter are working effectively, as the correct model order is being selected as the most probable model order, $k_{(T,MAP)}$. After conditioning on this model order and renormalising the particle weights associated with this model order to calculate the MMSE estimate of the parameters, one can see that, even in the presence of significant noise levels, the estimates obtained for the parameters are close to the true values. It has been found in the simulations that it is often much harder to estimate the dilations than it is to estimate the translations. This intuitively makes sense. Although this set of results is presented for just one simulation, several more simulation results using the same data set have been provided in appendix 5, which further confirm the findings presented in the simulation presented here. Again the mean RMSE averaged over 20 simulations and the standard deviation for the mean RMSE for 20 simulations is provided in table 12. This RMSE was calculated in the same manner as in simulation 1, where a Least Squares estimate of the amplitudes was obtained.

True α values	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055
True τ values	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813
True β values	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046
Estimated α	8.5916	-5.6142	6.4504	4.9900	-5.0045	-9.5095
Estimated τ	1.4840	44.1483	61.8961	79.5078	82.4975	92.5832
Estimated β	0.3274	0.2202	0.3872	0.4071	0.2744	0.2632
Ave. $\overline{\text{RMSE}}$	0.4073					
Std. $\overline{\text{RMSE}}$	0.1753					
Ave. MAP k	6.3					

Table 12: True Parameter Values versus Parameter Values Estimated

The reconstructed signal using the estimated parameter values versus the true signal and the noisy observations is presented in figure 15. It demonstrates visually how well the algorithm is performing on this given simulation.

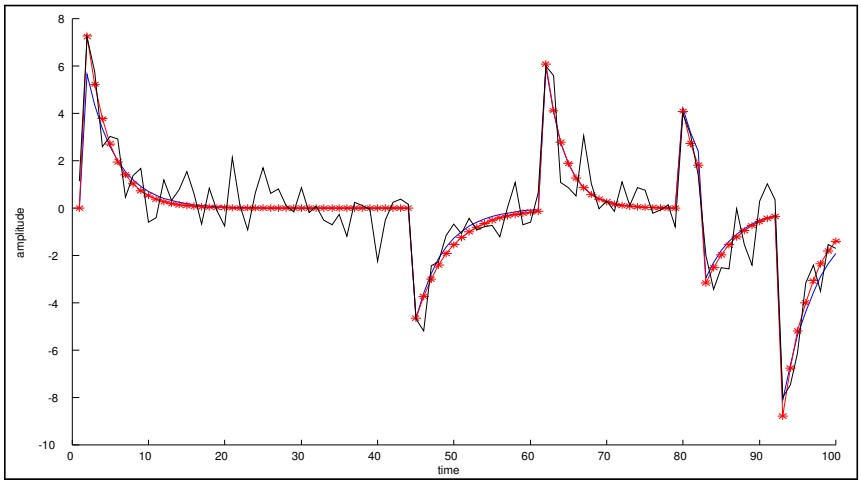

Figure 15: True noise free signal: blue; Noisy observations: black; Reconstructed estimate: red

The plot in figure 16, shows a histogram developed from the particle weights for the marginal posterior for the model order $p\left(k_T|y_{1:T}\right)$. In this example, again it is clear that the correct model is selected with very high probability, greater than 0.9. When compared to the previous example other model orders are assigned higher probabilities. However, it is clear that the majority of particles are exploring the support of the posterior in the correct model order.

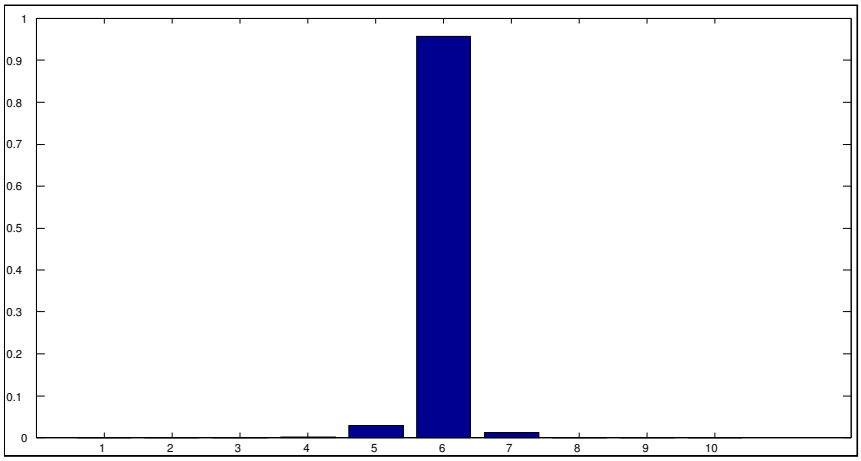

Figure 16: Histogram of estimated model order.

The histogram of the marginal posterior of the translations and dilations, $p(\tau_{1:k_{(T,MAP)}}, \beta_{1:k_{(T,MAP)}} | k_{(T,MAP)}, y_{1:T})$, at time T given the mode estimate of the model order, $k_{(T,MAP)}$, is presented below. Again this is obtained by resampling one thousand times from the renormalised weights associated to the model order $k_{(T,MAP)}$, then plotting a histogram of the results.

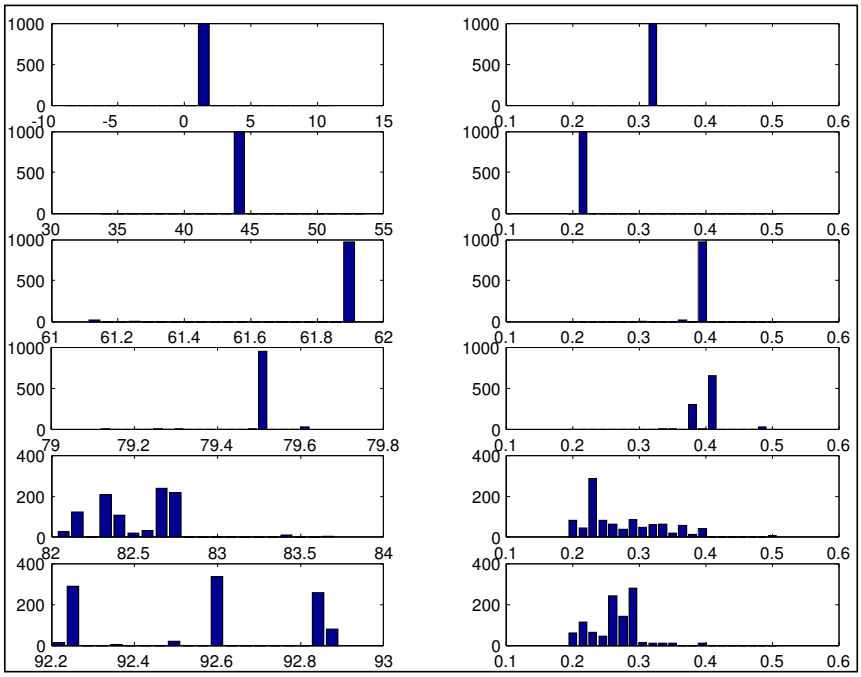

Figure 17: Histogram of estimated parameters conditioned on MAP estimate of model order. Left: translations, Right: dilations

It is clear from figure 17, that in this simulation the coalescence of the particle paths has still occurred, but not to the same extent that was presented in the previous example. The results presented here demonstrate that the correct model order is being selected with a high probability and that the particle estimate is still maintaining a diverse set of values at least within the range $[t - \Delta, t]$. The results presented in figure 17 also help to demonstrate that sample impoverishment is not a serious problem. Hence resampling has been used to combat degeneracy in the particle weights and after resampling one

can say that the sample estimate is not badly impoverished. However to ensure that sample impoverishment is not an issue one may think about adding a MCMC step or a RJMCMC step to increase the diversity of the sample. This was not carried out in these simulations as it will increase the variance of the estimates obtained, but could be carried out if required.

The translations for the exponentials, used to create the data, were generated from a Poisson distribution with rate 1/30. In practice this value is not known and must be estimated, within a reasonable range. If the TDSMC algorithm is going to be effectively applied in practice, it is important that the algorithm is fairly robust to poor choices of user set parameters. The results in the next section demonstrate that varying user set parameter λ_q, over the range $[10, 60]$, does not significantly effect model order selected. In Figure 18, a range of histograms are plotted for the particle estimate of the marginal posterior for the model order, $p\left(k_T|y_{1:T}\right)$, for different values of λ_q and the same data set and parameters used in simulation 2.

It is evident from the results presented in Figure 18 that as the mean for the model order prior, denoted, λ_q, becomes close to the true rate, $\lambda_{data} = 1/30$, used to generate the data, then the correct model order becomes the most probable model. However, even when there is a significant difference between the true rate λ_{data} and the prior mean λ_q, the correct model order is still highly probable. This demonstrates that the model order selected is robust to the user set parameter λ_q, and although these results represent just one simulation, in appendix 5 simulation 2 there are many more simulations to support this finding.

Another point to mention is that the empirical estimate of the model order marginal posterior is concentrated around the correct answer. This means that model averaging can be performed with low computational cost. This is illustrated for example by the first plot in Figure 18, where the empirical estimate for the model order is significantly concentrated on two model orders, the correct model order which is six and has probability approximately 0.45 and model order seven with probability approximately 0.52.

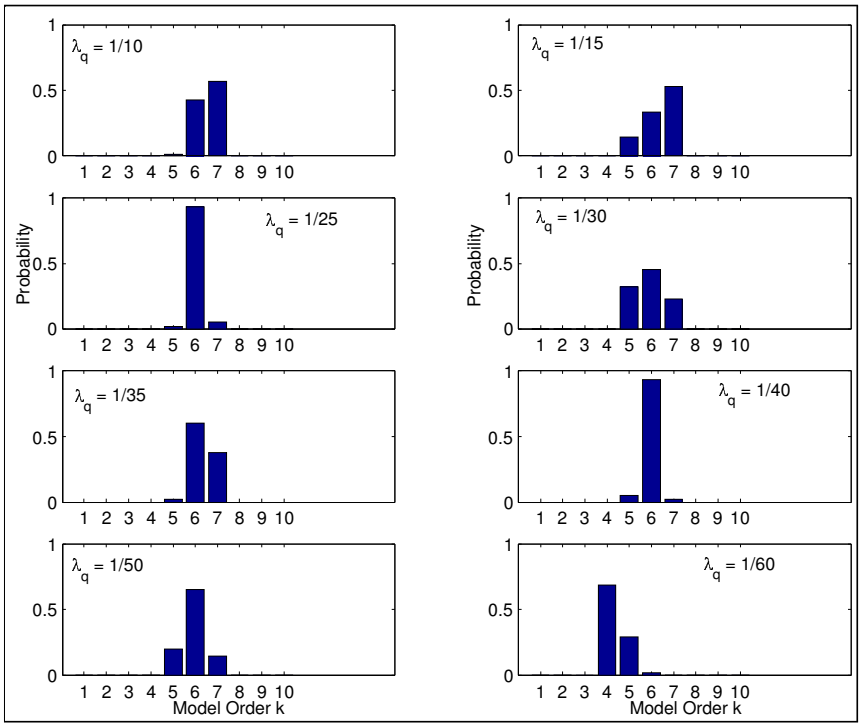

Figure 18: Histogram of estimated model order marginal posterior for different values of parameter λ_q.

The next simulations that were carried out involved determining what effect the window length, Δ, has on the robustness of the algorithm. The results of simulations in which the window length is increased are presented in Figure 19. It is clear that as the window length increases the probability of selecting the correct model order is increasing, this makes sense as one is including more and more of the data and also allowing more opportunity to adjust previous basis functions which can be changed or removed within the window $[t - \Delta, t]$. The simulations were carried out on the same data set presented in Figure 14 and the parameters used for all the simulations were $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$ and the value of the window length, Δ, used in each simulation is presented in the plots.

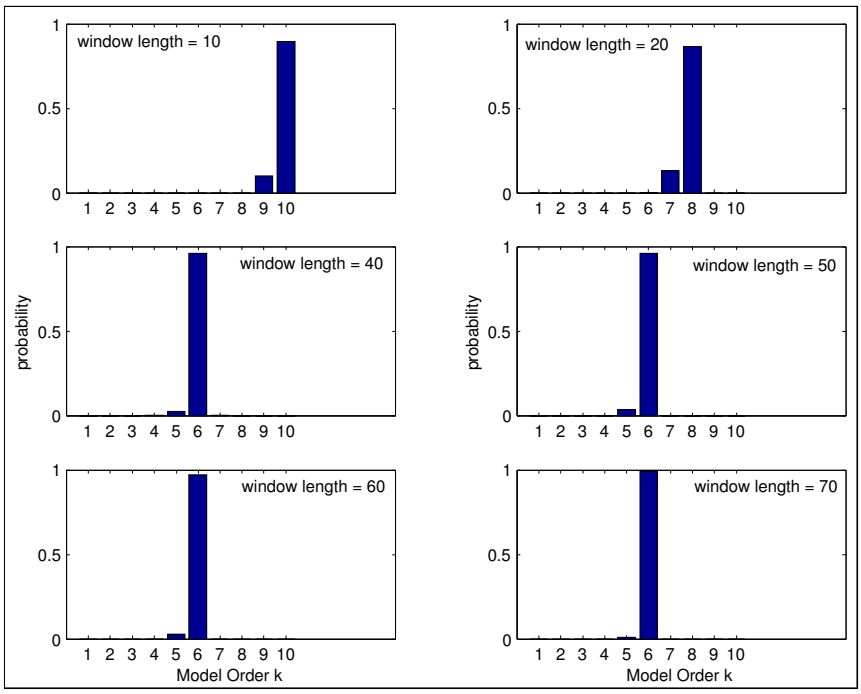

Figure 19: Histogram of estimated model order marginal posterior for different values of parameter Δ.

These results demonstrate what is already intuitive, as one increases the length of the window Δ, the performance of the algorithm improves. Clearly a window of length $\Delta = 10$ is not sufficient as the most probable model order selected is wrong. When one looks at the MMSE for this simulation, which can be found in Appendix 5, it is evident that the correct translations are being selected, however some significant noise spikes are also being selected. Then as the window length increases the algorithms performance improves rapidly, since more of the observations are being included and there is more opportunity for adjustment of the basis function parameters proposed, in light of more observations. In simulation 1 presented above, the window length was $\Delta = 20$, this worked well for this simulation. One reason for this performance, even though the window length was quite small, is due to the fact that more particles were used, $N = 5000$ compared with $N = 500$. Hence, there will always be a computational expense trade-off between the number of particles used and the length of the window.

Simulation 3

The final simulation that will be presented in this chapter involves the data presented in Figure 20. This example is presented as it demonstrates that the performance of the algorithm is not affected by periods in which no basis functions are present in intervals longer than the window length, Δ. The parameters used to perform this simulation were; $\lambda_{data} = 1/40$, $N = 1000$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/20$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

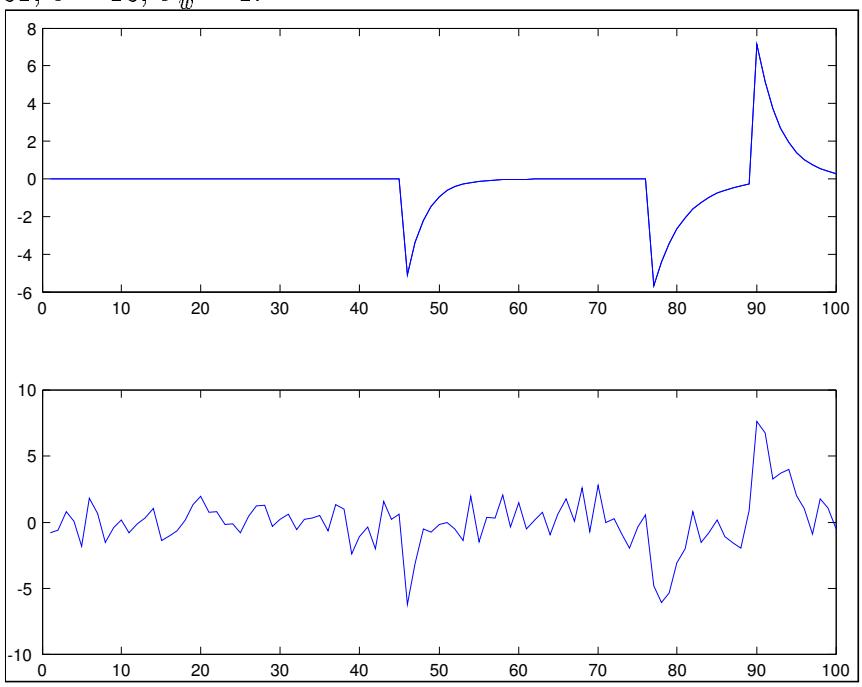

Figure 20: Top: True sequence over time; Bottom: Noisy observations over time

The results of one simulation are presented in Table 13 and more extensive results are again found in Appendix 5 under the section titled simulation 3. Again, the mean RMSE averaged over 20 simulations and the standard deviation of the mean RMSE for 20 simulations is also presented in table 13. The results in table 13 again demonstrate that the translations are being estimated well, and as was the case in many of the other simulations it is more difficult to estimate the dilations.

True α values	-5.8930	-6.9228	9.7091
True τ values	45.6468	76.2097	89.1299
True β values	0.4215	0.2529	0.3217
Estimated α	-6.4074	-8.9307	9.3580
Estimated τ	45.3459	76.1242	89.5452
Estimated β	0.4078	0.3265	0.2995
Ave. $\overline{\text{RMSE}}$	0.2573		
Std. $\overline{\text{RMSE}}$	0.0541		
Ave. MAP k	3.6		

Table 13: True Parameter Values versus Parameter Values Estimated

The reconstructed signal versus the true signal and the noisy observations is presented in figure 21. It provides a visual demonstration of how well the TDSMC algorithm can perform in the presence of high noise levels.

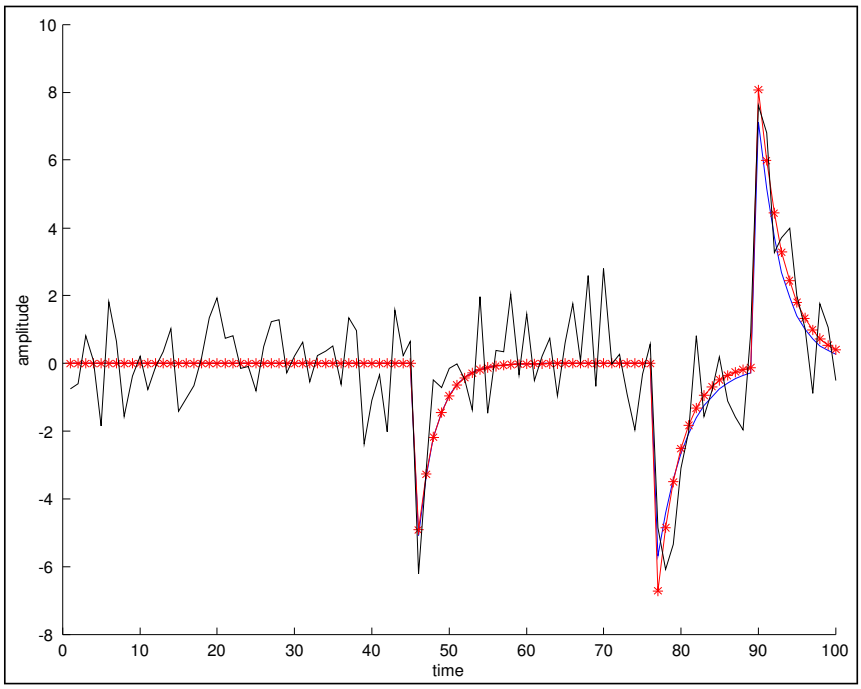

Figure 21: True noise free signal: blue, Noisy observations: black, Reconstructed estimate: red

The estimated marginal posterior histogram is not worth presenting as the correct model order was selected with a very high probability, approximately 1. It can be concluded, from this simulation and the many more found in Appendix 5 which analysed this data set, that the effect of the long period between $t \in [0, 45]$ in which there was no basis functions has not had an adverse effect on the outcome of the simulations.

5.3 Summary

This chapter introduced two detailed applications of the TDSMC algorithm to perform joint model order determination and parameter estimation using sequential data. The first example, which involved estimation of the rate function of an inhomogeneous Poisson Process, demonstrated how the TDSMC algorithm could be used as a viable alternative to RJMCMC. Simulations were carried out on some artificial simulated data, which demonstrated that different functions used to generate the inhomogeneous Poisson Process observation data could accurately be estimated using a simple function approximation in which the number of knot points, the knot point positions and intensity values were estimated using the TDSMC algorithm. This algorithm was then applied to a real data set which has been analysed several times in the literature, most notably by Green in [52]. The data set involved coal mine disasters between 1851 and 1962. The analysis of this data set involved a comparison between the TDSMC algorithm presented and the RJMCMC algorithm. The results demonstrated that the TDSMC algorithm may be considered as a viable alternative to RJMCMC, at least in situations which are sequential in nature or in situations in which the data set is massive. Improvement could be made in this analysis by including more complicated move types, however for the purpose of this thesis the aim was to present simulations which demonstrate that the TDSMC algorithm works well and to compare the TDSMC algorithm with existing algorithms in analysis of a real data set.

The second part of this chapter involved the presentation of an algorithm which allows one to carry out basis function regression for the GLM. A generic algorithm is presented which allows for the estimation of the number of basis functions and the parameters of the basis functions "sequentially" in time. The moves used in this example are presented in a general setting so that they may be applied to any application of this form with the minimum of effort. Then an example is presented in which the basis functions used are exponentials and the number and parameters are unknown. Estimation of the model order, translations and dilations is performed for three different data sets. An analysis of the effect of different user specified parameters is included to demonstrate that the algorithm is robust to the choice of these parameters.

Chapter 6

Conclusions

Initially, the literature survey points out how one may formulate a Bayesian inference problem, then in this context model selection is discussed. Different techniques which can be considered as standard statistical methods for sampling from a target distribution were presented along with guides as to when they can be used in practice. An account of fundamental Monte Carlo theory was presented with explanation of why basic sampling techniques such as inversion and rejection sampling are impractical when one has complicated target distributions. Then the methodology of Markov Chain Monte Carlo was presented, which included the Metropolis-Hastings algorithm and its many variants and finally the methodology of Importance Sampling followed by Sequential Importance Sampling and Sequential Monte Carlo were presented. The basic SMC algorithm was detailed and a guide about practical issues regarding its implementation were presented. These techniques were discussed in the context of estimation and inference in batch settings for the MCMC and in sequential settings for the SMC.

The new methodology introduced in this thesis demonstrated how the analysis of batch data could be carried out sequentially, and when such an approach would be beneficial. This involved comparison between the existing batch analysis strategies and the new sequential strategies developed in the thesis. The Reversible Jump Markov Chain Monte Carlo algorithm was also presented and it was explained how this can

be used to perform both parameter estimation and model selection. The RJMCMC algorithm is an important part in the thesis as a comparative tool, since a sequential methodology to perform model selection and parameter estimation was presented. In this context, in the same sense that Reversible Jump Markov Chain Monte Carlo can be considered as an extension of Markov Chain Monte Carlo methodology to sample from target distributions defined on spaces which include both the model order and the associated model parameters, one may analogously consider the Trans-Dimensional Sequential Monte Carlo algorithm as a natural extension of Sequential Monte Carlo Samplers methodology. These ideas were discussed and detailed analysis which involved theoretical justifications were presented in order to develop efficient generic algorithms for both the Sequential Monte Carlo Samplers methodology and its Trans-Dimensional variant.

The next section motivated the basic premise of this thesis which was to present a class of methods which allow one to sample from a sequence of distributions which are defined on the same space. The sequence of distributions can be very general, as long as they can be evaluated pointwise up to a normalizing constant, which makes the techniques presented applicable to a broad range of problems. The thesis highlights some examples where this methodology could be used. The examples mentioned included, optimisation, sampling from an easy to sample distribution and moving to a target distribution of interest through a sequence of intermediate distributions, which is similar to Annealed Importance Sampling. Also, sequential Bayesian inference of a sequence of posterior distributions conditional on the data till some time t, where t is growing with each iteration was presented as a problem which could be tackled with the SMC Samplers methodology. After motivating the need for this new methodology, the construction of the framework used to create SMC Samplers methodology is presented with justification, for certain settings in the framework, coming in the form of theoretical analysis. The theoretical results presented involved obtaining the optimal selection of auxiliary kernels which would minimise the variance of the importance weights for the particle estimates.

This optimal solution was found to be difficult to evaluate in practice and hence clever approximations and alternatives were developed. In this regard several links to existing algorithms were made, in particular links to the Annealed Importance Sampling algorithm and the ISIS algorithms were presented.

An application was presented which involved Bayesian variable selection. Two simulations were presented, the first involved moving from an easy to sample distribution to the target posterior of the indicator variables, for basis functions in the model, conditioned on the data, via a sequence of intermediate distributions. This example involved comparison to existing batch techniques such as MCMC, which would usually be used to perform such a batch analysis problem. There was also a comparison performed between AIS and the SMC Samplers methodology. The SMC Samplers approach to this problem was shown to be a very effective means of sampling from such a target distribution, computationally efficiently and with low variance in the importance weights. It was shown in all simulations performed that the resampling step introduced in the SMC Samplers methodology produced a reduction in the variance, cheaply. This reduction was demonstrated to be most prominent when the difference between adjacent distributions in the sequence of distributions was large, which confirms what one would intuitively expect.

The second example demonstrated optimisation of the posterior distribution. Comparison between parallel non-interacting Simulated Annealing and a long chain Simulated Annealing simulation was presented. It was found that the SMC Samplers algorithm outperformed both of these algorithms and this was again most apparent when the difference between adjacent distributions in the sequence of distributions was significant. This basically corresponds to situations in which few steps are used in the annealing schedule. This behaviour, as before, can be attributed to the resampling steps introduced by the SMC Samplers methodology which allowed the parallel Markov chains to interact in a principled manner.

The next section presented the new framework for Trans-Dimensional Sequential Monte Carlo. The methodology presented in this section allowed for the development of a

generic TDSMC algorithm which is non-iterative, and based on a generalisation of importance sampling to spaces of variable dimension. The ideas behind the TDSMC algorithm were developed and several move types described. A straightforward asymptotic analysis of the variance of the Importance weights was presented which was a direct application of the results presented for SMC Samplers. Two examples were analysed using TDSMC and comparisons to existing batch techniques were made. The first example involved a simulated data set and the second application involved the Boston Housing data set.

These examples were used to demonstrate sequential kernel regression on a batch data set and hence provided an example of sequential analysis being used to solve batch estimation problems. The results of the TDSMC algorithm were comparable with a variety of batch algorithms. This is remarkable considering the fact that the TDSMC algorithm is non-iterative, and only requires a single pass over the data.

The final section involved the detailed development of two applications of the TDSMC framework. The first example involved estimation of the rate of an inhomogeneous Poisson Process using a simple piecewise constant approximation in which the number of knot points, the positions of the knot points and the amplitude of the constant rate segments were the unknowns to be estimated. This problem could be framed as one in which the posterior takes support on a disjoint union of subspaces and it is required to estimate the model order and parameters. Analysis was carried out for some simulated data using different types of rate functions such as exponential and sinusoidal. Then an analysis of a real data set was performed, the data in question was for coal mine disasters from 1851-1962, this allowed for a direct comparison between the RJMCMC algorithm and the TDSMC algorithm developed. The results of this simulation demonstrated that TDSMC could be implemented successfully as an alternative to RJMCMC for batch analysis problems.

The second application was to basis function regression for the General Linear Model. This example was developed using a different approach to the example presented in Chapter 4. New generic move types were presented which allowed for approximation of

the optimal auxiliary transition kernel in the TDSMC framework. These moves may be applied in many different problems, they were then used to formulate a generic algorithm for the TDSMC method. An analysis was carried out on simulated data which involved an unknown number of exponential functions in the presence of significant additive Gaussian noise. The estimation involved inference from the target posterior on the number of exponential basis functions present and the translation, dilation and amplitudes of these basis functions conditional on the noisy observations. The results demonstrated that the technique performed well in the presence of significant noise levels and that the algorithm was robust to the choice of user set parameter values.

The future work that could be considered as a result of this thesis can be split into two categories. The first involves algorithmic improvements to the SMC Sampler and TDSMC methodologies. This might include designing smarter moves such as split and merge for the TDSMC algorithm, designing problem specific approximations to optimal transition kernels and corresponding auxiliary kernels. The second category involves SMC Samplers in the context of moving from an easy to sample distribution to a target posterior through a sequence of intermediate distributions. It is important to either determine an optimal schedule for this progression or alternatively, the development of an adaptive schedule. Such an approach would ideally allow one to adjust the target posterior at the next iteration to allow for the fact that the transition kernel used at a particular level in the schedule was not effective in placing the particles in regions of high posterior mass for the new target distribution in the sequence. Finally, for a fixed computational complexity, there is a trade-off between the number of particles and the length of runs. If the transition kernels are mixing well, the algorithm should favour shorter runs with many particles whereas if they mix slowly longer runs with less particles should be used. It could be interesting to devise quantitative measures for this behaviour to decide between longer runs with less particles or short runs with more particles.

Chapter 7

Appendix

7.1 Appendix 1

Proof of Proposition 1. The expression 3.6 comes from the delta method. Expression 3.7 is obtained via rewriting the variance expression of (Del Moral, 2004; section 9.4, pp. 300-306); see also (Chopin, 2004; theorem 1) for an alternative derivation. Using the notation of Chopin (2004). The variance is given by

$$\sigma^2_{SMC,t}(\varphi) = E_{\mu_1}\left[w_1^2 \mathcal{E}^2_{2:t}(\varphi - E_{\pi_t}(\varphi))\right] + \sum_{s=2}^{t} E_{\pi_{s-1}K_s}\left[w_s^2 \mathcal{E}^2_{s+1:t}(\varphi - E_{\pi_t}(\varphi))\right] \tag{7.1}$$

where $\mathcal{E}_{t+1:t}(\varphi) = \varphi$,

$$\mathcal{E}_{s+1:t}(\varphi) = \mathcal{E}_{s+1} \circ \cdots \circ \mathcal{E}_t(\varphi)$$

and

$$\mathcal{E}_t(\varphi)(x_{t-1}) = E_{K_t(x_{t-1},\cdot)}\left[w_t(x_{t-1}, X_t)\,\varphi(X_t)\right]$$

where $w_t(\cdot,\cdot)$ is defined in 3.4. The expression given by 7.1 is not easily interpreted as it has no obvious intuitive meaning. However after rearranging this expression as will be

shown below, an intuitive interpretation shall be obtained. The key is to notice that

$$\begin{aligned}
\mathcal{E}_t(\varphi)(x_{t-1}) &= E_{K_t(x_{t-1},\cdot)}\left[w_t(x_{t-1},X_t)\,\varphi(X_t)\right] \\
&= \int K_t(x_{t-1},x_t)\,\frac{\pi_t(x_t)\,L_{t-1}(x_t,x_{t-1})}{\pi_{t-1}(x_{t-1})\,K_t(x_{t-1},x_t)}\varphi(x_t)\,dx_t \\
&= \frac{1}{\pi_{t-1}(x_{t-1})}\int \varphi(x_t)\,\pi_t(x_t)\,L_{t-1}(x_t,x_{t-1})\,dx_t. \\
&= \frac{\widetilde{\pi}_t(x_{t-1})}{\pi_{t-1}(x_{t-1})}\int \varphi(x_t)\,\widetilde{\pi}_t(x_t|\,x_{t-1})\,dx_t
\end{aligned}$$

Similarly, one obtains

$$\begin{aligned}
& \mathcal{E}_{t-1:t}(\varphi) \\
= \;& \mathcal{E}_{t-1}(\mathcal{E}_t(\varphi))(x_{t-2}) \\
= \;& E_{K_{t-1}(x_{t-2},\cdot)}\left[w_{t-1}(x_{t-2:t-1})\,\mathcal{E}_t(\varphi)(x_{t-1})\right] \\
= \;& \frac{1}{\pi_{t-2}(x_{t-2})}\int\left(\frac{1}{\pi_{t-1}(x_{t-1})}\int \varphi(x_t)\,\pi_t(x_t)\,L_{t-1}(x_t,x_{t-1})\,dx_t\right) \\
& \qquad\qquad \times \pi_{t-1}(x_{t-1})\,L_{t-2}(x_{t-1},x_{t-2})\,dx_{t-1}. \\
= \;& \frac{1}{\pi_{t-2}(x_{t-2})}\int\left(\int \varphi(x_t)\,\widetilde{\pi}_t(x_{t-1:t}|\,x_{t-2})\,dx_{t-1:t}\right)\widetilde{\pi}_{t-2}(x_{t-2})\,dx_{t-1}. \\
= \;& \frac{\widetilde{\pi}_{t-1}(x_{t-2})}{\pi_{t-2}(x_{t-2})}\int \varphi(x_t)\,\widetilde{\pi}_t(x_t|\,x_{t-2})\,dx_t
\end{aligned}$$

and, by induction, one gets

$$\begin{aligned}
\mathcal{E}_{s+1:t}(\varphi) &= \frac{1}{\pi_s(x_s)}\int\cdots\int \varphi(x_t)\,\pi_t(x_t)\prod_{i=s}^{t-1}L_i(x_i,x_{i-1})\,dx_{s+1:t}. \qquad (7.2) \\
&= \frac{\widetilde{\pi}_t(x_s)}{\pi_s(x_s)}\int \varphi(x_t)\,\widetilde{\pi}_t(x_t|\,x_s)\,dx_t.
\end{aligned}$$

The expression of $\sigma^2_{SMC,t}(\varphi)$, given 3.7, follows now directly from 7.2 and 7.1.

7.2 Appendix 2

Proof of Proposition 2. Utilising the variance decomposition formula

$$var\left[w\left(X_{1:t}\right)\right] = E\left[var\left[w\left(X_{1:t}\right)\middle| X_t\right]\right] + var\left[E\left[w\left(X_{1:t}\right)\middle| X_t\right]\right]. \tag{7.3}$$

The second term on the right hand side of (7.3) is independent of $\widetilde{\pi}_t\left(x_{1:t-1}\middle| x_t\right)$ as

$$E\left[w\left(X_{1:t}\right)\middle| X_t\right] = \frac{\pi_t\left(X_t\right)}{\mu_t\left(X_t\right)}$$

whereas $var\left[w\left(X_{1:t}\right)\middle| X_t\right]$ is equal to zero if using (3.12). It is straightforward to check that (3.12) admits the form (3.2) for $\{L_t\}$ given by (3.13), i.e.

$$\mu_1\left(x_1\right)\prod_{s=2}^{t} K_s\left(x_{s-1}, x_s\right) = \mu_t\left(x_t\right)\prod_{s=2}^{t}\frac{\mu_{s-1}\left(x_{s-1}\right)K_s\left(x_{s-1}, x_s\right)}{\mu_s\left(x_s\right)}. \tag{7.4}$$

Note that (7.4) is simply the forward-backward formula for Markov processes.

7.3 Appendix 3

This section provides the results for the simulations carried out for the sequential kernel regression problem using the sinc data set. The results are averaged over 50 simulations for each value of N. The number of kernels presented here is the weighted average number of kernels.

Number of Particles N	RMSE	σ_{RMSE}	Number of Kernels
$\lambda = 1$			
10	0.1932	0.0904	1.9
50	0.1443	0.0771	3.5
100	0.1334	0.0762	3.8
250	0.0999	0.0435	4.4
500	0.0911	0.0395	4.3
1000	0.0880	0.0433	4.1
$\lambda = 2$			
10	0.1609	0.0759	3.4
50	0.0999	0.0581	4.7
100	0.0932	0.0601	5.0
250	0.0851	0.0563	5.1
500	0.0821	0.0456	5.0
1000	0.0821	0.0456	4.8
$\lambda = 3$			
10	0.1288	0.0673	4.1
50	0.1035	0.0554	5.6
100	0.0823	0.0360	5.7
250	0.0737	0.0293	6.0
500	0.0761	0.0309	5.6
1000	0.0691	0.0276	5.3

Number of Particles N	RMSE	σ_{RMSE}	Number of Kernels
$\lambda = 4$			
10	0.0969	0.0453	5.1
50	0.0715	0.0290	6.2
100	0.0697	0.0310	6.4
250	0.0700	0.0312	6.2
500	0.0648	0.0252	6.5
1000	0.0708	0.0302	5.6
$\lambda = 5$			
10	0.0894	0.0455	6.0
50	0.0728	0.0365	6.6
100	0.0669	0.0231	7.1
250	0.0607	0.0216	7.1
500	0.0595	0.0178	7.1
1000	0.0563	0.0121	7.0
$\lambda = 6$			
10	0.0825	0.0310	6.1
50	0.0663	0.0304	7.5
100	0.0604	0.0192	7.4
250	0.0587	0.0173	7.6
500	0.0609	0.0206	7.5
1000	0.0586	0.0157	6.9
$\lambda = 7$			
10	0.0799	0.0448	6.9
50	0.0648	0.0264	7.8
100	0.0687	0.0310	8.0
250	0.0589	0.0149	8.4
500	0.0587	0.0198	7.9
1000	0.0587	0.0197	7.4

Number of Particles N	RMSE	σ_{RMSE}	Number of Kernels
$\lambda = 8$			
10	0.0687	0.0247	7.9
50	0.0606	0.0266	8.4
100	0.0649	0.0294	8.5
250	0.0544	0.0155	9.1
500	0.0565	0.0136	8.3
1000	0.0587	0.0136	7.8

7.4 Appendix 4

This section provides the results for the simulations carried out for the sequential kernel regression problem using the Boston housing data set. The results are averaged over 10 simulations for each value of λ, with each simulation having a random partitioning of the data set into 300 training / 206 test data points and the number of particles $N = 250$ was used. The λ parameter is the mean for the truncated Poisson prior used for the model order. The number of kernels presented here is the weighted average number of kernels and the error range is one standard deviation.

λ value	Test Error	ave. Number of Kernels
5	8.3454±0.4922	3.8±0.81
10	8.0409±0.5815	5.26±1.00
15	7.9625±0.6615	8.60±2.39
20	7.7367±0.4995	13.81±1.70
25	7.6094±0.3543	16.40±2.60
30	7.6528±0.4439	19.99±2.02

7.5 Appendix 5

All of these experiments present the MMSE results for the parameters after conditioning on the model order $k_{(T,MAP)}$ as was carried out in Chapter 5. The following gives a guide as to what is contained in the tables of results.

- α - **True Amplitude**
- τ - **True translations**
- β - **True dilations**
- $\widehat{\alpha}$ - **Estimated amplitudes**
- $\widehat{\tau}$ - **Estimated translations**
- $\widehat{\beta}$ - **Estimated dilations**

7.5.1 Simulation 1: Chapter 5

Experiment 1: Parameters $\mathbf{N} = \mathbf{100}$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/20$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	-5.3227	7.0514	5.3895	-5.9342	-9.3055	-9.0064	6.3609	
τ	17.6266	44.4703	61.5432	73.8207	79.1937	82.1407	92.1813	
β	0.2596	0.2046	0.4240	0.3335	0.4795	0.3398	0.3256	
$\widehat{\tau}$	0.9565	17.1536	44.9302	51.7504	73.3591	79.4929	82.7980	92.3751
$\widehat{\beta}$	0.3077	0.2546	0.3896	0.2800	0.2499	0.3214	0.3457	0.3424

Experiment 2: Parameters $\mathbf{N = 500}$ **particles,** $a = 0.2$, $b = 0.5$, $\lambda_q = 1/20$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	-5.3227	7.0514	5.3895	-5.9342	-9.3055	-9.0064	6.3609	
τ	17.6266	44.4703	61.5432	73.8207	79.1937	82.1407	92.1813	
β	0.2596	0.2046	0.4240	0.3335	0.4795	0.3398	0.3256	
$\widehat{\tau}$	0.0795	17.6793	44.5014	55.1400	73.5754	79.6666	82.5891	92.2663
$\widehat{\beta}$	0.4004	0.3702	0.2401	0.4259	0.3277	0.3095	0.2963	0.3036

Experiment 3: Parameters $\mathbf{N = 1000}$ **particles**, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/20$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	-5.3227	7.0514	5.3895	-5.9342	-9.3055	-9.0064	6.3609	
τ	17.6266	44.4703	61.5432	73.8207	79.1937	82.1407	92.1813	
β	0.2596	0.2046	0.4240	0.3335	0.4795	0.3398	0.3256	
$\widehat{\tau}$	0.0424	17.6489	44.8620	61.8219	73.3552	79.5678	82.7316	92.5202
$\widehat{\beta}$	0.4857	0.3057	0.3113	0.3438	0.3104	0.3546	0.3367	0.2799

The next section presents some of the 20 simulations carried out for the data set presented in Chapter 5, simulation 1.

Experiment 4: Parameters $\mathbf{N = 5000}$ **particles**, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/20$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	-5.3227	7.0514	5.3895	-5.9342	-9.3055	-9.0064	6.3609	
τ	17.6266	44.4703	61.5432	73.8207	79.1937	82.1407	92.1813	
β	0.2596	0.2046	0.4240	0.3335	0.4795	0.3398	0.3256	
	Simulation	**1**						
$\widehat{\tau}$	0.3605	17.2828	44.1214	73.1710	79.5342	82.3799	92.5556	
$\widehat{\beta}$	0.3088	0.3274	0.4860	0.2660	0.3589	0.3106	0.2341	
$\widehat{\alpha}$	0.7444	-5.4727	10.2127	-5.5037	-6.2764	-8.0488	5.4133	
	Simulation	**2**						
$\widehat{\tau}$	0.5242	17.8963	44.8125	60.7365	73.4909	79.7005	82.4310	92.0023
$\widehat{\beta}$	0.4124	0.4485	0.3494	0.2779	0.2286	0.4392	0.3400	0.2569
$\widehat{\alpha}$	0.7243	-5.0380	6.6235	3.7716	-4.8515	-6.0534	-8.6771	6.4236
	Simulation	**3**						
$\widehat{\tau}$	0.0251	17.5585	44.3268	73.5493	79.8679	82.5602	92.3994	
$\widehat{\beta}$	0.3897	0.4914	0.4416	0.3238	0.3623	0.3246	0.2779	
$\widehat{\alpha}$	0.8786	-6.1195	8.8206	-5.3927	-5.8525	-7.9113	6.0374	
	Simulation	**4**						
$\widehat{\tau}$	17.1872	44.6980	60.1263	73.8326	79.3749	82.5157	92.7360	
$\widehat{\beta}$	0.4017	0.2040	0.2070	0.2525	0.3338	0.3705	0.2183	
$\widehat{\alpha}$	-6.4484	5.4898	3.6494	-4.7851	-6.3720	-8.2190	4.9060	
	Simulation	**5**						
$\widehat{\tau}$	17.7909	44.2141	60.7540	73.1880	79.5596	82.4615	92.5171	
$\widehat{\beta}$	0.4445	0.4401	0.3738	0.3048	0.3298	0.3094	0.2835	
$\widehat{\alpha}$	-5.2639	9.2486	4.0533	-5.9103	-6.2368	-7.6901	5.9640	

7.5.2 Simulation 2: Chapter 5

Changing the λ_q parameter

Experiment 5: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\boldsymbol{\lambda}_q = \mathbf{1/10}$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055	
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813	
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046	
$\widehat{\tau}$	1.2320	21.9405	45.1096	61.9936	79.4947	82.5977	92.5164
$\widehat{\beta}$	0.3800	0.4878	0.4232	0.3190	0.3891	0.2526	0.2912

Experiment 6: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\boldsymbol{\lambda}_q = \mathbf{1/15}$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055	
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813	
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046	
$\widehat{\tau}$	1.2320	21.9405	45.1096	61.9936	79.4947	82.5977	92.5164
$\widehat{\beta}$	0.3800	0.4878	0.4232	0.3190	0.3891	0.2526	0.2912

Experiment 7: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\boldsymbol{\lambda}_q = \mathbf{1/25}$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046
$\widehat{\tau}$	1.0759	20.4043	44.9558	61.9425	83.4822	92.2399
$\widehat{\beta}$	0.3529	0.4336	0.2919	0.4223	0.4336	0.2708

Experiment 8: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\boldsymbol{\lambda}_q = \mathbf{1/30}$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046
$\widehat{\tau}$	1.8234	20.9505	44.0814	61.5498	83.3535	92.4162
$\widehat{\beta}$	0.3849	0.2991	0.3468	0.3514	0.3602	0.2672

Experiment 9: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\boldsymbol{\lambda}_q = \mathbf{1/35}$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046
$\widehat{\tau}$	1.4324	21.5712	44.0802	61.4281	82.3485	92.5969
$\widehat{\beta}$	0.3097	0.3739	0.3262	0.3792	0.2780	0.2715

Experiment 10: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\boldsymbol{\lambda}_q = \mathbf{1/40}$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046
$\widehat{\tau}$	1.0479	19.2023	44.3825	61.4312	83.9596	92.5199
$\widehat{\beta}$	0.4509	0.2219	0.2178	0.2998	0.3524	0.2851

Experiment 11: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\boldsymbol{\lambda}_q = \mathbf{1/60}$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046
$\widehat{\tau}$	1.4653	44.4220	61.4833	92.6219		
$\widehat{\beta}$	0.2746	0.4908	0.3126	0.2923		

The next section presents some of the 20 simulations carried out for the data set presented in Chapter 5, simulation 2.

Experiment 12: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/30$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$..

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055	
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813	
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046	
	Simulation	**1**					
$\widehat{\tau}$	1.8848	28.6636	44.2774	61.8130	83.0805	92.3412	
$\widehat{\beta}$	0.2817	0.4534	0.2475	0.3484	0.4576	0.2848	
$\widehat{\alpha}$	7.0916	0.0737	-5.8581	6.3260	-5.9055	-10.8045	
	Simulation	**2**					
$\widehat{\tau}$	1.4160	24.4461	44.8328	61.8240	79.6521	82.3793	92.1517
$\widehat{\beta}$	0.3220	0.4117	0.2781	0.4374	0.2823	0.2646	0.2742
$\widehat{\alpha}$	8.7066	1.7513	-5.3751	6.8699	4.1443	-5.7309	-10.9084
	Simulation	**3**					
$\widehat{\tau}$	1.3374	44.1720	61.8716	79.4339	82.3907	92.2135	
$\widehat{\beta}$	0.3315	0.2697	0.2662	0.3365	0.2953	0.2431	
$\widehat{\alpha}$	9.0770	-6.3403	5.4538	4.6777	-5.7004	-10.1214	
	Simulation	**4**					
$\widehat{\tau}$	1.9976	44.4055	61.4049	79.6414	82.6427	92.6755	
$\widehat{\beta}$	0.4158	0.3009	0.3044	0.4475	0.3092	0.2454	
$\widehat{\alpha}$	7.8095	-6.3160	6.6725	4.8818	-4.9071	-9.0970	
	Simulation	**5**					
$\widehat{\tau}$	1.6730	24.6536	44.2181	61.9679	83.1435	92.5191	
$\widehat{\beta}$	0.2669	0.2910	0.2349	0.4123	0.3769	0.2914	
$\widehat{\alpha}$	7.3292	1.2474	-5.7585	6.3836	-5.0774	-10.3102	

Changing the Δ parameter

Experiment 13: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/30$, $\mathbf{\Delta = 10}$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055	
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813	
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046	
$\widehat{\tau}$	1.5028	6.0644	19.7134	26.9303	29.0260	44.8597	61.3547
$\widehat{\beta}$	0.2669	0.3192	0.2267	0.3838	0.2938	0.2637	0.4272

$\widehat{\tau}$	78.3935	82.5512	92.2035
$\widehat{\beta}$	0.2466	0.4354	0.3076

Experiment 14: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/30$, $\mathbf{\Delta = 20}$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055		
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813		
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046		
$\widehat{\tau}$	1.5249	20.3420	39.5349	44.4066	61.8383	79.2332	81.1197	92.3447
$\widehat{\beta}$	0.4474	0.3595	0.3641	0.3093	0.3848	0.2840	0.3291	0.2664

Experiment 15: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/30$, $\mathbf{\Delta = 40}$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046
$\widehat{\tau}$	1.3114	44.1248	61.5732	79.7884	82.9595	92.3883
$\widehat{\beta}$	0.3428	0.2396	0.3136	0.4292	0.3210	0.2575

Experiment 16: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/30$, $\mathbf{\Delta = 50}$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046
$\widehat{\tau}$	1.3083	44.0173	61.9223	79.5733	82.3892	92.4223
$\widehat{\beta}$	0.2278	0.2588	0.4387	0.3637	0.2929	0.2484

Experiment 17: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/30$, $\mathbf{\Delta = 60}$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046
$\widehat{\tau}$	1.3591	44.7666	61.8504	79.4904	82.5570	92.7994
$\widehat{\beta}$	0.3291	0.2355	0.3696	0.4091	0.2823	0.2750

Experiment 18: Parameters $N = 500$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/30$, $\mathbf{\Delta = 70}$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	5.8813	-5.3227	7.1514	5.2895	-5.9342	-9.3055
τ	1.8504	44.4703	61.5432	79.1937	82.1407	92.1813
β	0.2608	0.2596	0.3811	0.2817	0.2596	0.2046
$\widehat{\tau}$	1.2119	44.5536	61.7973	79.1753	82.6995	92.3016
$\widehat{\beta}$	0.2872	0.3155	0.4054	0.4416	0.3423	0.2628

7.5.3 Simulation 3: Chapter 5

Experiment 19: Parameters $N = 1000$ particles, $a = 0.2$, $b = 0.5$, $\lambda_q = 1/20$, $\Delta = 30$, $g = 20$, $\nu_0 = \gamma_0 = 0.01$, $\delta = 10$, $\sigma_w^2 = 1$.

α	-5.8930	-6.9228	9.7091		
τ	45.6468	76.2097	89.1299		
β	0.4215	0.2529	0.3217		
	Simulation	**1**			
$\widehat{\tau}$	45.7088	76.3257	89.1208		
$\widehat{\beta}$	0.4011	0.2765	0.4331		
$\widehat{\alpha}$	-5.4700	-7.6559	13.3961		
	Simulation	**2**			
$\widehat{\tau}$	45.0873	76.6406	89.3285		
$\widehat{\beta}$	0.3875	0.2926	0.2837		
$\widehat{\alpha}$	-6.8140	-7.2062	9.7516		
	Simulation	**3**			
$\widehat{\tau}$	45.7088	76.3257	89.1208		
$\widehat{\beta}$	0.4011	0.2765	0.4331		
$\widehat{\alpha}$	-5.4700	-7.6559	13.3961		
	Simulation	**4**			
$\widehat{\tau}$	0.2753	18.2579	45.6903	76.5512	89.9041
$\widehat{\beta}$	0.3391	0.2851	0.4525	0.2116	0.3002
$\widehat{\alpha}$	-0.6215	1.9638	-5.9269	-6.2944	8.7384
	Simulation	**5**			
$\widehat{\tau}$	45.9387	76.5382	89.4400		
$\widehat{\beta}$	0.4726	0.3396	0.3085		
$\widehat{\alpha}$	-5.4041	-7.9351	9.7879		

Bibliography

[1] B.D.O. Anderson and J.B. Moore, Optimal Filtering, Prentice-Hall, October 1978.

[2] C. Andrieu, N. De Freitas, A. Doucet and M. Jordan, "An Introduction to MCMC for Machine Learning", Machine Learning, 50, 5-43, 2003.

[3] C. Andrieu, M. Davy and A. Doucet, "Efficient Particle Filtering for Jump Markov Systems", Proc. IEEE ICASSP, 2002.

[4] C. Andrieu, M. Davy and A. Doucet, "Improved Auxiliary Particle Filtering: Application to Time-Varying Spectral Analysis", IEEE SSP 2001, Singapore, August 2001.

[5] C. Andrieu and N. de Freitas and A. Doucet, "Robust Full Bayesian Learning for Radial Basis Networks", Neural Computation, vol. 13, 10, 2359-2407, 2001.

[6] C. Andrieu, E. Barat and A. Doucet, "Bayesian Deconvolution of Noisy Filtered Point Processes", IEEE Trans. on Signal Processing, vol. 49, 1, January 2001.

[7] C. Andrieu, J. F. G. de Freitas and A. Doucet, "Reversible Jump MCMC Simulated Annealing for Neural Networks", Uncertainty in Articial Intelligence, Morgan Kaufmann, 11-18, 2000.

[8] C. Andrieu and A. Doucet, "Joint Bayesian Model Selection and Estimation of Noisy Sinusoids via Reversible Jump MCMC", IEEE Trans. on Signal Processing, October 1999.

[9] E. Arjas and J. Heikkinen, "An Algorithm for Nonparametric Bayesian Estimation of a Poisson Intensity", Computational Statistics, No.12, 385-402, 1997.

[10] A. Doucet, N. J.Gordon and V. Krishnamurthy, "Particle Filters for State Estimation of Jump Markov Linear Systems", IEE Transactions on Signal Processing, Vol. 49, No. 3, March 2001.

[11] S. Aralampalam, S. Maskell, N. J. Gordon and T. Clapp, "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking", IEEE Trans on Signal Processing, Vol. 50(2), 174-188, 2002.

[12] T.Bayes, "An Essay Towards Solving a Problem with the Doctrine of Chances", Biometrika, 45, 293-315, 1958.

[13] O. Barndorff-Nielsen, D. Cox and C. Kluppelberg, Complex Stochastic Systems, Chapman & Hall/CRC, 1999.

[14] N. Bergman, "Recursive Bayesian Estimation : Navigation and Tracking Applications", PhD. Dissertation, Linkoping University, Linkoping, Sweden, 1999.

[15] Jose M. Bernardo and Adrian F. M. Smith, Bayesian Theory, Wiley Series in Probability and Statistics, Wiley, 1994.

[16] C. M. Bishop and M. E. Tipping, "Variational Relevance Vector Machines", Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence", 46-53, editor: C. Boutilier and M. Goldszmidt, Morgan Kaufmann, 2000.

[17] C. M. Bishop and M. E. Tipping, "Bayesian Regression and Classification.", In J. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle (Eds.), Advances in Learning Theory: Methods, Models and Applications, Volume 190, 267–285. IOS Press, NATO Science Series III: Computer and Systems Sciences, 2003.

[18] K. Borovkov, Elements of Stochastic Modelling, World Scientific, 2003.

[19] Box and Tiao, Bayesian Inference in Statistical Analysis,Wiley Classics Library, 1992.

[20] S.P. Brooks, P. Giudici and G.O. Roberts, "Efficient Construction of Reversible Jump MCMC Proposal Distributions (with discussion).", Journal of the Royal Statistical Society, Series B. 65. 3-55, 2003.

[21] S. P. Brooks, N. Friel and R. King, "Classical Model Selection via Simulated Annealing", Journal of Royal Statistical Society, Series B, 65, Part 2, 503-520, 2003.

[22] S. P. Brooks and P. Giudici, "Convergence Assessment for Reversible Jump MCMC Simulations", Bayesian Statistics 6, Oxford University Press, 1998.

[23] O. Cappé, A. Guillin, J.M. Marin and C.P. Robert, "Population Monte Carlo". J. Comp. Graph. Stat., to appear.

[24] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved Particle Filter for Nonlinear Problems.", IEE Proc. Radar Sonar Navigation, 146(1):27, 1999.

[25] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings Algorithm", The American Statistician, 49, 4, 1995.

[26] T. Chonavel, Statistical Signal Processing, Advanced Text Books In Control and Signal Processing, Springer, 2003.

[27] N. Chopin, "Central Limit Theorem for Sequential Monte Carlo Methods and its Application to Bayesian Inference", Annals of Statistics, 2004.

[28] N. Chopin, "A Sequential Particle Filter Method for Static Models", Biometrika, 89, 3, 539-551, 2002.

[29] D. Crisan and A. Doucet, "A survey of Convergence Results on Particle Filtering Mehtods for Practitioners", IEEE Trans on Signal Processing, Vol. 50, No. 3, 2002.

[30] M. Davy, C. Andrieu and A. Doucet, "Improved Auxiliary Particle Filtering: Applications to Time-Varying Spectral Analysis", IEEE SSP 2001, Singapore, August 2001.

[31] P. Del Moral, A. Doucet and G. W. Peters," Sequential Monte Carlo Samplers", submitted and in review for Journal of Royal Statistical Society Series B, 2004.

[32] P. Del Moral, Feynman-Kac Formulae : Genealogical and Interacting Particle Systems with Applications, Springer, 2004.

[33] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from incomplete data via the EM algorithm", Journal of Royal Statistical Society, B, 39, 1-38.

[34] D. G. Denison, C. C. Holmes, B. K. Mallick and A. F. M. Smith, Bayesian Methods for Nonlinear Classification and Regression, Wiley Series in Probability and Statistics, Wiley, 2002.

[35] L. Devroye, Non-Uniform Random Variate Generation, Springer-Verlag, New York, 1986.

[36] P. Diaconis, S. Holmes and R. Neal, "Analysis of a Non-Reversible Markov Chain Sampler", to appear in Annals of Applied Probability.

[37] A. Doucet, N. de Freitas and N. Gordon, Sequential Monte Carlo Methods in Practice, Statistics for Engineering and Information Science, Springer, 2001.

[38] A. Doucet and S. Senecal, "Fixed-lag Sequential Monte Carlo", Proc. EUSIPCO 2004.

[39] A. Doucet and D. Crisan, "Convergence of Sequential Monte Carlo Methods", Cambridge University, CUED/F-INFENG/TR381, 2000.

[40] A. Doucet and C. Andrieu, "Joint Bayesian Model Selection and Estimation of Noisy Sinusoids via Reversible Jump MCMC", IEEE Trans on Signal Processing, Vol. 47, No. 10, October 1999.

[41] A. Doucet, S. Godsill and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering", Statistics and Computing,10, 197-208, 1998.

[42] A. Doucet, "On Sequential Simulation-Based Methods for Bayesian Filtering", Cambridge University, CUED/F-INFENG/TR310, 1998.

[43] A. Faul and M. E. Tipping, "A Variational Approach to Robust Regression.", In G. Dorffner, H. Bischof, and K. Hornik (Eds.), Proceedings of ICANN'01, 95–102. Springer, 2001.

[44] W. Fitzgerald, "An Introduction to Bayesian Inference Applied to Signal and Data Processing", Reading Group Series, Cambridge University Engineering Department, 2004.

[45] R. Gilks and C. Berzuini, "Following a Moving Target-Monte Carlo Inference for Dynamic Bayesian Models", Journal of Royal Statistical Society Series B, 1999.

[46] W. R. Gilks, S. Richardson and D. J. Spiegelhalter, Markov Chain Monte Carlo in Practice, Chapman and Hall, 1996.

[47] A. Gelman and X. L. Meng, "Simulating Normalizing Constants: From Importance Sampling to Bridge Sampling to Path Sampling", Stat. Science, 13, 163-185, 1998.

[48] A. Gelman, J.B. Carlin, H.S. Stem and D.B. Rubin, Bayesian Data Analysis, Chapman and Hall, 1995.

[49] C.J. Geyer and E. A. Thompson, "Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference.", Journal of American Statistical Association, **90**, 909-920, 1995.

[50] N.J. Gordon, D.J.Salmond, and A.F.M Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation.", IEE-Proc. F, 140(2):107133, 1993.

[51] P. Green, "Trans-Dimensional Markov Chain Monte Carlo", chapter for the book on Highly Structured Stochastic Systems, to be published by OUP, 2003.

[52] P. J. Green, "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination", Biometrika, vol. 82, 711-732, 1995.

[53] U. Grenander and M. I. Miller, "Representations of Knowledge in Complex Systems." Journal of the Royal Statistical Society, B, 56, 549-603, 1994.

[54] D. Harrison and D. L. Rubinfeld, "Hedonic Prices and the Demand for Clean Air", J. Environ. Economics & Management, 5: 81-102, 1978.

[55] W. K. Hastings, "Monte Carlo Sampling Methods using Markov Chains and their Applications", Biometrika, 57, 97-109, 1970.

[56] C. Jarzinski, "Nonequilibrium Equality for Free Energy Differences", Physical Review Leters, 78, 2690-2693, 1997.

[57] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing", Science, Vol. 220, 671-680, 1983.

[58] G. Kitigawa, "Monte Carlo Filter and Smoother for Non-Gaussian Non-Linear State Space Models", Journal of Computational and Graphical Statistics, 5(1):1-25, 1996.

[59] G. Kitagawa, "A Monte Carlo Filtering and Smoothing Method for Non-Gaussian Non-Linear State Space Models.", Proceeding of 2nd US-Japan Joint Seminar on Statistical Time Series Analysis, 110-131, 1993.

[60] R. Kohn, M. Smith and D. Chan, Nonparametric Regression using Linear Combinations of Basis Functions, Statistics and Computing, New York: Springer Verlag, 2001.

[61] A. Kong, J. Liu and W. Wong, "Sequential Imputations and Bayesian Missing Data Problems", Journal of American Statistical Association, Vol. 89, N0. 425, Theory and Methods, 1994.

[62] H. Kunsch, "Recursive Monte Carlo Filters : Algorithms and Theoretical Analysis", Research Report N0. 112, Seminar for Statistics, Switzerland, 2003.

[63] J. Liu, Monte Carlo Strategies in Scientific Computing, Springer, 2001.

[64] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems.", Journal of the American Statistical Association, 93(443):10321044, 1998.

[65] J. Liu, F. Liang and W. H. Wong, "The Use of Multiple-Try Method and Local Optimization in Metropolis Sampling ", Journal of American Statistical Association, 121-134, 95.

[66] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, "Equations of State Calculations by Fast Computing Machine", J. Chem. Phys., 21, 1087-1091, 1953.

[67] S. P. Meyn and R. L. Tweedie, Markov Chains and Stochastic Stability, Springer, 1993.

[68] R. Neal, "Improving Asymptotic Variance of MCMC Estimators : Non-Reversible Chains are Better", Technical Report No. 0406, Department of Statistics, University of Toronto, 2004.

[69] R. Neal, "Annealed Importance Sampling", Statistics and Computing, New York : Springer-Verlag, 2001.

[70] M. K Pitt and N. Shephard, "Filtering via Simulation : Auxiliary Particle Filters" Journal of the American Statistical Association, 94, 590-599, 1999.

[71] A. Raftery and Y. Zheng (2003), "Long Run Performance of Bayesian Model Averaging", Technical Report 433, Department of Stats. Univerity of Washington.

[72] A.E Raftery, "Hypothesis Testing and Model Selection. In Markov Chain Monte Carlo in Practice" (W.R. Gilks, D.J. Spiegelhalter and S. Richardson, eds.), London: Chapman and Hall, 163–188, 1996.

[73] G. Ridgeway and D. Madigan, "Bayesian Analysis of Massive Data Sets via Particle Filters", Proceedings of the Eigth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.

[74] B.D. Ripley, Stochastic Simulation, John Wiley, 1987.

[75] C.P.Robert, The Bayesian Choice, 2nd Ed., Springer Texts in Statistics, 2004.

[76] C.P.Robert and G. Casella, Monte Carlo Statistical Methods, Springer Texts in Statistics, 1999.

[77] G. O. Roberts and J. S, Rosenthal, "Optimal Scaling for Various Metropolis-Hastings Algorithms", Statistical Science, Vol. 16, No. 4, 351-367, 2001.

[78] G. O. Roberts, A. Gelman and W. R. Gilks, "Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms", research report 94.16, Statistical Laboratory, University of Cambridge, 1994.

[79] J. Ruanaidh and W. Fitzgerald, Numerical Bayesian Methods Applied to Signal Processing, Statistics and Computing, Springer, 1996.

[80] J.I. Siepmann and D. Frenkel, "Configurational-Bias Monte Carlo - A New Sampling Scheme for Flexible Chains", Mol. Phys. 75, 59-70, 1992.

[81] A.F.M. Smith and A.E. Gelfand, "Bayesian Statistics without Tears: a Sampling-Resampling Perspective.", American Statistician, 46(2):84–8, May 1992.

[82] D. L. Snyder and M. I Miller, Random Point Processes in Time and Space, 2nd edition, Springer-Verlag, 1975.

[83] J.A. Starck and W.J. Fitzgerald, "Parameter Based Hypothesis Tests for Model Selection", Signal Processing, 46:169-178, 1995.

[84] L. Tierney, "Markov Chains for Exploring Posterior Distributions", The Annals of Statistics, Vol. 22, No. 4, 1701-1762, 1994.

[85] M. E. Tipping and A. C. Faul, "Fast Marginal Likelihood Maximisation for Sparse Bayesian Models", Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, C. M. Bishop and B. J. Frey, 2003.

[86] M. E. Tipping, "The Relevance Vector Machine.", In S. A. Solla, T. K. Leen, and K.-R. Müller (Eds.), Advances in Neural Information Processing Systems 12, 652–658. MIT Press, 2000.

[87] A. W. van der Vaart, Asymptotic Statistics, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 1998.

[88] V. N. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, 1998.

[89] J. Vermaak, S. J. Godsill, and A. Doucet, "Radial Basis Function Regression using Trans-Dimensional Sequential Monte Carlo.", In IEEE Workshop on Statistical Signal Processing, 2003.

[90] J. Vermaak, S. Godsill, and A. Doucet. "Sequential Bayesian Kernel Regression.", In S. Thrun, L. Saul, and B. Scholkopf, editors, Advances in Neural Information Processing Systems 16, 2004.

[91] R. Waagpetersen and D. Sorensen, "A Tutorial on Reversible Jump MCMC with a view toward QTL-Mapping.", International Statistical Review, 69, 49-61, 2001.

[92] M. West, "Approximating Posterior Distributions by Mixture", Journal of Royal Statistical Society. Series B, 55:409-422, 1993.