Computational Intelligence Sequential Monte Carlos for Recursive Bayesian Estimation

Muhammed Ayub Hanif

A thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

University College London

Declaration

I, Ayub Hanif, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Ayub Hanif

Abstract

Recursive Bayesian estimation using sequential Monte Carlos methods is a powerful numerical technique to understand latent dynamics of non-linear non-Gaussian dynamical systems. Classical sequential Monte Carlos suffer from weight degeneracy which is where the number of distinct particles collapse. Traditionally this is addressed by resampling, which effectively replaces high weight particles with many particles with high inter-particle correlation. Frequent resampling, however, leads to a lack of diversity amongst the particle set in a problem known as sample impoverishment. Traditional sequential Monte Carlo methods attempt to resolve this correlated problem however introduce further data processing issues leading to minimal to comparable performance improvements over the sequential Monte Carlo particle filter.

A new method, the adaptive path particle filter, is proposed for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems. Our method addresses the weight degeneracy and sample impoverishment problem by embedding a computational intelligence step of adaptive path switching between generations based on maximal likelihood as a fitness function.

Preliminary tests on a scalar estimation problem with non-linear non-Gaussian dynamics and a non-stationary observation model and the traditional univariate stochastic volatility problem are presented. Building on these preliminary results, we evaluate our adaptive path particle filter on the stochastic volatility estimation problem. We calibrate the Heston stochastic volatility model employing a Markov chain Monte Carlo on six securities. Finally, we investigate the efficacy of sequential Monte Carlos for recursive Bayesian estimation of astrophysical time series. We posit latent dynamics for both regularized and irregular astrophysical time series, calibrating fifty-five quasar time series using the CAR(1) model. We find the adaptive path particle filter to statistically significantly outperform the standard sequential importance resampling particle filter, the Markov chain Monte Carlo particle filter and, upon Heston model estimation, the particle learning algorithm particle filter. In addition, from our quasar MCMC calibration we find the characteristic timescale τ to be first-order stable in contradiction to the literature though indicative of a unified underlying structure. We offer detailed analysis throughout, and conclude with a discussion and suggestions for future work.

Acknowledgements

Foremost, I would like to express my deepest appreciation to my supervisors Dr. Robert Elliott Smith and Dr. Piotr Karasinski for their enthusiasm, guidance, motivation and knowledge. Their patience and guidance has seen me through this research and writing. In addition, I would like to thank Prof. David Rosenblum and Prof. Anthony Finkelstein for their guidance at inception and for availing me this opportunity. I would like to thank Prof. John Shawe-Taylor for entertaining my musings on time series analysis and his invaluable guidance and advice.

I would like to thank my advisor at Harvard IACS, Dr. Pavlos Protopapas for his enthusiasm and undying energy to see our work through. For pursuing and persevering when all else had failed. For this I am grateful. To Prof. Efthimios Kaxiras for his warm welcome and support during my time at Harvard. And to Christie Gilliland for providing me sanity in the dungeon, and for seeing my ray of light. My work at Harvard could not have been possible without the support of my dear friend Alberto G. Brugnoni. May your future be prosperous and showered with bounteous mercy and blessings.

I would like to thank Shakir Hussain for his assistance and guidance through calibration and testing. And to, Abdigani Diriye: you my friend I am indebted to the most. You believed in me when the chips were down. May all your expectations be exceeded and may our fellowship thrive.

In dilectione mea.

Contents

List of Figures

Modeling & Estimation in Astrophysics

Discussion

List of Tables

Chapter 1

Introduction

The frequentist approach to probability, the populist view, is whereby the probability of an uncertain event is defined by the frequency of that event based on previous observations. Such an approach is sufficient provided you have an accurate account of many past instances of the event. Bayesian probability theory enables plausible reasoning from logical consistency, where we reason in situations where we cannot argue with certainty (as in the frequentist approach). It is frequently conflated to be subjective in contrast to the objectivity afforded to the frequentist approach. However, the subjectivity in the Bayesian approach is a mis-drawn fact, mainly by scholars of competing schools. Bayesian probability does indicate a degree-of-belief but this is conditioned on ones knowledge, as in rational actors, which is why Bayesian statistics always explicitly states the conditioning (Sivia & Skilling 2006). Our research takes the Bayesian approach, enabling us to reason about uncertainty in our beliefs in the temporal evolution of data.

A time series is sequentially ordered data which could take discrete or continuous values. Time series analysis, the process of discerning patterns within observed serial data, is both well known and widely studied across disciplines. State space modeling of time series enables us to reason probabilistically over time under a Bayesian framework. Here, Bayesian inference aims to elucidate sufficient variables which accurately describe the dynamics.

Stochastic filtering is the process of recovering the latent state variable by removing observation errors and computing the joint posterior distribution over the most recent state. Bayesian filtering adopts Bayesian reasoning within the stochastic filtering paradigm to approximate calculation of the joint posterior distribution. This process is repeated upon every new observation to allow us to revise our understanding. This iterative, recursive Bayesian filtering process enables us to estimate the latent state variable through the observation of a large quantity of data. The Bayesian recursions include a prediction and update step: we estimate the latent state a priori and using this prediction and a new observation obtain an a posteriori estimate.

Linear estimators do not provide closed-form solutions to distribution approximations frequently encountered in real-world problems. Sequential Monte Carlo approximation is a numerical method for optimal estimation in non-linear non-Gaussian settings. The main advantage of sequential Monte Carlos for recursive Bayesian estimation is that they do not rely on any local linearization or abstract functional approximation. Here, the state space is populated with particles weighted according to some probability measure. The particle system evolves and adapts to the temporal evolution of the state space.

The main problem with sequential Monte Carlo methods is weight degeneracy. This is where the particle systems collapses onto a few non-zero importance weights. Typically degeneracy is addressed using resampling - replacing particles with high weight with many particles with high inter-particle correlation - though this leads to a lack of diversity amongst the particles (a problem known as sample impoverishment). Traditional sequential Monte Carlo methods attempt to resolve these correlated problems however introduce further data processing issues resulting in minimal to comparable performance improvements over the traditional sequential Monte Carlo particle filter. There have been a number of attempts to address this problem by hybridizing the original particle filtering algorithm with ideas from evolutionary computation and genetic algorithms with application to a variety of fields and problems.

1.1 Motivation & Approach

Recursive Bayesian estimation is a very important task in many real-world applications. It enables us to reason under uncertainty and addresses shortcomings underlying deterministic systems and control theories which do not provide sufficient means of performing analysis and design (Maybeck 1979). In addition, parametric techniques such as the Kalman filter and its extensions, though they are computationally efficient, do not reliably compute states and cannot be used to learn stochastic problems where the underlying process is non-linear and non-Gaussian.

For instance, sequential Monte Carlo methods have long been applied to the stochastic volatility estimation problem in computational finance. Latent dynamics of the stochastic volatility process are known to be highly complex, commonly modeled as mean-reverting square-root diffusions. It has real-world implications both for the market as a whole and as for the institution performing the estimation. Clearly, it is imperative to have the most accurate estimate to aid correct derivative pricing, securities trading and to manage exposure to the market. Similarly, there are many non-linear non-Gaussian dynamical systems in the physi-

cal world around us where mathematical models posited by deterministic system and control theories simply fail.

Our research is primarily focused on recursive Bayesian estimation of non-linear non-Gaussian dynamical systems. We use discrete time observations to discern the stochastic process evolution and benchmark these time series using a Markov chain Monte Carlo to calibrate the underlying state space model. This allows us to estimate the parameter and state estimation posteriors. Thereafter we run sequential Monte Carlos and compare estimation accuracy in aid of providing practical results for direct application in real-world settings.

1.2 Problem Statement

Stochastic filtering and optimal estimation of complex distributions is a particularly pertinent problem in real-world applications. Recursive Bayesian estimation using sequential Monte Carlo methods is a common numerical technique, used to solve non-linear non-Gaussian estimation problems where linear estimators fail.

Despite a concerted effort from within the sequential Monte Carlo community to address weight degeneracy and sample impoverishment, the major open problem within the sequential Monte Carlo domain, there still lacks a powerful filter which addresses this problem conclusively and provides statistically significant improvements in estimation accuracy. We propose to address the weight degeneracy and sample impoverishment problem by embedding an evolutionary computation step (a heuristic selection scheme) of adaptive path switching between generations based on maximal likelihood as a fitness function into the new *adaptive path* particle filter (APPF).

Our research hypothesizes our APPF will yield increased accuracy for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems compared to contemporary filters. We shall test this on simulated and real financial securities time series, estimating the stochastic volatility. In addition we shall test this in combination with assessing the efficacy of sequential Monte Carlo methods for astrophysical time series analysis.

1.3 Contribution

This thesis provides a novel sequential Monte Carlo method which leverages a computational intelligence step of adaptive path switching between generations based on maximal likelihood as a fitness function to yield enhanced estimation accuracy for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems compared to contemporary filters, and an assessment of the efficacy of the use of sequential Monte Carlo methods for modeling astrophysical time series.

This thesis makes the following contributions:

1. The development of a new sequential Monte Carlo method based on computational intelligence for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems.
   (a) Outperformed contemporary filters in a scalar estimation problem and the univariate log-stochastic volatility estimation problem.
   (b) Successfully addresses weight degeneracy and sample impoverishment problem of traditional sequential Monte Carlo methods.
2. The application of a new sequential Monte Carlo method to the stochastic volatility problem.
   (a) Calibrated on Heston stochastic volatility model.
   (b) Outperformed contemporary filters in estimation of six securities.
3. The pioneering application of sequential Monte Carlo methods to astrophysical time series analysis.
   (a) Postulated latent dynamics of regularized and irregular quasar time series.
   (b) Our novel sequential Monte Carlo method outperformed contemporary filters in estimation of regularized quasar time series.
   (c) Our novel sequential Monte Carlo method outperformed contemporary filters in estimation of irregular quasar time series.
      i. Calibrated CAR(1) model on fifty-five quasar time series.
      ii. Found the characteristic timescale τ of quasars to be first-order stable.

1.4 Thesis Outline

The thesis is organized as follows:

- Chapter 2 proceeds to detail state space modeling and Bayesian inference. It sets out the material relevant to our substantive work including an exposition of sequential Monte Carlo methods, Markov chain Monte Carlo and computational intelligence for optimization. We highlight and discuss open problems in the sequential Monte Carlo community.
- Chapter 3 introduces our model including its conceptual underpinnings. The model is formally justified alongside its introduction. We provide some preliminary results on a scalar estimation problem and on the log-stochastic volatility estimation problem.

- Chapter 4 provides details of our stochastic volatility modeling experimental results.
- In Chapter 5 we detail our investigation into the efficacy of sequential Monte Carlo methods for astrophysical time series analysis. We provide details of quasar light curve estimation using sequential Monte Carlo methods..
- Chapter 6 discusses our work in context, focusing on our methodology, significance and contributions.
- Chapter 7 provides summary conclusions of our work and discusses avenues for future work.

1.5 Publications

- A. Hanif and R. Smith. "Stochastic Volatility Modeling with Computational Intelligence Particle Filters." Genetic and Evolutionary Computation Conference (GECCO), ACM, 2013
- A. Hanif and P. Protopapas. "Recursive Bayesian Estimation of Regularized and Irregular Astrophysical Time Series." Monthly Notices of the Royal Astronomical Society, submitted. 2013
- A. Hanif and R. Smith. "Selection Schemes in Sequential Monte Carlos for Stochastic Volatility Estimation." IEEE Transactions on Cybernetics, submitted. 2013.
- A. Hanif and R. Smith. "Generation Based Path-Switching in Sequential Monte-Carlo Methods." IEEE Congress on Evolutionary Computation (CEC), 2012 , pages1–7. IEEE, 2012.

Chapter 2

State Space Models & Bayesian Inference

In this chapter we detail the background to state space modeling and Bayesian inference. We begin in Section 2.1 by formally introducing the underlying concepts. We discuss control theory concepts and proceed to describe stochastic filtering. In Section 2.2 we detail sequential Monte Carlo methods, exploring the state-of-the-art of these algorithms for recursive Bayesian estimation. We move onto describe Markov chain Monte Carlo methods in Section 2.3. These batch processing techniques are used to learn latent components of complex distributions where we either know or do not know the complete latent conditional distributions. In Section 2.4 we introduce computational intelligence methods, and proceed to review the existing literature on the symbiosis of computational intelligence and sequential Monte Carlo methods. We conclude this Chapter in Section 2.5 by discussing the state-of-the-art of both sequential Monte Carlo methods and synergistic computational intelligence methods, highlighting open problems.

2.1 Stochastic Models, Estimation and Control Theory

To facilitate probabilistic reasoning over time we will be adopting state space representations under a Bayesian framework. A state space model of a time series $\{\boldsymbol{y}_t : t = 1, 2, ...\}$ is composed of two equations: the state equation and the observation equation. The observation equation relates the observed data $\{\boldsymbol{y}_t\}$ to the latent states $\{\boldsymbol{x}_t : t = 1, 2, ...\}$. Consider the discrete time estimation problem (Gordon, et al. 1993) with the system model:

$$\boldsymbol{y}_t = h_t(\boldsymbol{x}_t, \boldsymbol{v}_t) \tag{2.1}$$

$$\boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{w}_t) \tag{2.2}$$

in which we represent the observation vector at time t by $\boldsymbol{y}_t \in \mathbb{R}^p$, which satisfies (2.1) where $h_t : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^p$ is the observation function and $\boldsymbol{v}_t \in \mathbb{R}^r$ is the state error term whose known distribution is independent of both system noise and time. Similarly, we represent the state vector at time t by $\boldsymbol{x}_t \in \mathbb{R}^n$, which satisfies (2.2) where $f_t : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is the system transition function and $\boldsymbol{w}_t$ is an error term whose known distribution is temporally independent.

There are very few system models for which an analytic solution is available. Consider the linear Gaussian model:

$$\boldsymbol{Y}_t = H_t \boldsymbol{X}_t + \boldsymbol{W}_t \tag{2.3}$$

$$\boldsymbol{X}_{t+1} = F_t \boldsymbol{X}_t + \boldsymbol{V}_t \tag{2.4}$$

where (2.3) is the measurement equation with $\{\boldsymbol{W}_t\} \sim \mathcal{WN}(\boldsymbol{0}, R_t)$ observation errors; and (2.4) is the state equation with $\{\boldsymbol{V}_t\} \sim \mathcal{WN}(\boldsymbol{0}, Q_t)$ state description errors. We assume that the observation errors and state errors are uncorrelated. That is $\forall s, \forall t : \mathbb{E}[\boldsymbol{W}_s \boldsymbol{V}_t] = 0$. Finally, the initial state $\boldsymbol{X}_0$ is assumed to be uncorrelated with all errors $\{\boldsymbol{V}_t\}$ and $\{\boldsymbol{W}_t\}$. Here, the Kalman filter (KF) provides an analytic, optimal linear solution. However, such assumptions of linearity and Gaussianity do not hold in most real-world applications which conversely are of concern to us necessitating an investigation into models of estimation which can be used practically.

Bayesian inference aims to elucidate sufficient variables which accurately describe the dynamics of the process being modeled. In such temporal models there are three major inference problems (Russell, et al. 2010, Tsay 2010):

- *Filtering*: recovering the state variable $\boldsymbol{x}_t$ given F_t with data up to and including time t . To essentially remove observation errors and compute the posterior distribution over the most recent state: $P(\boldsymbol{X}_t | \boldsymbol{Y}_{0:t})$.

- *Prediction*: forecasting the observation $\boldsymbol{y}_{t+h}$ for $h > 0$ and F_t given all observations up to t . To calculate the posterior distribution over the future state given all evidence up to date: $P(\boldsymbol{X}_{t+h} | \boldsymbol{Y}_{0:t})$ where $h > 0$.

- *Smoothing*: to estimate the state $\boldsymbol{x}_t$ given F_T, where $T > t$. To calculate the posterior distribution over a past state given all evidence up to the present: $P(\boldsymbol{X}_t | \boldsymbol{Y}_{0:T})$ where $T > t$.

Stochastic filtering underlies Bayesian filtering and is an inverse statistical problem: you want to find inputs as you are given outputs (Chen 2003). The principle foundation of stochastic filtering lies in recursive Bayesian estimation where we are essentially trying to compute the joint posterior. There are two key assumptions in deriving the recursive Bayesian filter: (i) that the state process follows a first-order Markov process:

$$(\boldsymbol{x}_n | \boldsymbol{x}_{0:n-1}, \boldsymbol{y}_{0:n-1}) = p(\boldsymbol{x}_n | \boldsymbol{x}_{n-1}) \tag{2.5}$$

and (ii) that the observations and states are independent:

$$(\boldsymbol{y}_n|\boldsymbol{x}_{0:n-1}, \boldsymbol{y}_{0:n-1}) = p(\boldsymbol{y}_n|\boldsymbol{x}_n) \tag{2.6}$$

For simplicity, we shall denote $\mathcal{Y}_n$ as the set of observations $\boldsymbol{y}_{0:n} := \{\boldsymbol{y}_0, ..., \boldsymbol{y}_n\}$ and $p(\boldsymbol{x}_n|\mathcal{Y}_n)$ as the conditional probability density function (pdf) of $\boldsymbol{x}_n$. From Bayes rule we have:

$$\begin{aligned} p(\boldsymbol{x}_n|\mathcal{Y}_n) &= \frac{p(\mathcal{Y}_n|\boldsymbol{x}_n)p(\boldsymbol{x}_n)}{p(\mathcal{Y}_n)} \\ &= \frac{p(\boldsymbol{y}_n, \mathcal{Y}_{n-1}|\boldsymbol{x}_n)p(\boldsymbol{x}_n)}{p(\boldsymbol{y}_n, \mathcal{Y}_{n-1})} \\ &= \frac{p(\boldsymbol{y}_n|\mathcal{Y}_{n-1}, \boldsymbol{x}_n)p(\mathcal{Y}_{n-1}|\boldsymbol{x}_n)p(\boldsymbol{x}_n)}{p(\boldsymbol{y}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1})} \\ &= \frac{p(\boldsymbol{y}_n|\mathcal{Y}_{n-1}, \boldsymbol{x}_n)p(\boldsymbol{x}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1})p(\boldsymbol{x}_n)}{p(\boldsymbol{y}_n|\mathcal{Y}_{n-1})p(\mathcal{Y}_{n-1})p(\boldsymbol{x}_n)} \\ &= \frac{p(\boldsymbol{y}_n|\boldsymbol{x}_n)p(\boldsymbol{x}_n|\mathcal{Y}_{n-1})}{p(\boldsymbol{y}_n|\mathcal{Y}_{n-1})} \end{aligned} \tag{2.7}$$

We see from (2.7) the joint posterior density $p(\boldsymbol{x}_n|\mathcal{Y}_n)$ is described by three key terms:

- *Prior*: the knowledge of the model is described by the prior $p(\boldsymbol{x}_n|\mathcal{Y}_{n-1})$

$$p(\boldsymbol{x}_n|\mathcal{Y}_{n-1}) = \int p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})p(\boldsymbol{x}_{n-1}|\mathcal{Y}_{n-1})d\boldsymbol{x}_{n-1} \tag{2.8}$$

- *Likelihood*: $p(\boldsymbol{y}_n|\boldsymbol{x}_n)$ essentially determines the observation noise in (2.1).

- *Evidence*: the denominator of the pdf, involves an integral of the form

$$p(\boldsymbol{y}_n|\mathcal{Y}_{n-1}) = \int p(\boldsymbol{y}_n|\boldsymbol{x}_n)p(\boldsymbol{x}_n|\mathcal{Y}_{n-1})d\boldsymbol{x}_n \tag{2.9}$$

The calculation and or approximation of these three terms is the base of Bayesian filtering and inference.

2.2 Sequential Monte Carlo Methods

There are two approaches to obtain the posterior distribution of concern defined in (2.7): parametric Gaussian approximation or non-parametric approximation using Monte Carlo techniques (Nikolaev & Smirnov 2007). Though they are computationally efficient, parametric techniques, the KF and its extensions, do not reliably compute states and cannot be used to learn stochastic problems where the underlying process is non-linear and non-Gaussian. Anderson & Moore (1979) highlight some applications of Kalman techniques though, conversely, highlight difficulties in calculating closed-form solutions to distribution approximations and propose application of numerical methods to overcome these difficulties.

Sequential Monte Carlo approximation, a numerical method, of optimal estimation problems in non-linear non-Gaussian settings is commonly performed using particle methods (Chen 2003, Gordon et al. 1993, Kitagawa 1996, Liu & Chen 1998, Carpenter, et al. 1999, Bauwens, et al. 1999, Doucet, et al. 2001, Maskell 2004). The main advantage of these methods is that they do not rely on any local linearization or abstract functional approximation. This is at the cost of increased computational expense however, given breakthroughs in computing technology and the related decline in processing costs, this is not considered a barrier except in extreme circumstances.

2.2.1 Importance Sampling

Monte Carlo approximation using particle methods calculates the expectation of the pdf by sampling (Chen 2003). The state space is populated with particles weighted according to some probability measure. The higher this measure the denser the particle concentration. The state space evolves temporally with the particle system evolving around this. Specifically, from (2.7) and (Nikolaev & Smirnov 2007):

$$p(x_t|\boldsymbol{y}_{0:t}) = \frac{p(y_t|x_t)p(x_t|\boldsymbol{y}_{0:t-1})}{p(y_t|\boldsymbol{y}_{0:t-1})} \tag{2.10}$$

where $p(x_t|\boldsymbol{y}_{0:t})$ is the state posterior (filtering distribution), $p(y_t|x_t)$ is the likelihood, $p(x_t|\boldsymbol{y}_{0:t-1})$ is the state prior (predictive distribution) and the denominator $p(y_t|\boldsymbol{y}_{0:t-1})$ is the evidence. The state prior is defined by: $p(x_t|\boldsymbol{y}_{0:t-1}) = \int p(x_t|x_{t-1}, \boldsymbol{y}_{0:t-1})p(x_{t-1}|\boldsymbol{y}_{0:t-1})dx_{t-1}$ where $p(x_t|x_{t-1}, \boldsymbol{y}_{0:t-1})$ is the transition density and $p(x_{t-1}|\boldsymbol{y}_{0:t-1})$ is the previous filtering distribution.

We approximate the state posterior by $f(x_t)$ with i samples of $x_t^{(i)}$. To find the mean $\mathbb{E}[f(x_t)]$ of the state posterior $p(x_t|\boldsymbol{y}_{0:t})$ at t, we generate the state samples $x_t^{(i)} \sim p(x_t|\boldsymbol{y}_{0:t})$. Though theoretically plausible, empirically we are unable to observe and sample directly from the state posterior. We replace the state posterior by a proposal state distribution (importance distribution) π which is proportional to the true posterior at every point: $\pi(x_t|\boldsymbol{y}_{0:t}) \propto p(x_t|\boldsymbol{y}_{0:t})$. We are thus able to sample sequentially independently and identically distributed (i.i.d.) draws from $\pi(x_t|\boldsymbol{y}_{0:t})$ giving us:

$$\begin{aligned} \mathbb{E}[f(x_t)] &= \int f(x_t)\frac{p(x_t|\boldsymbol{y}_{0:t})}{\pi(x_t|\boldsymbol{y}_{0:t})}\pi(x_t|\boldsymbol{y}_{0:t})dx_t \\ &\approx \frac{\sum_{i=1}^{N} f(x_t^{(i)})w_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}} \end{aligned} \tag{2.11}$$

When increasing the number of draws N this average converges asymptotically (as $N \to \infty$) to the expectation of the true posterior according to the central limit theorem (Geweke 1989).

This convergence is the primary advantage of sequential Monte Carlo methods as they provide asymptotically consistent estimates of the true distribution $p(x_t|\mathbf{y}_{0:t})$ (Doucet & Johansen 2008).

The foundation of common particle methods is importance sampling (IS) which relies on the introduction of an importance density $q_t(x_{1:t})$ where given:

$$\pi_t(x_{1:t}) > 0 \Rightarrow q_t(x_{1:t}) > 0$$

we have the IS identities

$$\pi_t(x_{1:t}) \equiv \frac{w_t(x_{1:t})q_t(x_{1:t})}{Z_t}$$
$$Z_t \equiv \int w_t(x_{1:t})q_t(x_{1:t})dx_{1:t}$$

where $w_t(x_{1:t})$ is the unnormalized weight function, with $\gamma_t : \chi^t \to \mathbb{R}^+$ known pointwise:

$$w_t(x_{1:t}) = \frac{\gamma_t(x_{1:t})}{q_t(x_{1:t})}$$

2.2.2 Sequential Importance Sampling

IS allows us to sample from complex highly-dimensional distributions though exhibits linear increases in complexity upon each subsequent draw (Doucet & Johansen 2008). To admit fixed computational complexity we use sequential importance sampling (SIS). Given a general state space model of the form (2.1), (2.2) where the proposal distribution is Markovian, SIS is defined as in Table 2.1. Thus a weight reflects the particle density in the surrounding state space.

The SIS approach is very sensitive to the choice of proposal density with typical choices being the transition probability of states (Kitagawa 1996, Freitas, et al. 2000). This proposal density minimizes the variance of weights critical to the asymptotic convergence of the algorithm. However, SIS suffers from some critical issues. Primarily, the variance of estimates increases exponentially with n and leads to fewer and fewer non-zero importance weights (Doucet & Johansen 2008). This problem is known as weight degeneracy. To alleviate this issue, states are resampled to retain the most pertinent contributors, essentially removing particles with low weights with a high degree of certainty (Gordon et al. 1993). It addresses degeneracy by replacing particles with high weight with many particles with high inter-particle correlation (Chen 2003).

2.2.3 Sequential Importance Resampling

There are a number of resampling schemes that can be adopted. The three most common unbiased schemes are systematic, residual and multinomial. Of these multinomial is the most

Table 2.1: SIS Algorithm

1. Initialization: generate the prior state distribution

$$X_0^{(i)} \sim q(x_0)$$

and propagate the particles to compute the weights

$$w_0(X_0^{(i)})$$
$$W_0^{(i)} \propto w_0(X_0^{(i)})$$

and the likelihoods

$$y_t = h_t(x_t^{(i)}, w_t) \leftrightarrow p(y_t|x_t^{(i)})$$

2. Importance sampling: For each subsequent time step, sample

$$X_t^{(i)} \sim q(x_t|X_{0:t-1}^{(i)})$$

3. Weight update: update weights by the likelihood, given by the definition $w_t = p(x_t|\mathbf{y}_{0:t})/q(x_t|\mathbf{y}_{0:t})$

$$w_t^{(i)} \approx w_{t-1}^{(i)} \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{t-1}^{(i)}, y_t)}$$
$$\approx w_{t-1}^{(i)} p(y_t|x_t^{(i)})$$

4. Repeat from importance sampling step.

computational efficient though systematic resampling is the most commonly used and performs better in most, but not all, scenarios compared to other unbiased sampling schemes (Douc & Cappé 2005). Resampling does add noise (Chopin 2004) however, in a sequential framework computational effort is focused on areas of high probability mass. It is theoretically possible to classify particles as false-positives though the resampling step overcomes these issues. It provides future stability at the cost of increased short term variance (Doucet & Johansen 2008). We provide the sequential importance resampling (SIR) algorithm using the transition prior as the proposal distribution (Chen 2003) in Table 2.2.

Resampling is at the cost of added variance, so it is best to resample only when degeneracy passes a certain threshold for a given measure. The effective sample size (ESS) is defined at time t:

$$N_{eff} = \left(\sum_{i=1}^{N} (W_n^{(i)})^2 \right)^{-1} \tag{2.12}$$

and measures the variability of unnormalized weights and allows us to measure degeneration (Kong, et al. 1994). We provide the full adaptive SIR algorithm using the transition prior as

Table 2.2: SIR Algorithm

1. Initialization: for $i = 1, ..., N_p$, sample

$$\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0)$$

with weights $W_0^{(i)} = \frac{1}{N_p}$.

For $t \geq 1$

2. Importance sampling: for $i = 1, ..., N_p$, draw samples

$$\hat{\boldsymbol{x}}_t^{(i)} \sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^{(i)})$$

set $$\hat{\boldsymbol{x}}_{0:t}^{(i)} = \{\boldsymbol{x}_{0:t-1}^{(i)}, \hat{\boldsymbol{x}}_t^{(i)}\}$$

3. Weight update: calculate the importance weights

$$W_t^{(i)} = p(\boldsymbol{y}_t | \hat{\boldsymbol{x}}_t^{(i)})$$

4. Normalize weights:

$$\tilde{W}_t^{(i)} = \frac{W_t^{(i)}}{\sum_{j=1}^{N_p} W_t^{(j)}}$$

5. Resampling: Generate N_p new particles $\boldsymbol{x}_t^{(i)}$ from the set $\{\hat{\boldsymbol{x}}_t^{(i)}\}$ according to the importance weights $\tilde{W}_t^{(i)}$.

6. Repeat from importance sampling step.

the proposal distribution (Chen 2003) in Table 2.3. Unfortunately, frequent resampling leads to a lack of diversity amongst particles in a problem known as sample impoverishment (when a particle set is impoverished there are many repeated points). Thus the weight degeneracy and sample impoverishment are part of one larger correlated problem.

2.2.4 Auxiliary Particle Filter

Despite circumvention schemes (i.e. asymptotic justification) it is inherently impossible to represent a highly-dimensional distribution by a finite sample set, however ergodicity of the underlying process will smooth and prevent errors accumulating over time. Application of resampling methods before importance weight calculations minimizes the loss of information. The SIR algorithm uses *ex-ante* information from the likelihood model to inform sampling, avoiding sampling of low likelihood and thus less informative particles. A generalization of this *ex-ante* information usage would be advantageous in situations where it is not possible to make use of the optimal proposal distribution (Doucet & Johansen 2008, Chen 2003). We shall now introduce an algorithm which achieves just this.

Table 2.3: SIS/R Algorithm

1. Initialization: for $i = 1, ..., N_p$, sample

$$\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0)$$

with weights $W_0^{(i)} = \frac{1}{N_p}$.

For $t \geq 1$

2. Importance sampling: for $i = 1, ..., N_p$, draw samples

$$\hat{\boldsymbol{x}}_t^{(i)} \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(i)}, \boldsymbol{y}_{o:n})$$

set $$\hat{\boldsymbol{x}}_{0:t}^{(i)} = \{\boldsymbol{x}_{0:t-1}^{(i)}, \hat{\boldsymbol{x}}_t^{(i)}\}$$

3. Weight update: calculate the importance weights

$$W_t^{(i)} = W_{t-1}^{(i)} p(\boldsymbol{y}_t|\hat{\boldsymbol{x}}_t^{(i)})$$

4. Normalize weights:

$$\tilde{W}_t^{(i)} = \frac{W_t^{(i)}}{\sum_{j=1}^{N_p} W_t^{(j)}}$$

5. Resampling: If $N_{eff} < N_{thres}$: generate N_p new particles $\boldsymbol{x}_t^{(i)}$ from the set $\{\hat{\boldsymbol{x}}_t^{(i)}\}$ according to the importance weights $\tilde{W}_t^{(i)}$, return otherwise.

6. Repeat from importance sampling step.

The general particle filter we have detailed suffers from two critical problems (Pitt & Shephard 1999). Firstly, when we have outliers the distribution of weights will become uneven and will lead to a breakdown of the particle system, requiring larger and larger values of N_p draws to approximate samples from the target filtering density. The second, and most debilitating weakness, is as we are using crude approximation techniques the tails of the distribution are poorly approximated. This is particularly pertinent in time series whose distributions are fat-tailed.

The auxiliary particle filter (APF) provides an approach to reduce the effects of outliers whilst handling fat-tailed distributions. It was introduced by Pitt & Shephard (1999) and, as the SIR, focuses on particles in high probability regions of the state space. The APF essentially simulates particles using auxiliary indices to highlight regions and informative particles (Nikolaev & Smirnov 2007). Given the filtering density:

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_{0:t}) &\propto p(\boldsymbol{x}_t|\boldsymbol{y}_t) \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{0:t-1}) d\boldsymbol{x}_{t-1} \\ &\propto \sum_{i=1}^{N_p} W_{t-1}^{(i)} p(\boldsymbol{y}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(i)}) \end{aligned} \tag{2.13}$$

we introduce an auxiliary variable ξ where $\xi \in \{1, ..., N_p\}$ which allows us to express an augmented joint probability density:

$$p(\boldsymbol{x}_t, \xi = i|\boldsymbol{y}_{0:t}) \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t, \xi = i|\boldsymbol{y}_{0:t-1}) \tag{2.14}$$

Combining (2.13) and (2.14) we can write the filtering density as the approximation:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{0:t}) \propto \sum_{i=1}^{N_p} w_{t-1}^{(i)} p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)}, \xi^i) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(i)}) \tag{2.15}$$

where the auxiliary variable i are indices of the particles from the previous time step $t-1$. The proposal distribution is thus:

$$q(\boldsymbol{x}_t, \xi|\boldsymbol{y}_{0:t}) \propto q(\xi|\boldsymbol{y}_{0:t}) q(\boldsymbol{x}_t|\xi, \boldsymbol{y}_{0:t}) \tag{2.16}$$

where:

$$q(\xi^i|\boldsymbol{y}_{0:t}) \propto p(\boldsymbol{y}_t|\mu_t^{(i)}) w_{t-1}^{(i)} \tag{2.17}$$

$$q(\boldsymbol{x}_t|\xi^i, \boldsymbol{y}_{0:t}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(i)}) \tag{2.18}$$

where $\mu_t^{(i)}$ is the point estimate mean, mode or sample value. The auxiliary variable ξ allows us to obtain initial point estimates $\mu_t^{(i)}$ that characterize the transition prior $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(i)})$ to evaluate predictive likelihoods of particles $p(\boldsymbol{y}_t|\mu_t^{(i)})$. These are used to compute simulation weights $w_{t-1}^{(i)} p(\boldsymbol{y}_t|\mu_t^{(i)})$ which are normalized and passed to the sampling algorithm to draw states $x_t^{(i)} \sim q(\boldsymbol{x}_t|\xi^i, \boldsymbol{y}_{0:t}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{\xi^i})$. These simulated particles with indices ξ^i from $t-1$ are likely to be close to the true state (Nikolaev & Smirnov 2007).

The true posterior is approximated by:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{0:t}) \propto \sum_{i=1}^{N_p} w_{t-1}^{(i)} p(\boldsymbol{y}_t|\mu_t^{(\xi=i)}) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(\xi=i)}) \tag{2.19}$$

and from (2.17) and (2.18) the importance weights are recursively updated by:

$$\begin{aligned} w_t^{(i)} &= w_{t-1}^{(\xi=i)} \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)}) p(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(\xi=i)})}{q(\boldsymbol{x}_t^{(i)}, \xi^i|\boldsymbol{y}_{0:t})} \\ &\propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)})}{p(\boldsymbol{y}_t|\mu_t^{(\xi=i)})} \end{aligned} \tag{2.20}$$

The APF takes advantage of *ex-ante* information however, when the process noise is large can fail to focus on pertinent particles and the difference between the PF and APF will be insignificant (Arulampalam, et al. 2002). Where process noise is small the APF performs significantly better. Additionally, APF is more flexible and reliable than the generic PF with typically lower variance results.

2.2.5 Gaussian Mixture Particle Filters

Kotecha & Djuric (2003a) introduce Gaussian particle filters (GPF) for filtering dynamic state space models. The proposed models have Gaussian noise terms with non-linear functions in the state and measurement equations. The underlying idea which forms the base of the GPF, and the Gaussian sum particle filter (GSPF) which we shall see later, comes from classical stochastic control theory. It states that the predictive and filtering distributions in dynamical models can be approximated as Gaussians (mixtures of the normal distribution) (Anderson & Moore 1979). We are motivated to use the posterior mean computed using the GPF as it converges asymptotically to the minimum mean square error (MMSE) estimate, something which is not exhibited in extensions of the KF (Haykin 2001, Wan & Van Der Merwe 2001).

The GPF delivers more accurate mean and variance estimates than other particle filters with the same number of particles, even in the face of severe non-linearities and noise, however is sensitive to outliers as the state noise is assumed to be normal (Kotecha & Djuric 2003a). To address this problem Kotecha & Djuric (2003b) model the state noise as another mixture density, bringing further flexibility to the algorithm. In the GSPF the state noise ε_t is a finite Gaussian mixture

$$\varepsilon_t = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\varepsilon_{tk}; \hat{\varepsilon}_{tk}, \Sigma_{tk})$$

where $\hat{\varepsilon}_{tk}$ is the mean noise and Σ_{tk} is the noise covariance matrix of the k-th mixand.

Kotecha & Djuric (2003b) show the GSPF exhibits greater performance and more accurate estimation than GPF where the noise is fat-tailed and non-Gaussian as fat-tailed densities can be modeled by Gaussian mixtures (Sengupta & Kay 1989, Kitagawa 1996). In addition, they show particle based Gaussian mixture filters perform better in approximation and computational cost than Kalman filter based Gaussian mixture filters.

2.3 Markov Chain Monte Carlo Methods

We have discussed a number of problems with particle methods and shall focus on a number of these limitations to motivate further techniques (Doucet & Johansen 2008). Use of the optimal importance distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{x}_{t-1})$ does not guarantee the efficiency of sequential Monte Carlo algorithms as when the variance of $p(\boldsymbol{y}_t|\boldsymbol{x}_{t-1})$ is high, the variance of the approximation shall be high. This shall result in frequent resampling and the particle approximation of the joint distribution shall be unreliable. Corollary, as $k \ll n$ the marginal distribution $\hat{p}(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:n})$ shall collapse onto a a few or single unique particle(s) as the machine would have resampled many times between k and n. Another major issue with the aforementioned particle methods is that they only sample variables $\{X_t^i\}$ at time t but the path values $\{X_{0:t-1}^i\}$ remain fixed.

A simple improvement of such techniques would incorporate path value modifications over a fixed lag $\{X^i_{t-L+1:t-1}\}$ for $L > 1$ with respect to a new observation y_t. We shall proceed to discuss some techniques to address these issues but firstly we shall take a small digression into a common numerical method.

Markov chain theory is concerned with finding conditions under which there exists an invariant distribution Q and conditions under which the iterations of the transition kernel $K(.,.)$ converges to the invariant distribution (Gilks & Berzuini 2001, Chien & Fu 1967). Markov chain Monte Carlo (MCMC) inverts Markov chain theory: the invariant distribution, corresponding to the target density $\pi(\boldsymbol{x})$, is assumed known but the transition kernel is unknown. MCMC methods draw random samples from a target density $\pi(\boldsymbol{x})$ (Johannes & Polson 2009). We shall be considering the distribution of parameters and variables given observed prices $p(\Theta, X|Y)$. Utilizing the Clifford-Hammersley theorem (Hammersley & Clifford 1968, Besag 1974), which states that a joint distribution can be specified by the complete conditional distributions, we can characterize $p(\Theta|X, Y)$ by $p(X|\Theta, Y)$ and $p(\Theta, X|Y)$. Sequences can be generated which are not necessarily i.i.d. but form Markov chains which, under a number of conditions and metrics, converge to the target distribution $p(\Theta, X|Y)$. The critical point within MCMC is that it is easier to specify the complete conditionals than to directly analyze and characterize the higher-dimensional joint distribution.

2.3.1 Markov Chain Monte Carlo Sampling

There are two different generic MCMC steps: (i) if the complete conditionals can be directly sampled we utilize the Gibbs sampler proposed by Geman, et al. (1984). Given $(\Theta^{(0)}, X^{(0)})$:

$$\text{1. Draw } \Theta^{(1)} \sim p(\Theta|X^{(0)}, Y) \tag{2.21}$$

$$\text{2. Draw } X^{(1)} \sim p(X|\Theta^{(1)}, Y) \tag{2.22}$$

Continue to generate the sequences of variables $\{\Theta^{(g)}, X^{(g)}\}_{g=1}^{G}$ which converges to $p(\Theta, X|Y)$.

(ii) In practice, one or more of the complete conditionals cannot be sampled and thus we cannot employ Gibbs sampling. In such situations we use Metropolis-Hastings algorithms. These algorithms draw a candidate from a proposal density and accept or reject based on acceptance criteria. Consider the case of the one parameter, single dimensional distribution $\pi(\Theta)$. To generate samples from $\pi(\Theta)$ the Metropolis-Hastings algorithm requires us to specify a proposal density $q(\Theta^{(g+1)}|\Theta^{(g)})$. We require the density ratio $\pi(\Theta^{(g+1)})/\pi(\Theta^{(g)})$ to be readily

and easily computable (Metropolis, et al. 1953). Given $(\Theta^{(0)}, X^{(0)})$ from a prior density:

$$1.\ \text{Draw } \Theta^{(g+1)} \sim q(\Theta^{(g+1)}|\Theta^{(g)}) \tag{2.23}$$

$$2.\ \text{Accept } \Theta^{(g+1)} \text{ with probability } \alpha(\Theta^{(g)}, \Theta^{(g+1)}) \tag{2.24}$$

where

$$\alpha(\Theta^{(g)}, \Theta^{(g+1)}) = \min\left(\frac{\pi(\Theta^{(g+1)})/q(\Theta^{(g+1)}|\Theta^{(g)})}{\pi(\Theta^{(g)})/q(\Theta^{(g)}|\Theta^{(g+1)})}, 1\right) \tag{2.25}$$

We effectively draw from the proposal, draw a uniform random variable and evaluate the acceptance criteria. The algorithm generates the samples $\{\Theta^{(g)}\}_{g=1}^{G}$ whose limiting distribution is $\pi(\Theta)$. Though theoretically there is no restriction on the proposal density, the choice of proposal distribution has a great affect on the algorithm's performance and convergence.

2.3.2 Markov Chain Monte Carlo Particle Filter

Having introduced MCMC methods we can proceed to define the MCMC particle filter. This filter and its two variants we shall be discussing address the limitations we discussed above: namely weight degeneracy and inadequate path value approximation (sample impoverishment). MCMC methods are used in particle filtering in either the sampling or resampling step to draw from the invariant distribution. Some examples of MCMC integration with particle methods can be found in Berzuini, et al. (1997), Liu & Chen (1998), Fearnhead (2004), MacEachern, et al. (1999), Pitt & Shephard (1999) and Fearnhead & Clifford (2003).

Resample-Move Algorithm

The weight degeneracy problem can be alleviated using the resample-move (RM) algorithm which is a special MCMC particle filter (Gilks & Wild 1992, Berzuini & Gilks 2001, Gilks & Berzuini 2001). MCMC uses Markov kernels to generate correlated samples however RM uses them to "jitter" particle locations and reduce degeneracy (Doucet & Johansen 2008). The algorithm is as follows (Gilks & Wild 1992): group particles into a set $S_t = \{\boldsymbol{x}_t^{(i)}\}_{i=1}^{(N_p)}$ at time t and propagate through state space equations using SIR and MCMC sampling; at $t+1$ move the resampled particles according to a Markov chain transition kernel to form S_{t+1}. Forming this new set S_{t+1} involves resampling to draw samples $\{\boldsymbol{x}_t^{(i)}\}$ from S_t such that they are selected with a proportional probability to $\{W(\boldsymbol{x}_t^{(i)})\}$ and then moving the selected particles by sampling from the Markov chain transition kernel (Chen 2003).

Block Sampling

RM suffers from a major limitation: it does reintroduce particle diversity however the importance weights have the same expressions as the generic particle filter (Chen 2003). The importance weights are reliant on the location before the MCMC move whilst the sample depends

upon the location after the move. Even if the transition kernel was perfect, leading to i.i.d. samples from the target distribution, some of those samples would be removed and some would be replicated in the resampling step. This leads to an insignificant reduction in the number of resampling steps compared to the standard particle filter. Doucet, et al. (2006) propose an alternative to RM called block sampling. Recapping RM, it aims to sample $\boldsymbol{x}_t$ in regions of high probability density and then uses MCMC to *rejuvenate* $\boldsymbol{x}_{t-L+1:t}$ after resampling. The optimal importance distribution (which minimizes the variance of the importance weights at t) is:

$$p(\boldsymbol{x}_{t-L+1:t}|\boldsymbol{y}_{t-L+1:t}, \boldsymbol{x}_{t-L}) = \frac{p(\boldsymbol{x}_{t-L:t}, \boldsymbol{y}_{t-L+1:t})}{p(\boldsymbol{y}_{t-L+1:t}|\boldsymbol{x}_{t-L})} \tag{2.26}$$

where

$$p(\boldsymbol{y}_{t-L+1:t}|\boldsymbol{x}_{t-L}) = \int \prod_{k=t-L+1}^{k} f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot g(\boldsymbol{y}_k|\boldsymbol{x}_k) d\boldsymbol{x}_{t-L+1:t} \tag{2.27}$$

As before, it is typically impossible to sample from (2.26) or to compute (2.27). We design an importance distribution which approximates the optimal importance distribution:

$$q(\boldsymbol{x}_{t-L+1:t}|\boldsymbol{y}_{t-L+1:t}, \boldsymbol{x}_{t-L}) \sim p(\boldsymbol{x}_{t-L+1:t}|\boldsymbol{y}_{t-L+1:t}, \boldsymbol{x}_{t-L}) \tag{2.28}$$

Using the optimal IS distribution, that is:

$$q(\boldsymbol{x}_{t-L+1:t}|\boldsymbol{y}_{t-L+1:t}, \boldsymbol{x}_{t-L}) = p(\boldsymbol{x}_{t-L+1:t}|\boldsymbol{y}_{t-L+1:t}, \boldsymbol{x}_{t-L}) \tag{2.29}$$

we obtain:

$$\begin{aligned} w_t(\bar{\boldsymbol{x}}_{0:t-1}, \boldsymbol{x}_{t-L+1:t}) &= \frac{p(\bar{\boldsymbol{x}}_{0:t-L}, \boldsymbol{x}_{t-L+1:t}, \boldsymbol{y}_{0:t}) p(\bar{\boldsymbol{x}}_{t-L+1:t-1}|\boldsymbol{y}_{t-L+1:t-1}, \bar{\boldsymbol{x}}_{t-L})}{p(\bar{\boldsymbol{x}}_{0:t-1}, \boldsymbol{y}_{0:t-1}) p(\boldsymbol{x}_{t-L+1:t}|\boldsymbol{y}_{t-L+1:t}, \bar{\boldsymbol{x}}_{t-L})} \\ &= p(\boldsymbol{y}_t|\boldsymbol{y}_{t-L+1:t}, \bar{\boldsymbol{x}}_{t-L}) \end{aligned} \tag{2.30}$$

where $\{\frac{1}{N}, \bar{\boldsymbol{X}}_t^i\}$ is the set of equally-weighted resampled particles. The optimal weight (2.30) has a variance which decreases exponentially fast with L. As a result, in adaptive sampling this strategy offers vastly superior performance and a significant reduction in the number of resampling steps.

2.4 Computational Intelligence for Optimization

In our introductory discussion on computational models of posterior distributions in Section 2.2 we emphasize the lack of exact methods, parametric techniques and solutions to many empirical problems. Metaheuristics are robust optimization techniques, particularly suited to non-linear continuous optimization problems, which typically synergize ideas drawn from nature and artificial intelligence. The two main, common concepts for metaheuristics are representation (encoding) of a solution and the definition of the objective function. They allow

us to tackle large-optimization problems with satisfactory results, reaching a balance between exploration of the search space (diversification) and exploitation of the best solutions (intensification) (Talbi 2009). Perceived in this light we can say sequential Monte Carlos are themselves single-solution based metaheuristics, manipulating and transforming a single-solution through the course of search.

Parameter tuning is a particularly important task in metaheuristics, dictating the efficiency and effectiveness of search. Carefully tuned parameters enable larger flexibility and robustness. This task can be achieved either offline or online, the difference between the two being metaheuristic execution with predefined parameters or with dynamically updated parameters. Similarly, parameter tuning is an important task in both sequential Monte Carlos and MCMCs. There have been a number of extensions of traditional sequential Monte Carlos imbued with population-based search characteristics to better guide exploration of the search space which we shall now examine.

2.4.1 Evolutionary & Genetic Algorithm Particle Filters

Kwok, et al. (2005) attempt to address the sample impoverishment problem through studies on the number of particles actually being used. They find through the use of the Chebyshev inequality (in that all numbers are close to the mean and can thus be used to bound estimation error) that the number of particles is the cause of impoverishment. They conduct studies of the gambler's ruin problem to try and understand impoverishment from resampling however incorrectly assume naive uniform resampling. In practice the stochastic effect of resampling will accelerate impoverishment.

They propose a hybridization scheme to address impoverishment which replaces traditional resampling with an evolutionary stochastic universal sampling step. This step is in essence identical to control variates from traditional Monte Carlo theory. They manipulate a pair of chromosomes to ensure one of them is lower than some threshold γ. The distance between the two is calculated with the manipulated chromosome repelled from the one with the higher fitness. This method populates unexplored areas of the solution space in aid of minimizing and preventing impoverishment. The proposed approach is tested on monobot location tracking with some stationary landmarks. The hybrid filter is tested against SIR and is found both with low and high numbers of particles to outperform SIR in tracking the bot. The higher number of particles shows a dramatic reduction in sample impoverishment with the solution space diffusing from single strands into a generalized trace envelope. The results are promising however could be furthered with additional scenarios and a detailed comparison with other filters.

Sample impoverishment is also the problem of concern for Han, et al. (2011). They address sample impoverishment through embedding an immune genetic algorithm (IGA) in front of the resampling step. IGAs leverage the idea of antibodies from biology. The antibody self-regulates in the promotion or inhibition of new antibodies, produced through crossover and mutation, thus controlling diversity. IGA effectively diffuses the particle set prior to resampling ensuring that new particles improve diversity of the particle set. Their evolutionary particle filter is used to track simulations of the univariate non-stationary growth model (UNGM):

$$x_t = 0.5x_{t-1} + \frac{25x_{t-1}}{1 + x_{t-1}^2} + 8cos[1.2(t-1)] + w_t \tag{2.31}$$

$$y_t = \frac{x_t^2}{20} + v_t \tag{2.32}$$

where w_t and v_t are both zero mean, unit variance Gaussian white noise processes. In their simulations they find that the evolutionary particle filter outperforms SIR with the IGA step increasing the number of meaningful particles surviving resampling. Applied to target tracking in video surveillance, they again find the evolutionary particle filter outperforming SIR. Further analysis shows that the evolutionary particle filter has a higher count of meaningful particles which enables the filter to better express the true state.

Uosaki, et al. (2005) combine evolutionary computation with non-linear estimation using particle systems. They recognize the similarities between sequential Monte Carlo methods and evolutionary strategies synthesizing the two to propose the evolution strategies based particle filter which replaces the resampling step in SIR with a deterministic selection process step from evolutionary strategies. Running some sample experiments they find that though their filter is similar in performance in mean square errors (MSE) and processing times as SIR it is more stable. Their investigation provides insights into the use of evolutionary computation in particle systems though they do not themselves provide a significantly improved filter.

Duan & Cai (2008) build an evolutionary particle filter for robust simultaneous localization and map building (SLAM) tasks of autonomous mobile robots. They augment the current state space dynamics to include a faulty robot i.e. sensor damage, sensor occlusion or wheels blocked by obstacles. Traditional SLAM techniques have found the traditional PF to be fairly accurate but the assumptions underlying traditional models are fairly loose and do not translate well into real-world environments. Extensions try and address these shortcomings. Most of these come across the same sampling impoverishment problem as described in Section 2.2. To diversify the particle set they use an adaptive mutation scheme straight after resampling. Unnormalized particles are mutated through simple Gaussian mutation with calibrated linear

mutation rates. Tested on a real robot they find that their evolutionary particle filter addresses sample impoverishment. The scheme essentially minimizes selective pressure.

A similar hybrid particle filter is proposed by Li & Honglei (2011). They attempt to resolve a number of problems: sample impoverishment and weight degeneracy for sequential Monte Carlos; and slow convergence speed and premature convergence for genetic algorithms (GA). They propose a unified framework to leverage ability from and across both GAs and PFs. Premature convergence is synonymous to sample impoverishment wherein both systems are failing due to a lack of diversity. They propose to parallelize the problem and to include the unscented Kalman Filter (UKF) as the importance distribution feeding into the PFs. Their parallel genetic unscented particle filter (PGUPF) thus aims to mitigate premature convergence, increase particle diversity whilst at the same time, increasing computational efficiency. They parallelize out across the number of particles N with resampling being carried out in a central unit. Upon importance sampling, the particles are optimized using arithmetic crossover and uniform mutation. The central unit then combines results from across the N streams to build an estimate.

The PGUPF was tested on the bearings-only tracking problem where satellite navigation and positioning of a dynamic vehicle is considered. Root mean square error (RMSE) values were compared against the posterior Cramer-Rao lower bound (CRLB) - if MSE is closer to CRLB then performance is better - across a number of algorithms and was found to be optimal compared to SIR, the sequential importance evolutionary particle filter (SIE-PF) of Uosaki et al. (2005) described above and the unscented particle filter (UPF) (Van Der Merwe, et al. 2001). When the vehicle is in uniform linear motion all filters track well however as soon as the vehicle starts to change behavior tracking performance starts to vary across the filters. When the vehicle is in uniform motion but spiraling SIR loses track. SIE-PF has improved positioning accuracy however with large tracking error. UPF and PGUPF have far better position accuracy with the PGUPF having smaller tracking error. Similar to Uosaki et al. (2005) above, Li & Honglei (2011) find sampling optimization using simple genetic operators such as crossover and mutation increases sample diversity. Comparing computational expense, the PGUPF has increased expense compared to SIR however has vastly superior performance compared to both SIE-PF and UPF. This result coupled with tracking accuracy highlights the benefits of a unified approach. Their tests were done both on single and multi-processor machines and where, as expected, found to provide and utilize added resource capacity compared to sequential execution algorithms.

Another genetic particle filter is proposed by Park, et al. (2007) who seek to advance Uosaki et al.'s (2005) work on the application of evolutionary algorithms to sequential Monte Carlos. Concurrent with our discussion above, Park et al. note that though they use computational intelligence techniques, Uosaki et al. do not propose a significantly improved filter. They aim to make fuller use of the GA toolkit addressing sample impoverishment but to also address jumps in series. The key step here is to vary the number of particles coupled with arithmetic crossover and residual mutation. The genetic operations are adaptively applied using calculations of ESS to address sample impoverishment. Varying the number of particles allows the filter to control computational load. The proposed genetic (algorithm) particle filter (GAPF) is applied to a toy tracking problem in two settings. In the first, the target is assumed to be moving at some constant speed. Here the errors in the distance, velocity and accuracy between SIR and GAPF are invariant. However, in the second case, where we have a maneuvering target SIR cannot track the real state. The GAPF captures the dynamics well. Further analysis of particle evolution shows the mutation operation to be capturing the jumps in the series, observed as particle deprivation, as the target becomes erratic.

2.4.2 Particle Swarm Optimization Particle Filters

Metaheuristics which leverage the collective behavior of species are known as swarm intelligence algorithms (Bonabeau, et al. 1999, Pinto, et al. 2005, Runkler 2008). Swarm intelligence algorithms are inspired from a simplified social model of competition for food. The main characteristics of swarm algorithms are that particles are simple agents, cooperating by indirect communication (self-organization using indirect cooperation is an important issue leveraged from biological systems (Camazine, et al. 2003)) and move in a decision space (Talbi 2009). Particle swarm optimization (PSO) is a particular swarm intelligence algorithm where potential solutions are represented as particles in a search space, and a fitness function is defined as the latent kinematic model (Kennedy & Eberhart 1995). The social metaphor underlying PSO is defined as individuals of a society hold an opinion that is part of the belief space which is shared by neighboring individuals. Individuals can modify the opinion state through an inertia, cognitive and social factor. Inertia refers to knowledge of the environment, the cognitive aspect reflects the individual's previous history of states, with the history of the individual's neighborhood forming the social factor. Driven by well-defined rules of interaction, individuals adopt the belief of more successful individuals in the society which, evolving over time, leads to a culture where individuals hold closely-related opinions (Zheng & Meng 2008).

Zheng & Meng (2008) imbue the generic PF with PSO-with-mutation with the aim of providing a robust, flexible solution to the sample impoverishment problem. The PSO with

mutation operator, proposed by Wang, et al. (2006), addresses the impoverishment problem by applying mutation to keep multiple modes of particle sets. The PSO algorithm is applied to the PF as a dynamic sampling and evolving algorithm, where instead of drawing samples by the importance functions, particles keep in motion after initialization and adjust their location according to PSO rules. Given the dynamic movement of particles, PSO is able to find solutions with a smaller number of particles than traditional particle methods however this is offset by the associated computational effort. The proposed PSO-PF is applied to visual object tracking in both indoor and outdoor settings. The PSO-PF is found to be adaptive and robust to both jitter and occlusion. Further analysis shows that the PSO-PF provides a flexible and adaptive mechanism to keep the society tight (intensification) whilst enabling individuals to explore new areas (diversification).

A similar extension of the APF with PSO is proposed by Yang, et al. (2010) for dual estimation (coupling state and parameter estimation). The aim here is to tune static parameters of dynamic models through a recursive maximum-likelihood estimator. The PSO-based fitness evaluation, similar to Zheng & Meng (2008) above, replaces the importance sampling step. The PSO-APF is tested on simulations of the Markov-switching stochastic volatility model (MSSV) from finance. Given a price series y_t, the MSSV model is given by:

$$y_t = \exp(x_t/2)w_t \tag{2.33}$$

$$x_t = \alpha_{s_t} + \beta_t x_{t-1} + \sigma v_t \tag{2.34}$$

where the initial state of time-varying log-volatility is $x_0 \sim \mathcal{N}(0, \frac{\delta^2}{1-\beta^2})$, and where v and w are uncorrelated Gaussian white noise sequences $\mathcal{N}(0,1)$. The regime variables s_t follows a first-order Markov process $p_{ij} = Pr(s_t = j|s_{t-1} = i)$ for $i, j = 1, ..., k$.

Averaged over 50 runs, the PSO-APF was able to correctly estimate the state and regime variables. In addition, online parameter estimation was able to correctly understand regimes and abrupt changes with a minimal 3.2% misclassification. Yang et al. clearly demonstrate the adaptive nature of their algorithm to changing latent dynamics however fail to demarcate between factors endogenous to the PSO-APF and those to the (A)PF. In addition, their assumption of running on static parameters throughout is a good building block but their assertion that this reflects practical applications is incorrect (Cont 2001). They have provided a base application example though this needs extrapolation to more realistic applications tested and evaluated on real data.

2.4.3 Hybrid & Metaheuristic Particle Filters

Smith & Hussain (2012) hybridize traditional particle filters and apply them to stochastic volatility estimation. Their hybridization involves evolution strategies and real-coded genetic algorithms. Evolution strategies are stochastic, derivative-free numerical optimization methods of non-linear or non-convex continuous optimization problems where new candidates are sampled according to a multivariate normal distribution with pairwise dependencies between variables represented in a covariance matrix. This matrix is updated using covariance matrix adaptation equating to learning the second-order model of the underlying objective function. Real-coded genetic algorithms do not use a coding scheme to code the population, rather they apply evolutionary operators (crossover, mutation, etc.) directly on the population. Applied to the stochastic volatility problem their hybrid metaheuristic real-coded genetic algorithm particle filter (RGAPF) outperforms the traditional PF and the more advanced particle learning algorithm particle filter (PLA). PLA is essentially the SIR-PF though with an MCMC step after every 50 time steps. The RGAPF provides more accurate results with less particles however is sensitive to choice of recombination operator.

Zhang, et al. (2011) address sample impoverishment through the hybridization of the SIR algorithm to include elements from both GA and PSO. Genetic operators are used to maintain particle diversity whilst PSO is used to optimize the particle distribution. The whole algorithm is then parallelized to reduce compute time. To enable parallel execution, the filtering task is divided into two major parts. One group uses elitism to reserve the N_{best} chromosomes into the next generation followed by arithmetic crossover on the remaining chromosomes to maintain diversity. The second group employs PSO to optimize resampling, updating the position and velocity of each particle towards the region of higher likelihood. Thereafter these two groups exchange information to enable the best $N_{migrate}$ particles in each group to migrate to the other group to replace poorly performing particles. This migratory step helps ensure diversity whilst minimizing the probability of premature convergence. The two groups are then integrated to estimate the state.

Applied to the UNGM and averaged across 50 independent simulation runs the proposed genetic algorithm particle swarm optimization particle filter (GAPSO-PF) shows better performance than the generic PF, GA-PF and the PSO-PF. Additionally, the computation time of the GAPSO-PF is lower compared to the other hybrid particle filters. Closer examination of tick-by-tick performance shows the GAPSO-PF adapting to abrupt changes quicker than the other filters, which is self-evidently pronounce in the problem at hand though it's performance through the race to convergence is comparable to the other filters. The RMSE results reported

are not statistically significantly better however the algorithm shows promise in this quite extreme experimental case.

Klamargias, et al. (2008) also test their PSO-PF on the UNGM. Their proposed PSO-PF attempts to alleviate sample impoverishment by addressing the problem of biasing samples significantly towards either the prior or the likelihood. This problem is tackled using conventional multi-objective optimization where the prior and posterior distributions are aggregated into a single objective function. Perturbing the prior sample to maximize the objective function assumes a balance between the prior and the posterior likelihood. The corresponding objective function is maximized using PSO the aim of which is to move particles towards regions of higher likelihood in the search space without allowing them to move significantly far away from the region of the prior.

In their experiments Klamargias et al. (2008) compare the efficacy of the PSO-PF and the PF with two varying particle set sizes $N = 20$ and $N = 50$. These initial samples were passed to the PSO which iterated 50 times to return the best positions as the final sample. The overall filtering was simulated 100 times over 60 time steps with the RMSE being compared using the Wilcoxon signed-rank sum test (used to compute the statistical significance between a given PSO-PF and PF run). For $N = 20$ particles the PSO-PF outperformed the PF across the three levels of noise - minimal, moderate and high. This result was evident in the $N = 50$ particles experiment however as expected, given the larger particle set and when there was both higher system and measurement noise, there was no statistically significant increase in performance between the PSO-PF and PF indicating that the PSO-PF provides robust solutions in high-noise systems with a smaller particle set.

Pantrigo & Sanchez (2005) hybridize particle filters with population based metaheuristics (PBM) in aid of solving dynamic optimization problems exploiting advantages of both approaches. As discussed above, hybrid metaheuristics are the skilled synergy of advantageous elements to find higher quality solutions. A hybridization framework is proposed to test different combinations of PFs and PBMs. The template aims to build a low-level hybridization between the two techniques to improve the quality of the solution. The template includes the two major strands of the chosen PF and PBM followed by a selection, replacement and estimation procedure. Various implementations of the template were applied to the articulated and multiple object tracking problem. Similar to Klamargias et al. (2008), Pantrigo & Sanchez find that the traditional PF performance increased when the number of particles increased and conversely the PBM-PF hybridizations produced robust results with a limited particle set. They have shown you can plug and play PBM into PF to solve dynamic optimization tasks.

Building on their work in Pantrigo & Sanchez (2005) and Pantrigo, et al. (2011), Cabido, et al. (2012) show an effective approach to visual tracking on a graphical processing unit using a hybridized particle filter. They propose a hybridization of the particle filter with a memetic algorithm to form the memetic algorithm particle filter (MAPF) for tracking single and multiple objects. Memetic algorithms are popular genetic local search algorithms for multi-objective optimization (Gen & Lin 2004, Ishibuchi & Murata 1998, Jaszkiewicz 2002, Talbi, et al. 2001). The key principle in memetic algorithms is the combination and hybridization of different heuristics in a population of solutions during the overall optimization process. This consists of imbuing an evolutionary algorithm search with a local search algorithm from which solutions are found through potentially local improvement procedures applied in different stages of the process.

The memetic algorithmic step replaces IS in the PF to form the MAPF. The memetic algorithm applies recombination, mutation to maintain diversity and an improvement step in order to sample solutions. The PF was initialized with $N = 256$ particles and the MAPF with only $N = 64$ particles. Compared on tracking a single synthetic sequence through the diagonal, the MAPF shows perfect tracking compared to minor vibrations in the PF's tracking of this perfect motion. Similar results are observed when tracking multiple objects in the same space however there is greater computational load in tracking with the PF. Further tested on tracking two squash players, these results are reinforced showing the accuracy and precision obtained with the MAPF in real conditions. In the MAPF, the PF is a sequential estimator with an imbued memetic algorithm refining the population to improve estimation. This is primarily attributable to the memetic algorithm's local search which explores the search space around the PF estimation more exhaustively. As noted, real applications of the proposed procedure would add steps to handle issues like occlusion, varying light and heavy noise.

Wang & Li (2010) propose a co-evolutionary particle filter to address sample impoverishment and to maintain diversity in PFs. Each particle is an intelligent agent having the ability of local perception, competitive selection and self-learning. These evolutionary behaviors enable these intelligent particles, constructed as a lattice, to swarm towards regions of higher likelihood. The proposed multi-agent co-evolutionary particle filter (MACoEPF) is tested on the moving object tracking problem. MACoEPF performance is compared to the generic PF. On face target tracking in a video sequence, PF tracking deteriorates as both time passes and with face moves: a direct result of sample impoverishment and a lack of diversity amongst the particles. The MACoEPF addresses these drawbacks and maintains tracking accuracy throughout the sequence. Their second experiment involves tracking a fast moving object in a complex scene by the MACoEPF. In such scenarios the PF rapidly deteriorates and as such is not included

for comparison. The MACoEPF tracks the fast moving target in real-time and accurately in presence of a number of similar objects in the sequence. Wang & Li (2010) have demonstrated the synergy of swarm intelligence with traditional PFs to address sample impoverishment and sample diversity.

Lima & Krohling (2011) also introduce a hybridization mechanism into the general PF framework. They insert a sampling mechanism inspired by differential evolution (DE) and PSO into the PF. DE is a relatively new though highly successful evolutionary algorithmic approach to continuous optimization (Storn & Price 1997). The main idea here is to use vector distances between samples to perturb the population. From an initial random population successive generations can be generated by adding to a random sample the weighted difference of two other random samples. DE uses the crossover operation to generate these new solutions which in turn increases the diversity of the population.

The proposed hybrid particle filters replace subsequent IS runs following initialization of SIR with a routine which finds fitter particles using either DE or PSO. Applied to the object tracking problem the PF, PSO-PF and DE-PF were used for histogram intersection detection (an effective similarity measure between images) for tracking in a 100 frame image sequence. The first version of the 100 frame sequence contained the original frames, whilst the second had noise added. Running with $N = 150$ particles the filters were run against the two sequence sets. On the original sequence performance was comparable towards the beginning of the sequence however as the frames progressed the PF began to deteriorate rapidly though it did not lose complete track. On the second sequence set, the PF quickly deteriorated losing track as samples degenerated, as measured through the ESS. Assessing tracking error the DE-PF performed the best out of the three filters whilst having both the lowest degeneration average and lowest average error. Assessing the influence of increasing the number of particles, the same experiments were run with $N = 300$ and $N = 600$ particles. This investigation showed that only the PF was influenced by increasing the number of samples reiterating results we have seen in the literature above.

Yoo, et al. (2012) address sample impoverishment and propose a novel evolutionary particle filter for sequential dependency learning from video data. The proposed scheme for dependency learning and segment summarization does not rely on prior labeling, using a Gaussian mixture model (GMM) for modeling the dependency structure complemented with the scalar invariant feature transform to estimate changepoints (switching from one scene to another) to enable segment summarization. Hybridizing the PF, particles are sampled base on a Euclidean distance fitness function with crossover. Mutation operations are performed to maintain di-

versity in the population and to avoid degeneracy. The fitness function uses ridge regression techniques to ensure a normalized measure of pertinence for each particle.

The evolutionary particle filter for stream segmentation and GMM for sequential dependency learning were tested on three broadcast television episodes compared and contrasted to 10 human participants. As expected the humans did not agree on changepoints however a characteristic set was built for each episode to aid comparison. The proposed method identified four times as many changepoints which enabled it to, through the transition probability matrix, correctly estimate segment sequences and thus represent a video stream in a much smaller and simpler data structure. Their results are promising however leave room for improvement in the number of changepoints identified by the learning process.

Xiaowei, et al. (2013) also address sample impoverishment with an evolutionary particle filter for improved object tracking. A self-adaptive genetic algorithm evolutionary mechanism (S-AGA) is added to the resampling step in SIR to make particles into intelligent agents with the ability of dynamic self-adaption. Self-adaptive crossover and mutation operators enable the evolution system to propose solutions which track the true state more accurately than conventional methods. S-AGA leverages the maximum fitness in the particle set, the largest fitness of the crossover individuals, the average fitness of the population and the fitness of the largest mutation individual to provide an adaptive evolution strategy where the crossover and mutation operators are dynamically computed to the evolutionary strategy and fitness changes.

The self-adaptive genetic algorithm particle filter (S-AGAPF) was tested against varying illumination, interference from similar targets and occlusion, and shape change. For face tracking with illumination variation the S-AGAPF outperforms the PF. Using only color information the PF fails to track as the target color changes dramatically as illumination is varied however, as the S-AGAPF is using texture cues from variations in illumination it can still track very well. Interference by a similar target and occlusion for face tracking highlight problems of particle degeneration and sample impoverishment in the traditional PF. With intelligent adaptive agents, the S-AGAPF maintains track through interference by similar targets and occlusion. Similar results are observed when testing against structural deformation to validate tracking performance. Building the posterior based on subtleties of the latent signal enables the S-AGAPF to outperform traditional methods for object tracking.

2.5 Discussion

Understanding complex non-linear non-Gaussian dynamics where exact inference is not possible leads us to use sequential Monte Carlo approximate inference methods. The underlying

temporal distribution and the associated generating pdf are understood through population based recursive Bayesian estimation techniques. Latent signal processing in such scenarios is particularly challenging as the underlying distributions are not i.i.d with respect to the observed series showing excess kurtosis and non-stationarity. In addition many complex systems are conditionally heteroskedastic which inherently leads to periods of turbulence followed by periods of calm.

As such linear techniques such as the KF have proven insightful but ultimately not useful for complex time series analysis. An extension, the extended Kalman filter, was proposed to handle such non-linearities and though an improvement on the simple KF performs poorly under non-Gaussianity and performs extremely poorly under severe non-linearity. The unscented Kalman filter is a variant of the KF which addresses such problems using the deterministic sampling unscented transform technique to provide increased mean and covariance estimation accuracy.

Extensions of the KF address some of the underlying problems however are restricted by linear Gaussian dynamics. Such dynamics are not prevalent in distributions of concern, motivating us to look into techniques which address non-linearity and non-Gaussianity. We are unable to obtain closed-form solutions to distribution approximations and thus apply numerical methods. The PF is a common sequential Monte Carlo method for optimal estimation in non-linear non-Gaussian settings. The main advantage of particle methods is their lack of reliance on any local linearization or crude functional approximation.

We have inspected IS which is the building block of particle methods. IS allows us to sample from complex highly-dimensional distributions but with linear increases in complexity upon each draw. To admit fixed computational complexity we use SIS though this suffers from the weight degeneracy problem which can be alleviated by resampling which conversely introduces sample impoverishment. The SIR algorithm replaces particles with high weight with many particles with high inter-particle correlation. The SIR algorithm coupled with adaptive resampling is the simple sequential PF we have defined. This algorithm is very sensitive to choice of proposal distribution.

The SIR filter suffers from two critical drawbacks; it does not approximate outliers or the tails of a distribution well. Not handling leptokurtosis is particularly debilitating in complex systems. The APF reduces the effects of outliers whilst handling fat-tailed distributions, taking advantage of *ex-ante* information. If the process noise is particularly large, the APF can fail to focus on pertinent particles and performs similarly to the sequential PF.

The GPF which filters dynamic state space models with Gaussian noise and non-linear functions in the state and measurement equations has been shown to deliver increased mean and variance estimation accuracy than other PFs, even when faced with severe non-linearities and noise. Unfortunately it is sensitive to outliers as the state noise is assumed to be Gaussian. Modeling the state noise as another mixture density, the GSPF overcomes this issue exhibiting greater performance than the GPF in the presence of outliers and leptokurtosis.

MCMC methods have been used to address a number of issues, specifically particle degeneracy and inadequate path value approximation, with sequential Monte Carlo algorithms. There are two MCMC sampling techniques. If we can state the complete conditional distributions describing the joint distribution we use the Gibbs sampler. In practice, one or more of the conditionals is unknown and we employ the Metropolis-Hastings algorithm. RM addresses particle degeneracy by jittering particle locations thus reducing degeneracy. This algorithm does introduce particle diversity but is essentially the same as the sequential PF with, typically, an insignificant reduction in the number of resampling steps. Block sampling is an alternative to RM which uses the optimal IS distribution when designing an importance density which approximates the optimal importance distribution. The resulting weight calculations have a variance which decreases at an exponential rate offering vastly superior performance and a significant reduction in the number of resampling steps.

We have discussed a number of extensions of sequential Monte Carlo methods which leverage techniques from computational intelligence. A number of evolutionary and hybrid particle filters are discussed across a variety of problems and are shown to mitigate sample impoverishment to some degree. In addition, we have discussed a number of evolutionary and genetic strategies which show the efficacy of particle filters imbued with metaheuristics.

Empirically, the stochastic effect of resampling accelerates sample impoverishment. Metaheuristics are used to diversify the particle set. Uosaki et al. (2005), Li & Honglei (2011) and Yoo et al. (2012) amongst others find sampling optimization using simple genetic operators such as crossover and mutation increases sample diversity. Further integration of PBMs into the PF framework has shown encouraging results. Dynamic particles in PSO based PFs are able to find solutions with a smaller number of particles than traditional methods however with an increased computational cost. Klamargias et al. (2008) and Pantrigo & Sanchez (2005) find that traditional PF performance increased when the number of particles increased and conversely PBM-PF hybridizations produce robust results with a limited particle set. Across a number of metaheuristics including evolutionary strategies, evolutionary algorithms, genetic algorithms,

PSO, DE, memetic algorithms, we have seen their synergy with PFs to solve dynamic optimization tasks.

Sequential Monte Carlo methods are an active research area. Given their success they have been used in many settings and various adaptations provide further avenues of research. From our discussion above and reviewed literature we can elicit some open problems within the area:

- The primary problem with particle methods is weight degeneracy where the distribution of the importance weights becomes more and more skewed over time with fewer and fewer non-zero importance weights. In introducing their SIR filter, Gordon et al. (1993) highlight this problem with the preceding and their own methods. ESS based adaptive algorithms have been advocated but the fundamental issue is with resampling itself. It does not prevent the problem of weight degeneracy it essentially discards meaningless particles from calculations. Additionally, resampling introduces the problem of sample impoverishment where particles with high weight w_k^i are statistically selected many times leading to a loss of diversity amongst the particles where the resultant sample will contain many repeated points (Arulampalam et al. 2002). Sample impoverishment is particularly acute for systems with small latent noise as all particles will collapse to a single point very, very quickly (Clapp 2000, Liu & West 1999).

- Being a numerical method, it is inherently impossible to represent a highly-dimensional distribution by a finite sample set despite asymptotic justification. Additionally, when such problems are complex it is not possible to obtain accurate approximations in a reasonable amount of time. Though they are parallelizable, most techniques when applied in practice are compute bound and naively assigning hardware to the problem will not speed up computation.

- Parameter estimation with a large number of parameters leads to slow exploration of the parameter space. In addition dual estimation, coupling state and parameter estimation, is extremely difficult with a limited number of particles. These can be amalgamated into a single higher-dimensional filter however such problems hinder such filter algorithms from working in practice (Doucet & Johansen 2008, Saha 2009). Further research in this area is needed.

- Particle methods are very sensitive to choice of proposal distribution. It has been suggested to use the transition prior however this does not incorporate the most recent observation, an omission which leads to and exacerbates sample impoverishment. Building

improved PFs necessitates understanding the dynamics of the system of concern and choosing a proposal distribution which is conditioned on the latest observation.

- Reliably detecting divergence is essential to running filters optimally. Divergence indicators are relatively sparse and is largely an unexplored field. In fly-by-wire and drive-by-wire systems, measurements are split up into a bank of filters. PF_i provides every n-th sample, for $i = 1, 2, ..., n$ (Gustafsson, et al. 2002). These PFs are approximately independent with voting used to restart each filter. This process has been effective and efficient in removing outliers from the data.

- Convergence analysis and asymptotic behavior in the tails, is sparsely treated in complex systems. Estimation accuracy, choice of proposal distribution, assumptions and various other inputs need to be understood in the convergence of the filters. Effort is needed in this area.

2.6 Summary

Stochastic filtering and optimal estimation of non-linear and non-Gaussian distributions is a particularly pertinent problem in many practical applications. Analyzing the underlying processes from such times series given noisy observations, we are unable to find closed-form solutions and use particle filters, a power numerical method, which work by partitioning the state space and populating it with particles with respect to some probability measure. The higher this measure, or weight, the denser the particle concentration. The state space evolves temporally and the particle system evolves around this, promoting particles with higher weights and discarding less pertinent particles adaptively. Particle methods have found prominence in many applications providing a robust, numerical method to tackle a number of online and stochastic inference problems, an active research area of which we have highlighted a number of open problems.

Chapter 3

Theoretical Model

There are a number of open problems in sequential Monte Carlos which have been highlighted through our exploration of the literature and field. Primal of these, we shall be focusing our research on the weight degeneracy and sample impoverishment problem and propose a new particle filter which yields increased accuracy for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems compared to contemporary filters. In Section 3.1 we briefly describe the principles and concepts behind our proposed method. In Section 3.2 we move onto a formal specification of the model and provide the mathematical reasoning in Section 3.3. We proceed to show some experimental results in Section 3.4. Thereafter we briefly summarize.

3.1 Principles & Conceptual Model

In evolutionary computation (EC), the generation gap is the fraction of the population which is replaced each cycle. It is closely linked to the concepts of overlapping and non-overlapping populations which define the competition between parents and offspring where in overlapping generations they compete with one another for survival and in non-overlapping models the entire parent population is replaced. Here, selection algorithms are evolutionary algorithms which makes decisions as to which parents and which offspring should survive (Sarma & De Jong 2000).

Particle filters can be viewed as evolutionary algorithms as they have overlapping generations compared to genetic algorithms which historically, are non-overlapping. Sampling can be seen as mutation, resampling as selection and the importance weight calculation as the fitness function (Uosaki et al. 2005).

Recollecting SIS/R and SIR, resampling retains the most pertinent particles however destroys information by discounting the potential future descriptive ability of particles. This phenomena has been well commented upon in a number of surveys e.g. Chen (2003) and Gustafsson (2010), with ESS based adaptive SIS/R filters as the answer. However we find a number of

techniques which address sample impoverishment better e.g. APF, MCMC-PF, from within the sequential Monte Carlo community. Recently there has been a concerted effort at advancements of computational intelligence sequential Monte Carlos which leverage ideas from traditional evolutionary computation and genetic algorithms to propose novel filters addressing practical and empirical shortcomings of the PF.

As commented above in our discussion of methodologies, resampling does not really prevent sample impoverishment: it simply excludes poor samples from calculations, providing future stability through short-term increases in variance. If we were to leverage the descriptive ability of naively discarded particles in an adaptive evolutionary environment with a well defined fitness function we posit that we shall see an increased accuracy for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems.

3.2 Formal Model

We embed a generation based adaptive particle switching step into the particle filter weight update, using the transition prior as our proposal distribution, to enable us to make use of previously discarded particles ψ if their discriminatory power is higher than the current particle set. This step is formalized into our adaptive path particle filter (APPF) and is detailed in Table 3.1.

The APPF is a recombinatory evolutionary algorithm. In Figure 3.1 we have $N = 20$ particles: at $t = 36$ upon evaluation of the fitness function we select 11 particles from the current importance samples $\hat{\mathbf{x}}_{36}$ (marked as the blue particles) and 9 particles from importance samples based on the previously resampled out set $\check{\mathbf{x}}_{36}$ (marked as the red particles). For example we can see $W_{36}^{(1)}$ is assigned as $\hat{\mathbf{x}}_{36}^{(1)}$, similarly $W_{36}^{(8)}$ is assigned as $\check{\mathbf{x}}_{36}^{(8)}$. As can be seen in the resultant particle set W_{36} we are leveraging the fittest particles amongst and across the two generations.

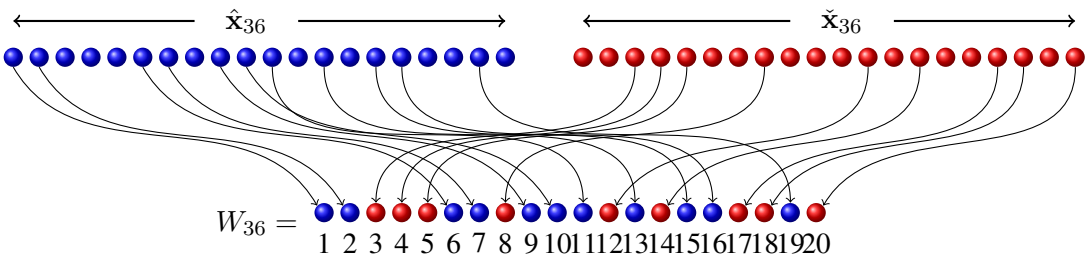

Figure 3.1: APPF weight update fitness assessment through inter-generational competition for $N = 20$ at $t = 36$:= $W_{36}^{(i)} = \max\left[p(\mathbf{y}_{36}|\hat{\mathbf{x}}_{36}^{(i)}), p(\mathbf{y}_{36}|\check{\mathbf{x}}_{36}^{(i)})\right]$

Table 3.1: APPF Algorithm

1. Initialization: for $i = 1, ..., N_p$, sample

$$\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0)$$
$$\boldsymbol{\psi}_0^{(i)} \sim p(\boldsymbol{x}_0)$$

with weights $W_0^{(i)} = \frac{1}{N_p}$.

For $t \geq 1$

2. Importance sampling: for $i = 1, ..., N_p$, draw samples

$$\hat{\boldsymbol{x}}_t^{(i)} \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(i)})$$
set $\hat{\boldsymbol{x}}_{0:t}^{(i)} = \{\boldsymbol{x}_{0:t-1}^{(i)}, \hat{\boldsymbol{x}}_t^{(i)}\}$
and draw $\check{\boldsymbol{x}}_t^{(i)} \sim p(\boldsymbol{x}_t|\boldsymbol{\psi}_{t-1}^{(i)})$
set $\check{\boldsymbol{x}}_{0:t}^{(i)} = \{\boldsymbol{x}_{0:t-1}^{(i)}, \check{\boldsymbol{x}}_t^{(i)}\}$

3. Weight update: assess fitness

$$W_t^{(i)} = \max\left[p(\boldsymbol{y}_t|\hat{\boldsymbol{x}}_t^{(i)}), p(\boldsymbol{y}_t|\check{\boldsymbol{x}}_t^{(i)})\right]$$

evaluate:
if $p(\boldsymbol{y}_t|\check{\boldsymbol{x}}_t^{(i)}) > p(\boldsymbol{y}_t|\hat{\boldsymbol{x}}_t^{(i)})$ **then**
$\hat{\boldsymbol{x}}_t^{(i)} = \boldsymbol{\psi}_t^{(i)}$
end if

4. Normalize weights:

$$\tilde{W}_t^{(i)} = \frac{W_t^{(i)}}{\sum_{j=1}^{N_p} W_t^{(j)}}$$

5. Commit pre-resample set of particles to memory:

$$\{\boldsymbol{\psi}_t^{(i)}\} = \{\hat{\boldsymbol{x}}_t^{(i)}\}$$

6. Resampling: Generate N_p new particles $\boldsymbol{x}_t^{(i)}$ from the set $\{\hat{\boldsymbol{x}}_t^{(i)}\}$ according to the importance weights $\tilde{W}_t^{(i)}$.

7. Repeat from importance sampling step 2.

At the level of the representation of the probability distribution, the primary output of a sequential Monte Carlo method, the APPF is using a fitness-based recombination of a past representation with a current representation. In this sense, we believe the algorithm draws upon the body of EC experience and theory. Admittedly, this is in essence an EC algorithm with a population of size 2. A natural extension, which would make APPF appear to be a more traditional EC algorithm, would be the maintenance and fitness-based recombination of several past iterations particles and thus a larger population. However, note that due to the nature of

the PF process itself - that of the state process being a Markov process by assumption (i) as Equation (2.5) in Section 2.1 - the immediate previous particle set contains information about those previous representations explicitly. In this sense, the recombination of only these two sets of particles in the APPF contains the implicit schema average fitness information upon which much of EC theory is based (Goldberg & Holland 1988). Thus, we feel that the dramatic improvements in performance the APPF offers, as shown in the following experimental results and further validated in this thesis, are directly related to EC theory and practice, and can be built on further using EC ideas.

3.3 Justification

The prediction phase follows the same process as the original bootstrap filter (Gordon et al. 1993). As such it follows that repeating the process of sampling $x_{k-1}(i)$ from $p(x_{k-1}|D_{k-1})$ generates $\{x_k^*(i) : i = 1, \ldots, N\}$ which are independently distributed as $p(x_k|D_{k-1})$.

Similarly the update phase relies upon the fundamental reasoning for Monte Carlo methods and the reasoning from Smith & Gelfand (1992). The main reasoning for Monte Carlo methods in general argues their convergence by a central limit theorem onto an invariant and thus the correct distribution as described in Section 2.2 and as follows (Van Der Merwe et al. 2001). As the posterior can be approximated by the empirical estimate

$$\hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{1}{N}\sum_{1=1}^{N}\delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t})$$

where the random samples $\{\mathbf{x}_{0:t}^{(i)}; i = 1, \ldots, N\}$ are drawn from the posterior distribution and $\delta(d.)$ is the Dirac delta function. Thus, any expectations of the form

$$\mathbb{E}[f(\mathbf{x}_{0:t})] = \int f(\mathbf{x}_{0:t})p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})d\mathbf{x}_{0:t}$$

can be approximated by

$$\overline{\mathbb{E}[f(\mathbf{x}_{0:t})]} = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_{0:t}^{(i)})$$

where the particles $\mathbf{x}_{0:t}^{(i)}$ are i.i.d. According to the law of large numbers we have

$$\overline{\mathbb{E}[f(\mathbf{x}_{0:t})]} \xrightarrow[N\to\infty]{a.s.} \mathbb{E}[f(\mathbf{x}_{0:t})]$$

where $\xrightarrow[N\to\infty]{a.s.}$ is almost surely convergence. Furthermore, if the posterior variance of $f(\mathbf{x}_{o:t})$ is bounded such that $var_{p(.|\mathbf{y}_{1:t})}f(\mathbf{x}_{0:t}) < \infty$ then the following central limit theorem holds

$$\sqrt{N}\left(\overline{\mathbb{E}[f(\mathbf{x}_{0:t})]} - \mathbb{E}[f(\mathbf{x}_{0:t})]\right) \underset{N\to\infty}{\Rightarrow} \mathcal{N}\left(0, var_{p(.|\mathbf{y}_{1:t})}f(\mathbf{x}_{0:t})\right)$$

where $\underset{N\to\infty}{\Rightarrow}$ denotes convergence in distribution.

In Smith & Gelfand (1992) it is shown that Bayes theorem can be implemented as a weighted bootstrap which is what was implemented and justified by Gordon et al. (1993). It must be noted that MCMC, sequential Monte Carlo and the proposed method are all heuristic methods. Consequently the main motivation for the APPF is no different in character to the main motivation for Monte Carlo methods.

3.4 Experimental Results

We present two experiments to illustrate the operation of the APPF. We compare estimation performance with the standard SIR filter and with the MCMC-PF. The first example is a synthetic scalar estimation problem and the second is the standard univariate log-stochastic volatility model estimation problem.

3.4.1 Synthetic Experiment

We explore the scalar estimation problem posed by Van Der Merwe et al. (2001), a common benchmark problem in the sequential Monte Carlo literature (Zhang, et al. 2010). A time series was generated by the process model:

$$x_{t+1} = 1 + \sin(\omega\pi t) + \phi_1 x_t + v_t \tag{3.1}$$

where $v \sim \mathcal{G}a(3,2)$ modeling the process noise, $\omega = 4e{-}2$ and $\phi_1 = 0.5$ are scalar parameters. We have the non-stationary observation model:

$$y_t = \begin{cases} \phi_2 x_t^2 + n_t & t \leq 30 \\ \phi_3 x_t - 2 + n_t & t > 30 \end{cases} \tag{3.2}$$

where $\phi_2 = 0.2$ and $\phi_3 = 0.5$. The observation noise n_t is Gaussian $\mathcal{N}(0, 0.00001)$. This problem is severely non-linear in both the system and measurement equations. This model was realized with an initial uniform prior.

Given noisy observations y_t we used the filters to estimate the latent states x_t for $t = 1...60$. The experiment was repeated 100 times with random re-initialization using $N = 200$ particles and residual resampling. Table 3.2 summarizes the performance of the various filters. It shows the mean and variance of the RMSEs of the state estimates. We observe a marked increase in performance of the APPF RMSE = 0.305 compared to the PF RMSE = 0.427 and MCMC-PF RMSE = 0.444.

Deeper inspection shows the APPF locking onto the latent signal quicker than the PF and MCMC-PF. Figure 3.2 shows results from our synthetic experiment for the 100^{th} run. The observed (realized) series is the blue line, the latent state (true x) is the red line, the PF state

Table 3.2: Synthetic Experiment Simulation Results. RMSE mean, and variance over 100 runs using 200 particles and residual resampling.

SMC	Mean	Var
PF	0.427	0.045083
MCMC-PF	0.444	0.051481
APPF	0.305	0.055635

estimate is the green line, the MCMC-PF state estimate is the yellow line and the APPF state estimate is the magenta line.

We observe around $t = 10$ and $t = 12$ the APPF reacting to the fluctuation in the latent signal quicker than the MCMC-PF which does not adapt until $t = 13$. Similarly between $t = 5$ and $t = 7$ the PF lags the APPF and MCMC-PF. Further issues can be seen around $t = 27$ through $t = 31$ as the non-stationarity of the underlying signal affects all filters with the APPF showing most stability. Such phenomena result in the lower RMSE of the APPF compared to the PF and MCMC-PF, highlighting the adaptability and robustness of track of the APPF.

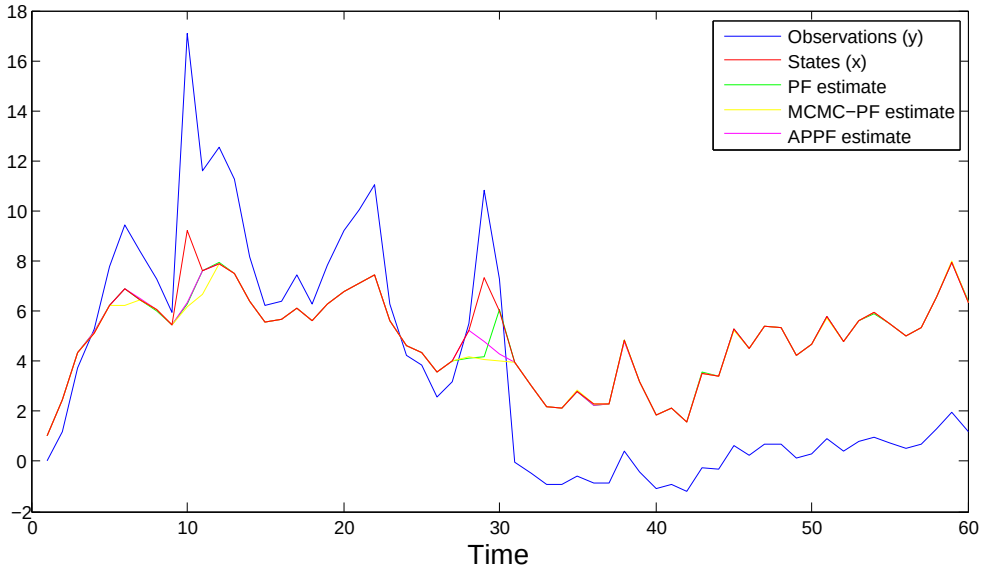

Figure 3.2: Synthetic Experiment Simulation Results Run 100/100 - Filter Estimates (posterior means) vs. True State PF (RMSE=0.58116), MCMC-PF (RMSE=0.66878) and APPF (RMSE=0.5505)

3.4.2 Stochastic Volatility Model Experiment

Consider sequential Monte Carlo implementations which attempt to estimate the standard univariate log-stochastic volatility model (Jacquier, et al. 2004, Ghysels, et al. 1996) where the returns y_t and log volatility states x_t follow a state space model of the form:

$$y_t = \exp(x_t/2)\varepsilon_t \tag{3.3}$$

$$x_t = \alpha + \beta x_{t-1} + \sigma\eta_t \tag{3.4}$$

with initial log volatility $x_0 \sim \mathcal{N}(m_0, C_0)$ and with σ as the volatility of volatility. The errors ε and η are uncorrelated white noise sequences $\mathcal{N}(0,1)$. In addition the observation process is given by:

$$y_t = \log P_t - \log P_{t-1} \tag{3.5}$$

We generate a 500 time-step time series using the parameter values

$$\theta = (\alpha, \beta, \sigma^2) = (-0.0084, 0.98, 0.04)$$

and an initial prior of $x_0 \sim \mathcal{N}(0,1)$ as proposed by Stroud, et al. (2004). We can see our simulated stochastic volatility series in Fig 3.3.

Given noisy observations y_t we estimated the latent state x_t using the sequential Monte Carlos for $t = 1...500$. The experiment was repeated 100 times with random re-initialization using $N = 10,000$ particles and systematic resampling. Table 3.3 summarizes the performance of the various filters. It shows the mean and variance of the RMSE of the state estimates. We observe a notable increase in performance of the APPF RMSE = 0.22688 compared to the PF RMSE = 0.28088 and MCMC-PF RMSE = 0.26521.

We can see a sample visualization from the 100$^{\text{th}}$ run of our recursive stochastic volatility estimation in Figure 3.4. As before, the observed (realized) series is the blue line, the latent state (true x) is the red line, the PF state estimate is the green line, the MCMC-PF state estimate is the yellow line and the APPF state estimate is the magenta line.

This simple realization shows clustering of volatility and the leptokurtic nature of financial time series. The effective advantage of our method can be seen from $t = 0$ through $t = 150$ where the APPF exhibits markedly superior estimation accuracy than both the PF and the MCMC-PF. Thereafter, all methods are fairly similar though this initial race to convergence provides empirical proof of the superiority of our method with the PF RMSE = 0.21441, MCMC-PF RMSE = 0.19744 and APPF RMSE = 0.061379.

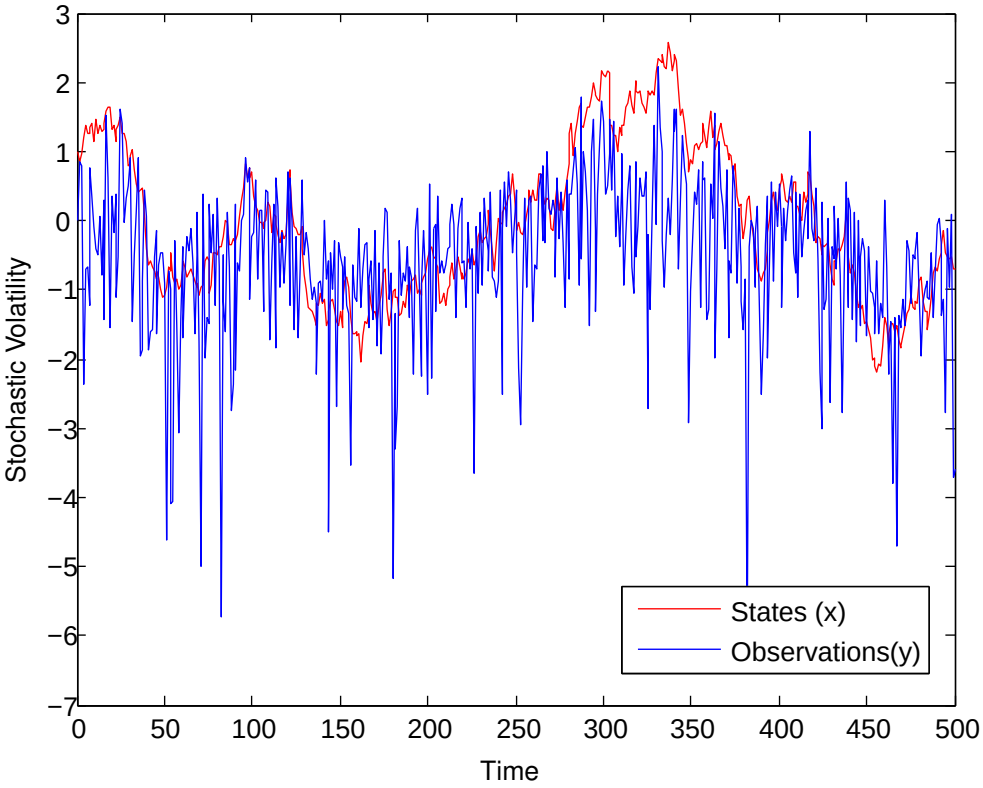

Figure 3.3: Simulated Stochastic Volatility Series

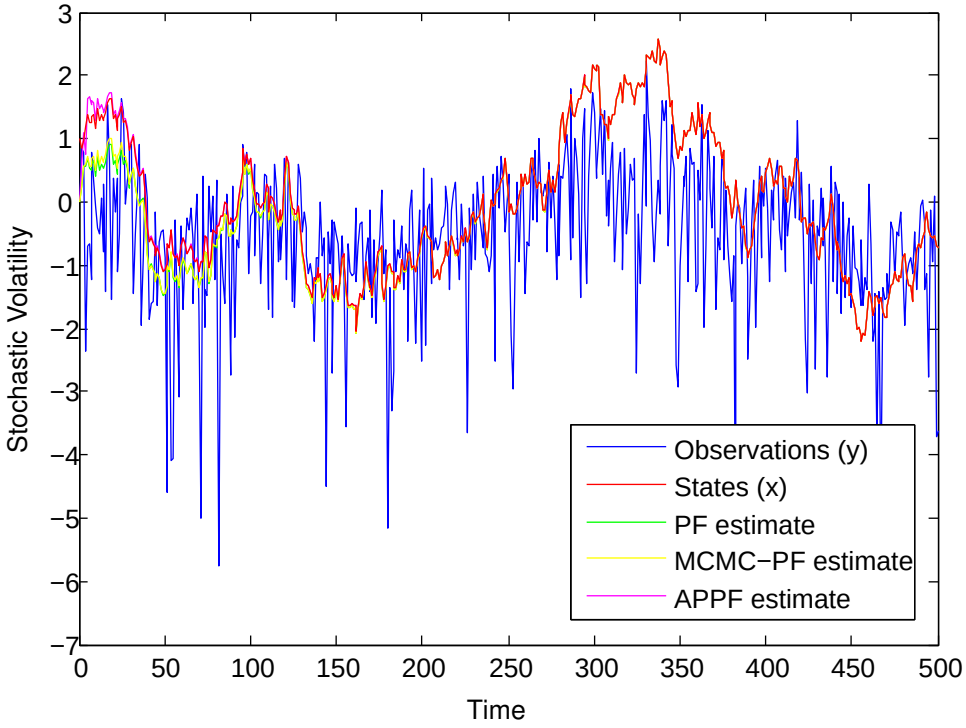

Figure 3.4: Stochastic Volatility Experiment Simulation Results Run 100/100 - Filter Estimates (posterior means) vs. True State

Table 3.3: Stochastic Volatility Simulation Results. RMSE mean, and variance over 100 runs using 10,000 particles and systematic resampling.

SMC	Mean	Var
PF	0.28088	0.007177
MCMC-PF	0.26521	0.009993
APPF	0.22688	0.017392

3.5 Summary

We have introduced our APPF which uses techniques from evolutionary computation to provide adaptive path switching between particle generations. Theoretically we should see an increase in accuracy in recursive Bayesian estimation of non-linear non-Gaussian dynamical systems as we leverage the descriptive ability of discarded particles.

We have provided some experimental results of the application of evolutionary computation selection techniques in sequential Monte Carlo methods both in a synthetic and real-world example. In both experiments we see an increase in recursive Bayesian estimation accuracy with the APPF compared to the PF and MCMC-PF.

Closer inspection of some sample realizations allow us to observe the mechanism through which our filter works and excels compared to contemporary filters. We see that whilst the PF and MCMC-PF are converging their signal onto the latent states the APPF has already converged, providing greater estimation accuracy in the meanwhile. These two preliminary experiments provide some exciting results in validation of our hypothesis on the advantages of adaptive path selection schemes in sequential Monte Carlos.

Chapter 4

Stochastic Volatility Modeling

We begin our evaluation of APPF performance on the stochastic volatility estimation problem from computational and quantitative finance. This problem has practical importance for both securities trading and derivative pricing. We begin in Section 4.1 by defining the problem and introducing the Heston stochastic volatility model. We describe the data we are using and our performance metrics in Section 4.2 and Section 4.3 respectively. We provide our results in Section 4.4, summarizing and concluding in Section 4.5.

4.1 Stochastic Volatility Estimation Problem

Derivative theory is founded on the ideas of delta hedging and no-arbitrage. Delta hedging is the concept of using carefully constructed products to remove the randomness (risk) of a security. A delta value is selected to compensate and adjust the riskiness of an asset. No-arbitrage is the idea that *caeteris paribus* there are no opportunities in the market to make riskless profit. These foundations form the basis of the Black-Scholes model for derivative pricing, the pre-eminent derivative pricing formula in the world, of which the delta and associated values are used for hedging and trading (Wilmott 2007).

Black-Scholes assumes constant volatility. It is understood to be a deterministic function of time and the asset price however both these axioms are not true. Analyses of volatility processes show it to be a highly unstable quantity. Statistical modeling of volatility tends to measure different types of volatility. Actual volatility is an instantaneous measure of randomness in an asset at a given time. It is a very difficult quantity to measure. Realized (historical) volatility is a measure of randomness over a period in the past. Implied volatility is the volatility measure back-computed from market prices for a given derivative of an asset. Forward volatility refers to the volatility measure, whether implied or actual, over some future period.

These traditional methods of measuring volatility all suffer from the same problem: they cannot be the future value of volatility (Wilmott 2007). Either they are market views or esti-

mated from the past. The correct value to be used in pricing derivatives of an asset cannot be known until the derivative itself has expired.

As the volatility measure is not constant, not predictable and not directly observable it is best modeled as a random variable (Wilmott 2007). Understanding the dynamics of the volatility process in tandem with the dynamics of the underlying asset in the same timescale enable us to measure the stochastic volatility (SV) process. However, modeling volatility as a stochastic process requires an observable volatility measure: this is the stochastic volatility estimation problem. We are interested in predicting a random variable in aid of derivative pricing and securities trading. Choosing a stochastic volatility model requires us to postulate and validate asset price dynamics. The most popular stochastic volatility model is the Heston model which uses both square-root and Ornstein-Uhlenbeck processes. We shall proceed to describe and estimate this model.

4.1.1 Heston Stochastic Volatility Model

Recent reformulations of the original Fourier integrals of the Heston stochastic volatility model (Heston 1993) have led to numerically stable and efficient computations of derivative prices (Andersen 2008, Lewis 2000, Lipton 2002, Carr & Madan 1999, Lee 2004). The Heston model to estimate stochastic volatility is defined by the coupled two-dimensional stochastic differential equation:

$$\mathrm{d}X(t)/X(t) = \sqrt{V(t)}\mathrm{d}W_X(t) \tag{4.1}$$

$$\mathrm{d}V(t) = \kappa(\theta - V(t))\mathrm{d}t + \varepsilon\sqrt{V(t)}\mathrm{d}W_V(t) \tag{4.2}$$

where κ,θ,ε are strictly positive constants, and where W_X and W_V are scalar Brownian motions in some probability measure; we assume that $\mathrm{d}W_X(t) \cdot \mathrm{d}W_V(t) = \rho\mathrm{d}t$, where the correlation measure ρ is some constant in $[-1, 1]$. $X(t)$ represents an asset price process and is assumed to be a martingale in the chosen probability measure. $V(t)$ represents the instantaneous variance of relative changes to $X(t)$ - the stochastic volatility - and is modeled as a mean-reverting square-root diffusion, with Ornstein-Uhlenbeck dynamics. The parameters are understood as: κ is the variance mean-reversion rate; θ is the long-run variance to which the variance of the asset converges; and ε is the volatility of volatility.

Recognizing that the asset price process $X(t)$ is relatively close to geometric Brownian motion, it is sensible to work with logarithms of $X(t)$. By Ito's lemma we have:

$$d\ln X(t) = -\frac{1}{2}V(t)\mathrm{d}t + \sqrt{V(t)}\mathrm{d}W_X(t) \tag{4.3}$$

$$\mathrm{d}V(t) = \kappa(\theta - V(t))\mathrm{d}t + \varepsilon\sqrt{V(t)}\mathrm{d}W_V(t) \tag{4.4}$$

Euler discretization of the stochastic differential equation (4.3)-(4.4) takes the form:

$$\ln \hat{X}(t+\Delta) = \ln \hat{X}(t) - \frac{1}{2}\hat{V}(t)\Delta + \sqrt{\hat{V}(t)}Z_X\sqrt{\Delta} \tag{4.5}$$

$$\hat{V}(t+\Delta) = V(t) + \kappa(\theta - \hat{V}(t))\Delta + \varepsilon\sqrt{\hat{V}(t)}Z_V\sqrt{\Delta} \tag{4.6}$$

where $\hat{X}$ and $\hat{V}$ are discrete-time approximation to X and V, respectively, and where Z_X and Z_V are Gaussian random variables with correlation ρ.

A critical problem with the naive Euler discretization above enables the discrete process for V to become negative with non-zero probability, makings the computation of $\sqrt{\hat{V}}$ impossible. A full truncation scheme produces the smallest discretization bias (Lord, et al. 2006), leading to the dynamics:

$$\ln \hat{X}(t+\Delta) = \ln \hat{X}(t) - \frac{1}{2}\hat{V}(t)^{+}\Delta + \sqrt{\hat{V}(t)^{+}}Z_X\sqrt{\Delta} \tag{4.7}$$

$$\hat{V}(t+\Delta) = V(t) + \kappa(\theta - \hat{V}(t)^{+})\Delta + \varepsilon\sqrt{\hat{V}(t)^{+}}Z_V\sqrt{\Delta} \tag{4.8}$$

where the operator $x^{+} = \max(x, 0)$ enables the process for V to go below zero thereafter becoming deterministic with an upward drift $\kappa\theta$, and where $\hat{X}(t)$ is the observed price process and $\hat{V}(t)$ is the stochastic volatility process to be estimated. In addition we need to calibrate the parameters κ,θ,ε to run the recursive Bayesian estimators.

4.2 Financial Data

The Heston model's underlying economic rationale and reasoning leads to its prevalence in stock index derivative pricing. This owes to the fact that stock indices are less susceptible to jumps as single stocks, however the above formulation has also found use for valuing single stock derivatives. Our research was carried out on historical data from across two major financial markets. We examine the S&P500 (SPX), NASDAQ and FTSE100 (FTSE) stock indices. In addition we examine three common stocks: General Electric (GE), Citigroup (C) and AT&T (T). We use closing price data from the start of 2010 till the end of 2012 for each of these securities. As we are running an online learning algorithm we run our experiments on the whole dataset.

Working on the log return asset price evolution a number of points bear mentioning. Financial time series are leptokurtic, with unit-root tests indicating non-stationarity. There is, however, a degree of predictability in the markets as volatility clusters. This clustering is due to the autoregressive conditionally heteroskedastic (ARCH) nature of financial time series which if captured correctly leads to high concentration of positive returns. We can see such a series in Figure 4.1. Here we have the daily closing price for SPX from 04-Jan-2010 through 28-Dec-2012 and the associated log-return price process evolution. We clearly observe ARCH effects

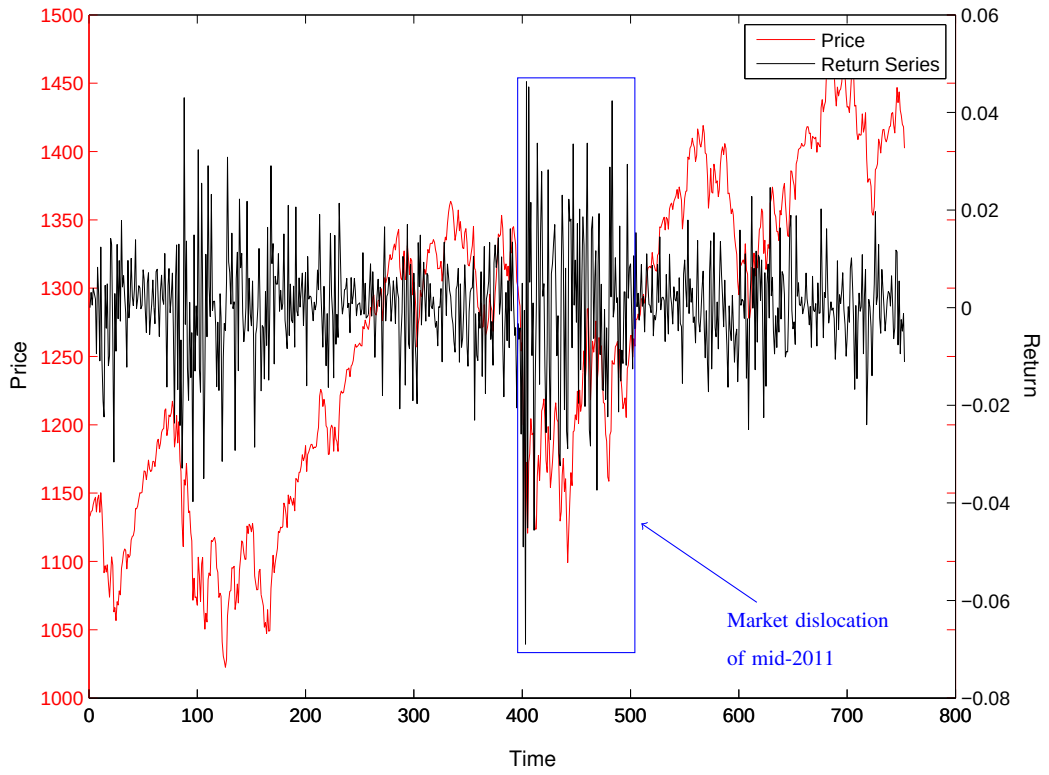

Figure 4.1: SPX daily closing price process for 04-Jan-2010 through 28-Dec-2012

throughout the series with this particularly acute in the middle of 2011 - around the 400 to 500 point mark highlighted in blue on the Figure.

4.3 Performance Analysis

Evaluating the estimation accuracy of sequential Monte Carlos is traditionally performed using the RMSE measure. There are a number of other performance metrics which could be used. Earlier, we saw Li & Honglei (2011) advocating the CRLB which is a computationally intensive calculation of estimation accuracy. Under the CRLB, if the MSE, or RMSE, is closer to the CRLB performance is better however Quang, et al. (2010) have demonstrated the convergence of the RMSE under an upper bound which does not show explicit dependency on the dimension of the hidden state. This classical result enables assignment of the RMSE as the Monte Carlo error (MC error) in discrete and continuous dynamical systems, the ultimate benchmark which is to be minimized.

4.4 Experimental Results

Given the price process we estimate the latent stochastic volatility process using the SIR, MCMC-PF, PLA and APPF particle filters run with $N = 1,000$ particles and systematic re-sampling. Furthermore, we explore the effects of increasing the number of particles. We take

the joint maximum a posteriori (MAP) estimates of κ and θ from our MCMC calibration and calculate the RMSE between the estimated volatility and actual stochastic volatility process for each particle filter. Our experiments were run on an Intel Core i7-2600k @ 3.4GHz processor with 16GB of DDR3 memory. Firstly we calibrate the model using an MCMC run and then run our experiments.

4.4.1 Stochastic Volatility Calibration & Parameter Estimation

Volatility calibration is an important part of computational finance having direct impact on both trading and pricing. MCMC methods have frequently been advocated for their path-dependent evaluation and parallel execution abilities. This is at the cost of extreme computational intensity (Ge, et al. 2000). To calibrate the SV process for each of our securities we ran a 10,000 iteration MCMC calibration to build an understanding of the price process (observation equation) and volatility process (state equation).

We took the joint MAP estimate of κ and θ from our MCMC calibration as per Chib, et al. (2002). The MAP estimate is a Bayesian parameter estimation technique which takes the mode of the posterior distribution. It is unlike maximum likelihood based point estimates which disregard the descriptive power of the MCMC process and associated pdfs. Our Heston model stochastic volatility calibration for SPX can be seen in Figure 4.2, where we can see the full truncation scheme forcing the SV process to be positive, and the associated parameter evolution can be seen in Figure 4.3. This process was repeated for the remaining securities of whose Heston stochastic volatility calibration and associated parameter estimation evolution can be seen in Figures 4.4 - 4.13.

Of note, we can see ε is a small constant throughout all the securities. This is attributable to the fact ε represents the volatility of volatility. If it were large and or varying across the securities we would not observe the coupling between and amongst securities in these markets as we do. This coupling, referred to as trend/momentum in finance, can be seen as the measure of similarity between the return processes in Figures 4.2, 4.4, 4.6, 4.8, 4.8 and 4.12. There are large periods of activity across certain time points (e.g. between $t = 400$ - $t = 500$ days) which are very similar across the securities indicative of an underlying trend within the market.

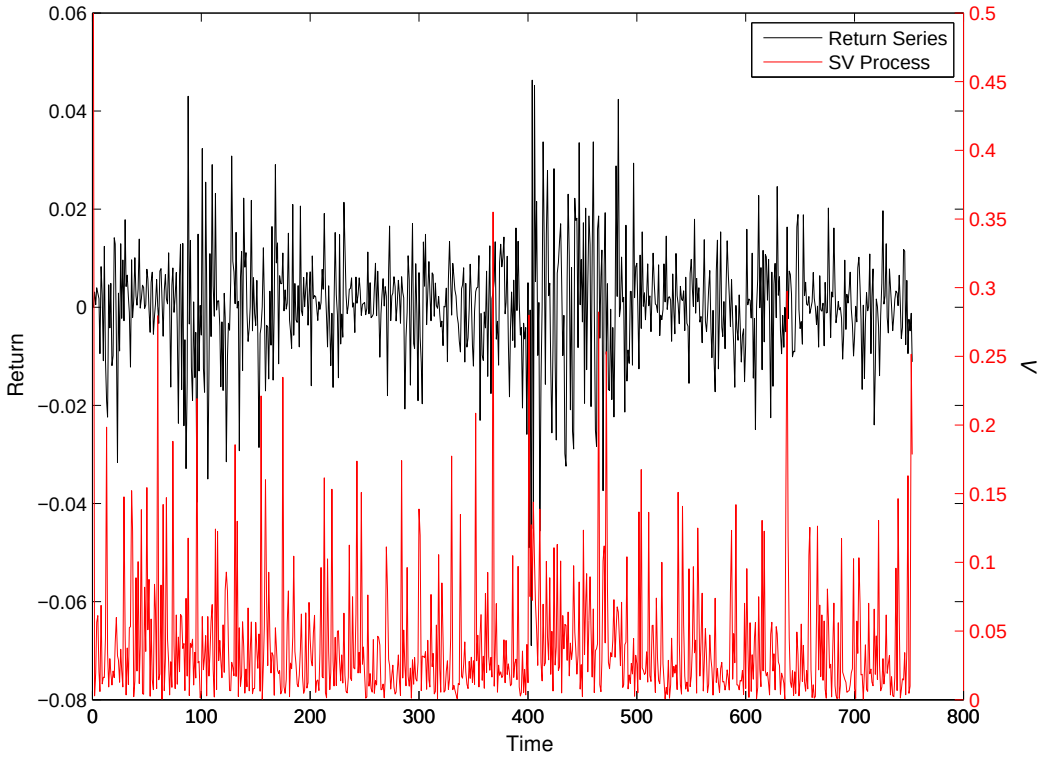

Figure 4.2: Heston model SPX daily closing stochastic volatility calibration using 10,000 iteration MCMC for 04-Jan-2010 through 28-Dec-2012

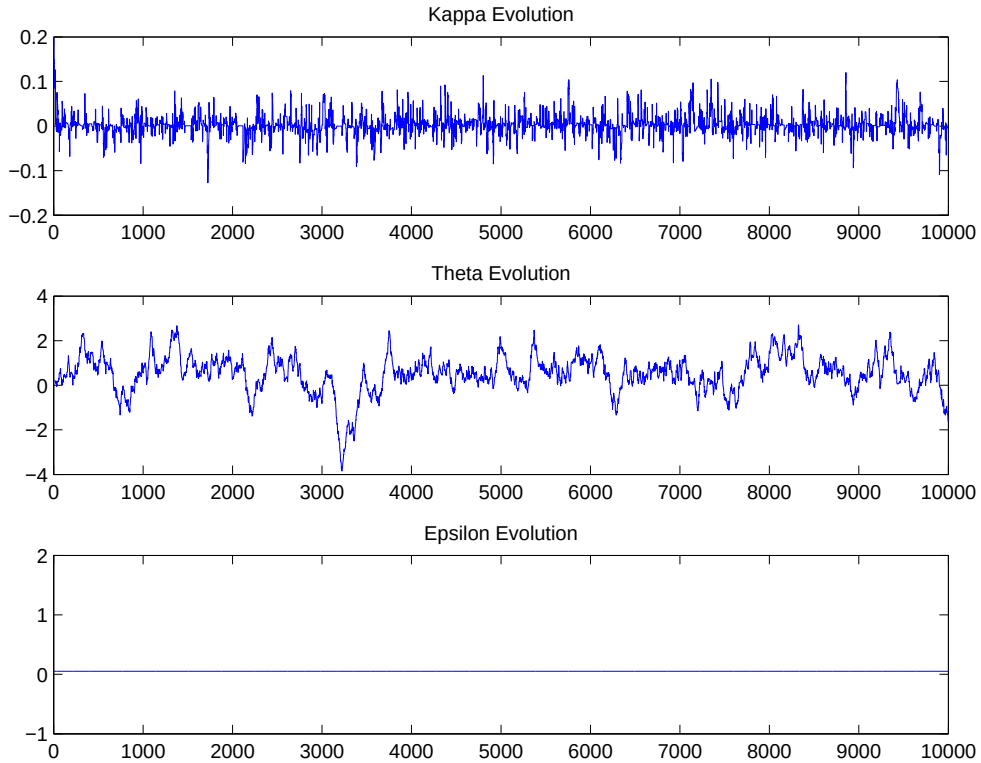

Figure 4.3: Heston model SPX daily closing stochastic volatility calibration using 10,000 iteration MCMC: parameter estimates and evolution for 04-Jan-2010 through 28-Dec-2012

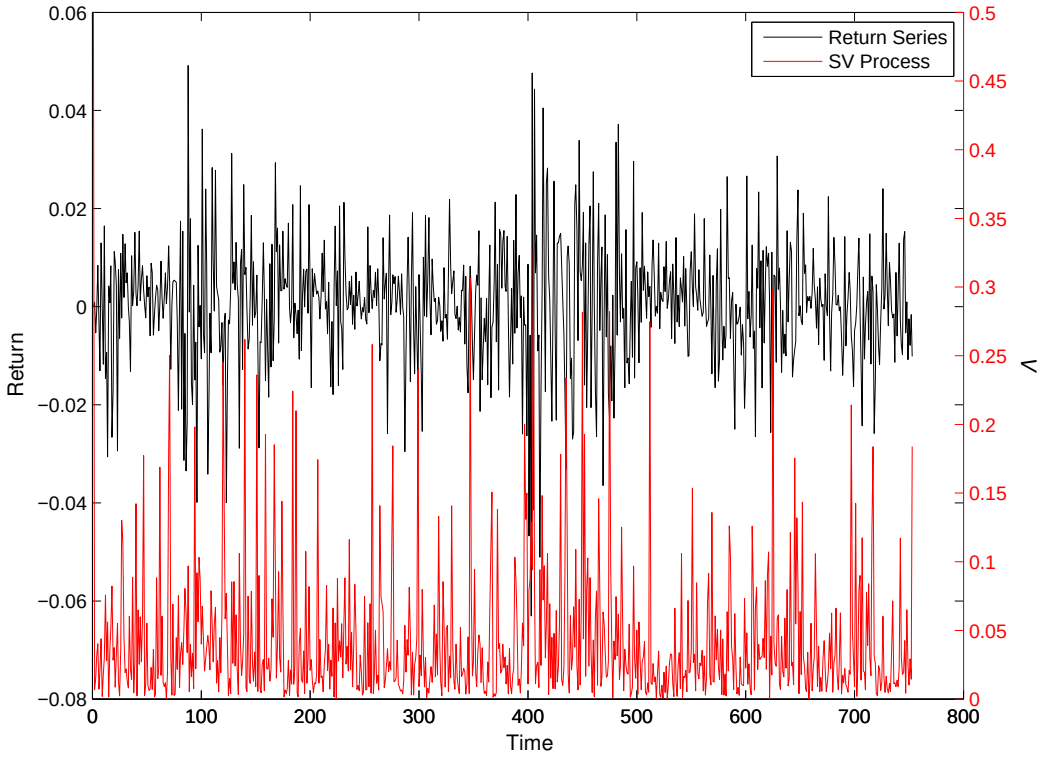

Figure 4.4: Heston model NASDAQ daily closing stochastic volatility calibration using 10,000 iteration MCMC for 04-Jan-2010 through 28-Dec-2012

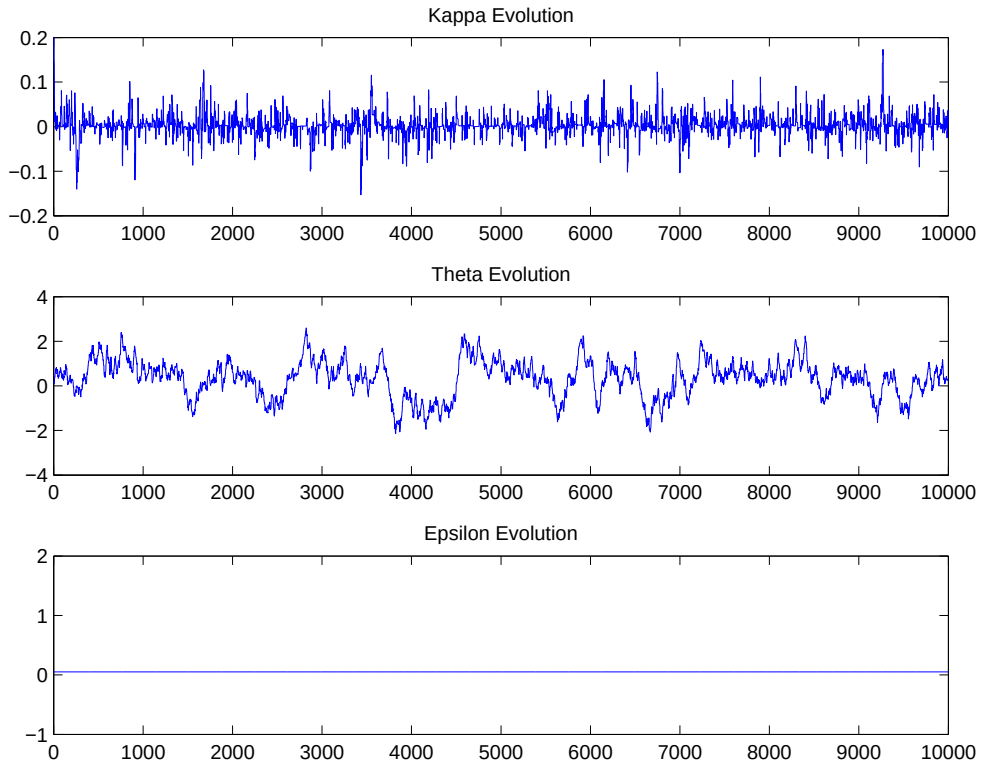

Figure 4.5: Heston model NASDAQ daily closing stochastic volatility calibration using 10,000 iteration MCMC: parameter estimates and evolution for 04-Jan-2010 through 28-Dec-2012

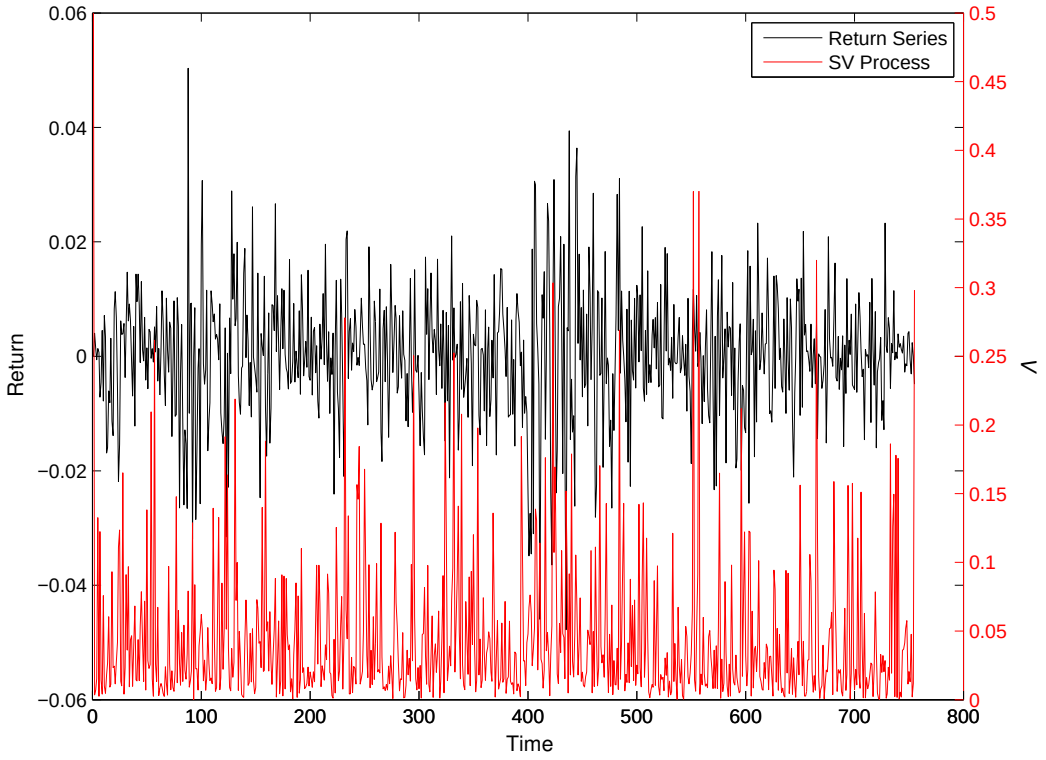

Figure 4.6: Heston model FTSE daily closing stochastic volatility calibration using 10,000 iteration MCMC for 04-Jan-2010 through 28-Dec-2012

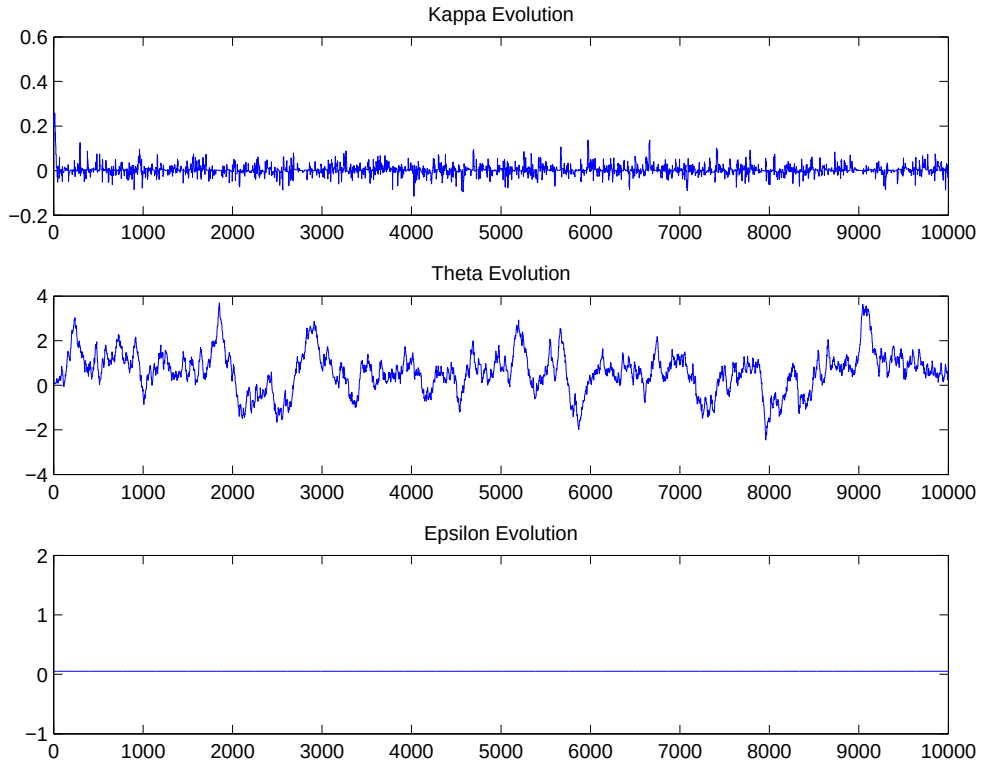

Figure 4.7: Heston model FTSE daily closing stochastic volatility calibration using 10,000 iteration MCMC: parameter estimates and evolution for 04-Jan-2010 through 28-Dec-2012

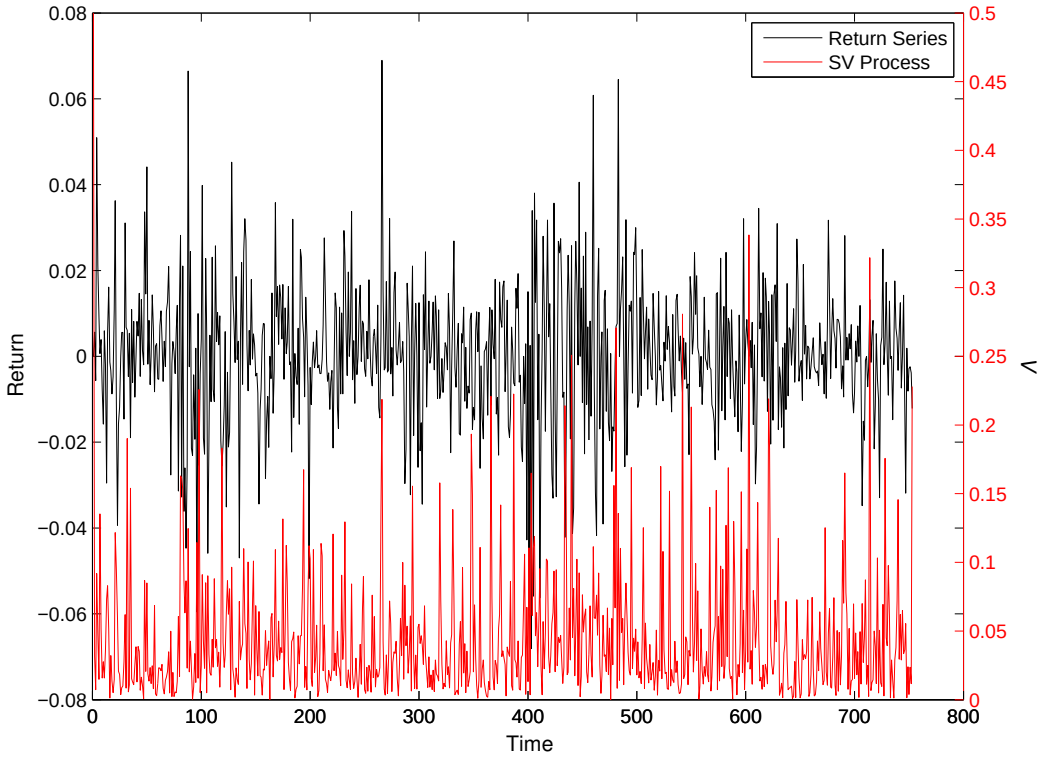

Figure 4.8: Heston model GE daily closing stochastic volatility calibration using 10,000 iteration MCMC for 04-Jan-2010 through 28-Dec-2012

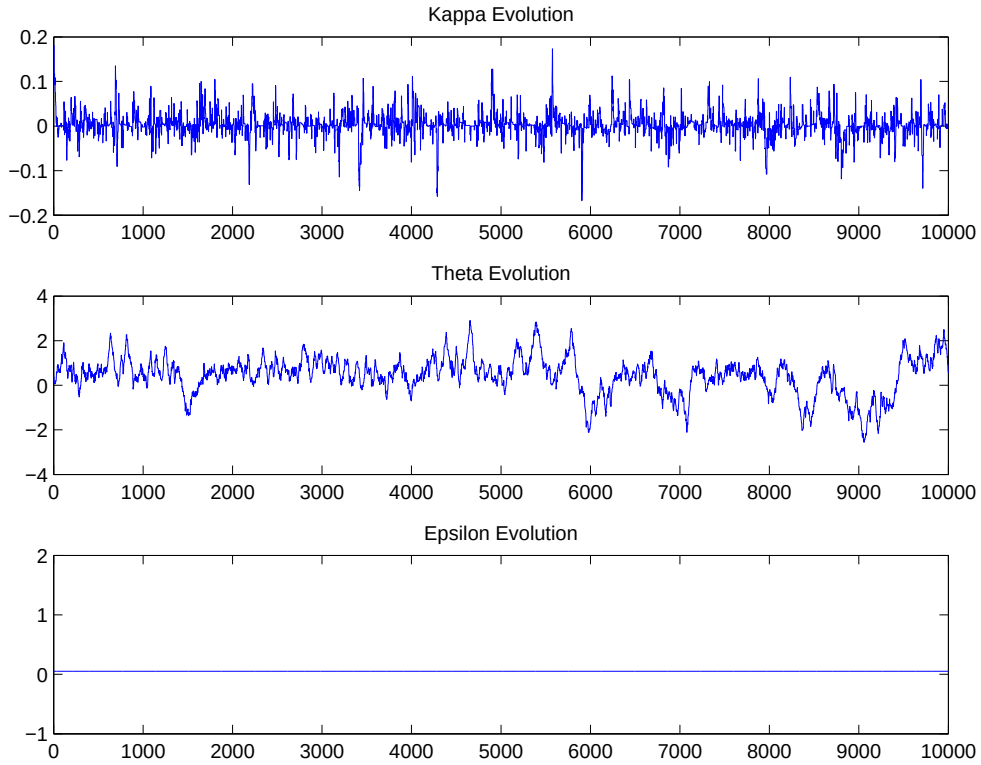

Figure 4.9: Heston model GE daily closing stochastic volatility calibration using 10,000 iteration MCMC: parameter estimates and evolution for 04-Jan-2010 through 28-Dec-2012

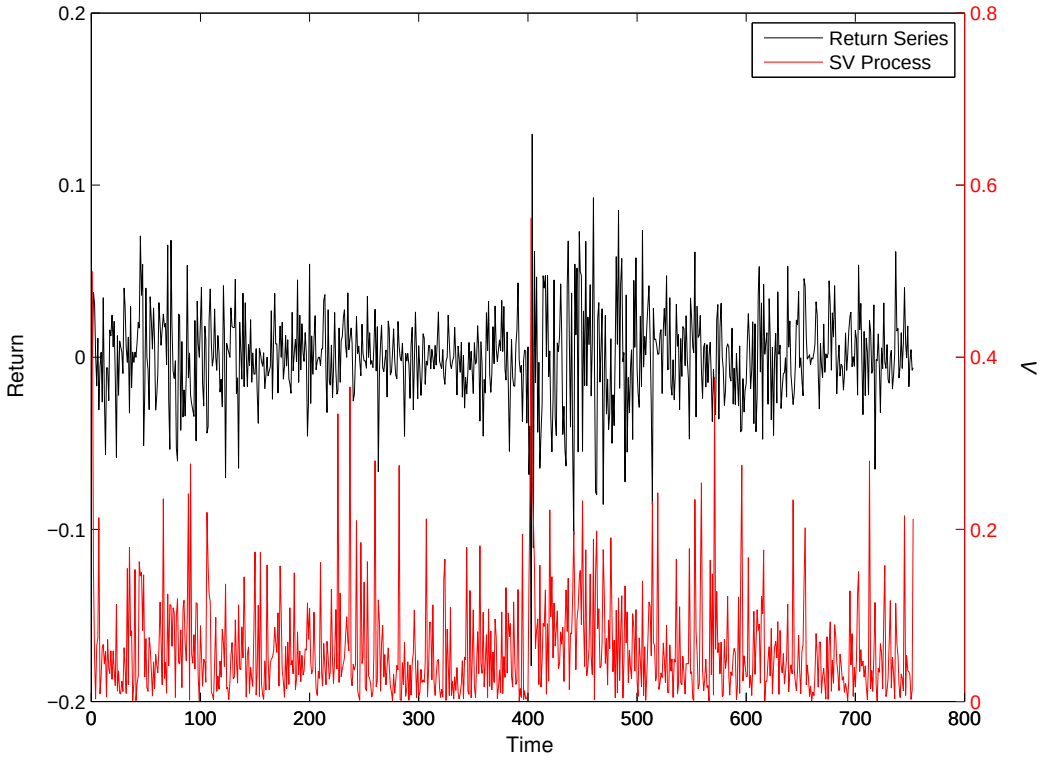

Figure 4.10: Heston model C daily closing stochastic volatility calibration using 10,000 iteration MCMC for 04-Jan-2010 through 28-Dec-2012

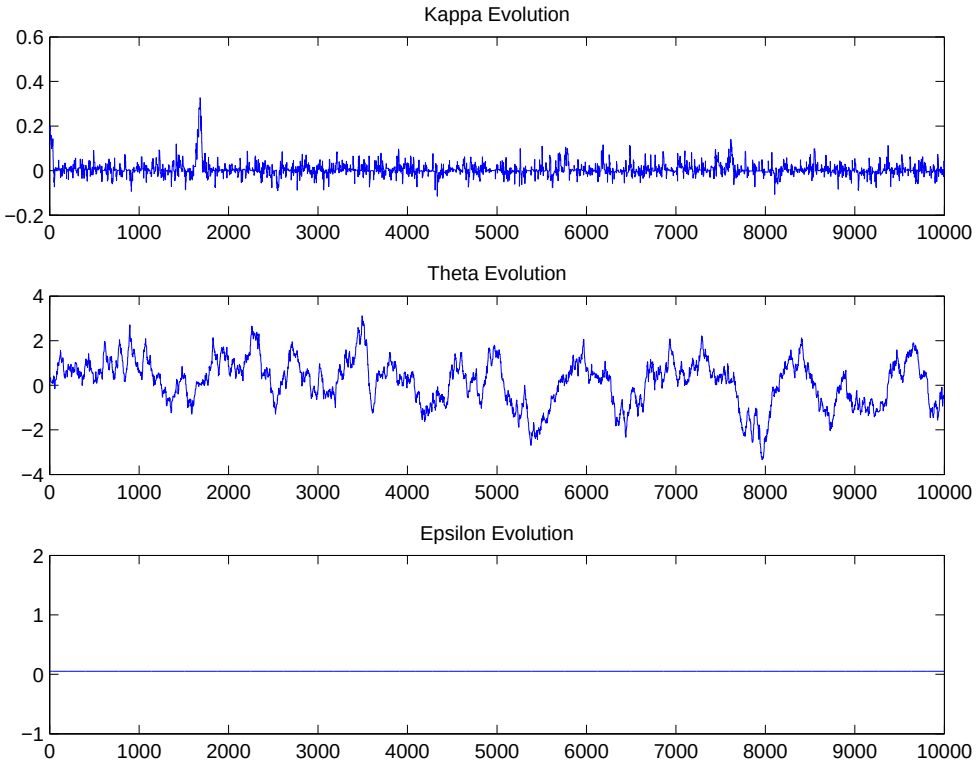

Figure 4.11: Heston model C daily closing stochastic volatility calibration using 10,000 iteration MCMC: parameter estimates and evolution for 04-Jan-2010 through 28-Dec-2012

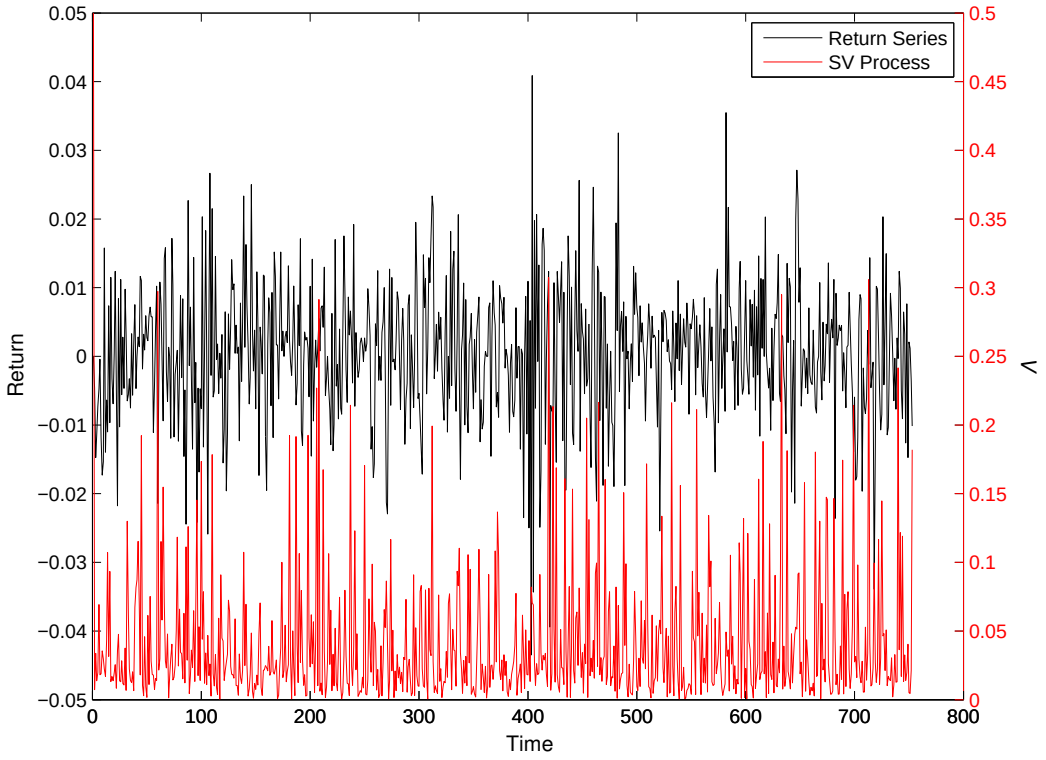

Figure 4.12: Heston model T daily closing stochastic volatility calibration using 10,000 iteration MCMC for 04-Jan-2010 through 28-Dec-2012

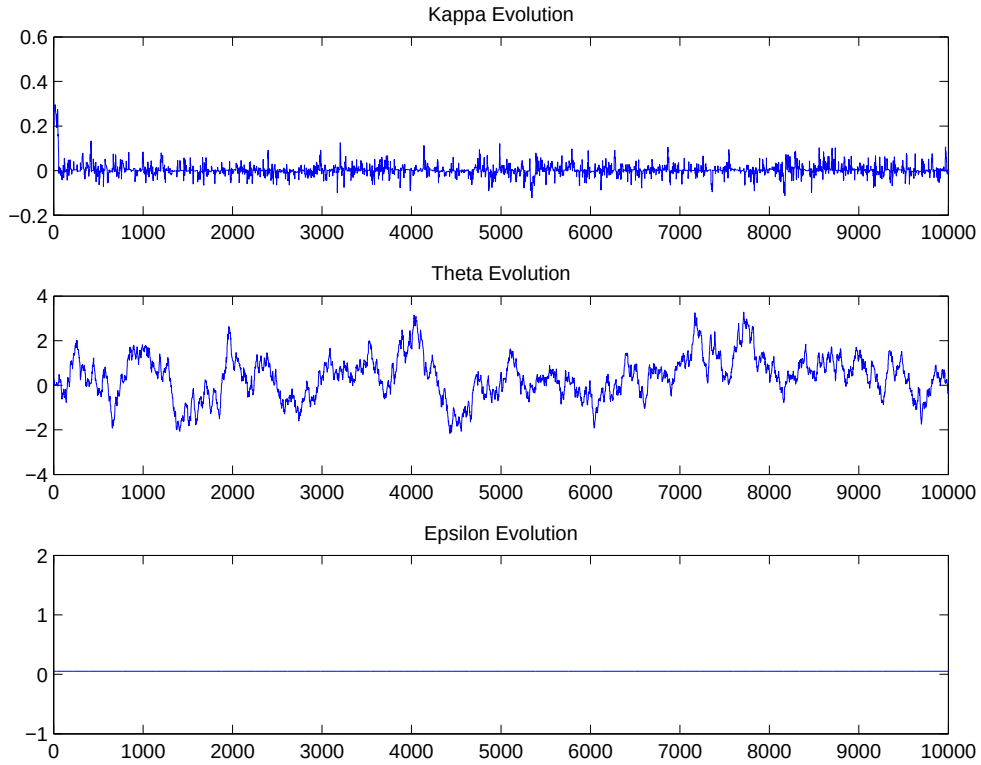

Figure 4.13: Heston model T daily closing stochastic volatility calibration using 10,000 iteration MCMC: parameter estimates and evolution for 04-Jan-2010 through 28-Dec-2012

4.4.2 Stochastic Volatility Estimation

Given the price process we estimate the latent stochastic volatility process using the SIR, MCMC-PF, PLA and APPF particle filters run with $N = 1,000$ particles and systematic resampling. Our results can be seen in Table 4.1 and in Figures 4.14 - 4.19 respectively. Across all securities we can clearly see the APPF providing more accurate estimates of the underlying stochastic volatility process compared to the other particle filters. As such the APPF provides statistically significant improvements in estimation accuracy compared to the other filters.

Upon estimating indices: in estimation of SPX SV (refer to Figure 4.14) we observe noticeably worse estimation by the MCMC-PF and PLA compared to the SIR PF with the APPF outperforming all these filters by a significant margin - a 6% improvement on SIR with moderate computational expense in reference to the other filters. In estimation of NASDAQ SV (refer to Figure 4.15) the MCMC-PF and PLA perform comparably to SIR with the APPF again outperforming all the filters, achieving a 5% increase in performance over SIR. We see similar performance in FSTE SV estimation (refer to Figure 4.16) with the MCMC-PF and PLA performing comparably to SIR and the APPF significantly outperforming all these filters. This trend continues upon estimation of the three common stocks. Estimating GE SV (refer to Figure 4.17) the MCMC-PF and PLA perform comparably to the SIR PF however the APPF markedly outperforms all three of these with an RMSE = 0.051 in comparison to PF RMSE = 0.061; a 16.6% improvement in performance. In estimation of C SV (refer to Figure 4.18), the MCMC-PF and PLA perform worse than the SIR PF, with the APPF outperforming all these filters once again with a 4.5% improvement in estimation accuracy compared to SIR. And finally, in estimation of T SV (refer to Figure 4.19), the MCMC-PF and PLA perform comparably to SIR whilst the APPF outperforms all these filters with a 4.8% improvement over SIR. Such impairments across the board can lead to both problematic trading and mispricing compared to derivatives factoring in the APPF SV estimates.

4.4.3 Stochastic Volatility Estimation with Increased Particle Set

To assess the influence of increasing the sample size on filter performance we ran an estimation of GE SV using $N = 5,000$ particles. Our results are summarized in Table 4.2. By taking the MAP estimate we reduced the parameter estimation problem to a 2-dimensional, deterministic control function and as such this enables us to focus our efforts on state estimation. We observe similar results as in the literature with only the PF showing any discernible increase in estimation performance. It is evident that increasing the particles does not increase APPF performance, outperforming the PF by 14.9%, and as such we can say the APPF provides robust performance with a limited particle set in comparison to the PF, MCMC-PF and PLA.

Table 4.1: Heston model experimental results. RMSE mean and execution time in seconds using 1,000 particles and systematic resampling.

SPX			NASDAQ		
	RMSE	Exec. (s)		RMSE	Exec. (s)
PF	0.05282	3.79	PF	0.05793	3.87
MCMC-PF	0.05393	59.37	MCMC-PF	0.05799	59.06
PLA	0.05317	21.30	PLA	0.05771	21.23
APPF	0.04961	39.33	APPF	0.05500	39.36
FSTE			**GE**		
	RMSE	Exec. (s)		RMSE	Exec. (s)
PF	0.06014	3.85	PF	0.06121	3.79
MCMC-PF	0.06067	58.95	MCMC-PF	0.06166	58.34
PLA	0.05970	21.29	PLA	0.06095	20.91
APPF	0.05725	39.32	APPF	0.05108	38.90
C			**T**		
	RMSE	Exec. (s)		RMSE	Exec. (s)
PF	0.06162	3.88	PF	0.05976	3.84
MCMC-PF	0.06233	59.32	MCMC-PF	0.05940	59.37
PLA	0.06175	21.33	PLA	0.06025	21.64
APPF	0.05887	39.65	APPF	0.05691	39.01

Table 4.2: Heston model experimental results - particle size experiment. RMSE mean and execution time in seconds using 5,000 particles and systematic resampling.

GE		
	RMSE	Exec. (s)
PF	0.06001	3.84
MCMC-PF	0.06154	58.80
PLA	0.06110	21.22
APPF	0.05108	39.22

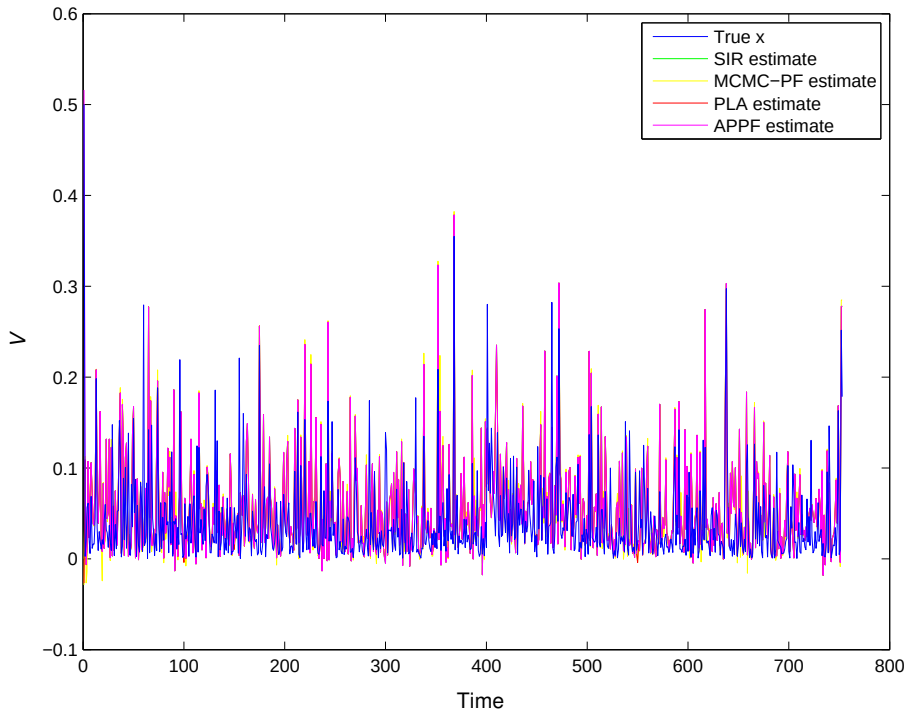

Figure 4.14: Heston model estimates for SPX - filter estimates (posterior means) vs. true state PF (RMSE=0.05282), MCMC-PF (RMSE=0.05393), PLA (RMSE=0.05317) and APPF (RMSE=0.04961)

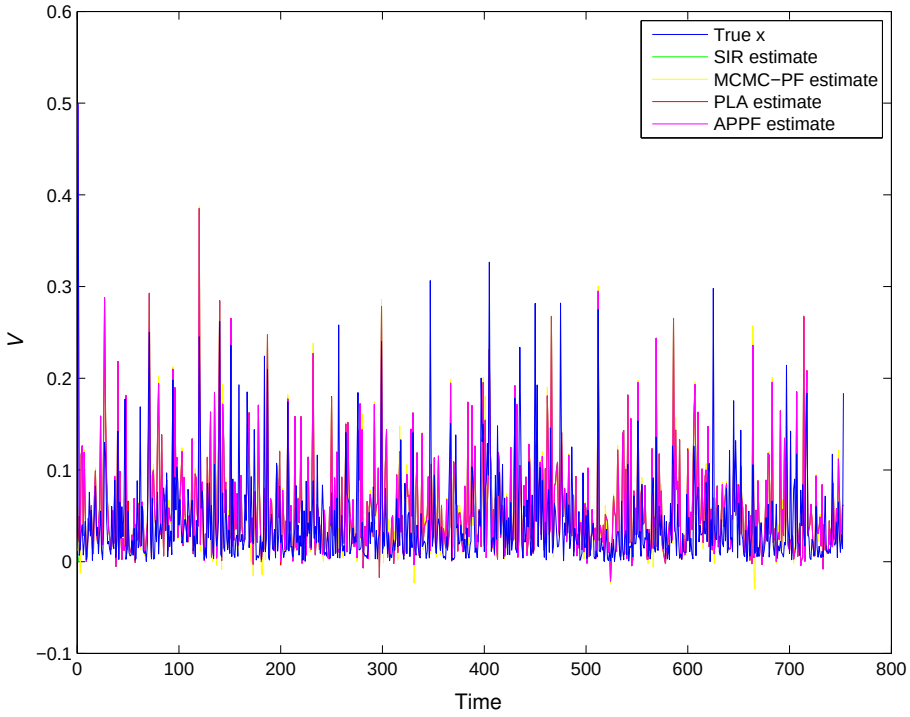

Figure 4.15: Heston model estimates for NASDAQ - filter estimates (posterior means) vs. true state PF (RMSE=0.05793), MCMC-PF (RMSE=0.05799), PLA (RMSE=0.05771) and APPF (RMSE=0.05500)

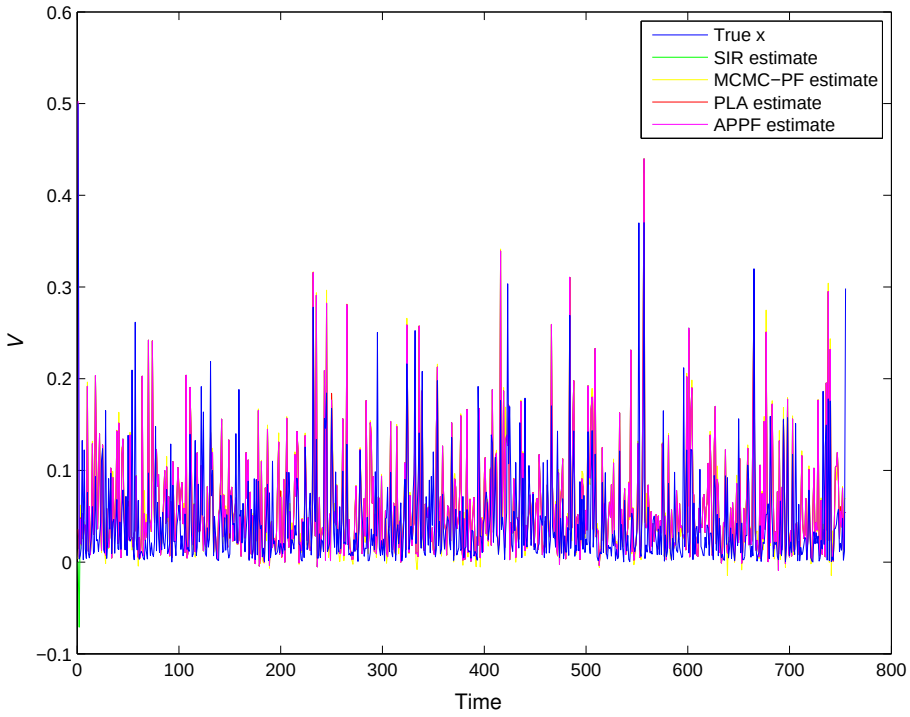

Figure 4.16: Heston model estimates for FTSE - filter estimates (posterior means) vs. true state PF (RMSE=0.06014), MCMC-PF (RMSE=0.06067), PLA (RMSE=0.05970) and APPF (RMSE=0.05725)

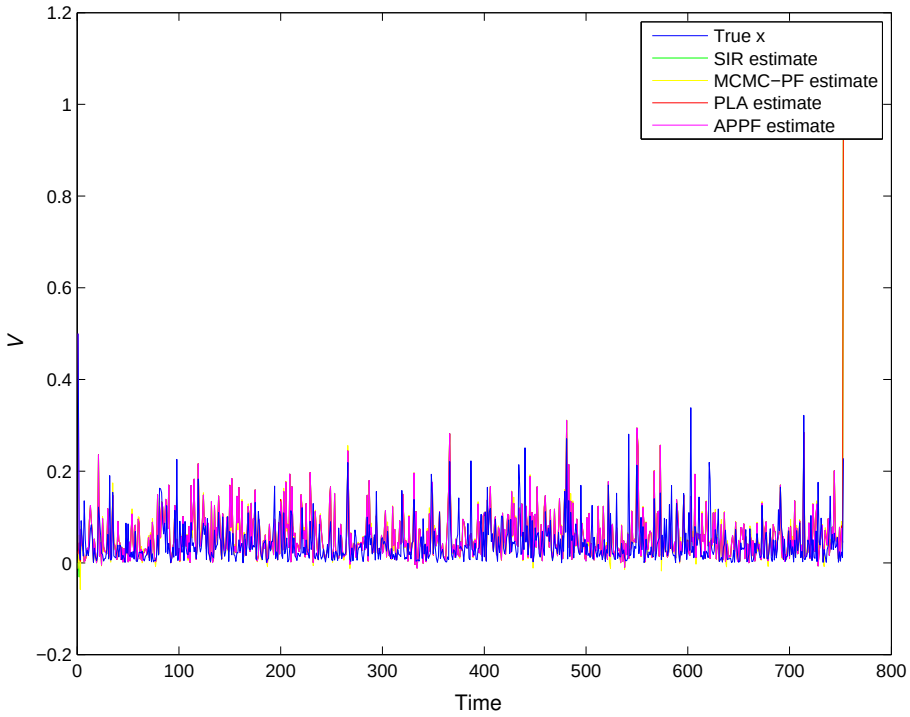

Figure 4.17: Heston model estimates for GE - filter estimates (posterior means) vs. true state PF (RMSE=0.06121), MCMC-PF (RMSE=0.06166), PLA (RMSE=0.06095) and APPF (RMSE=0.05108)

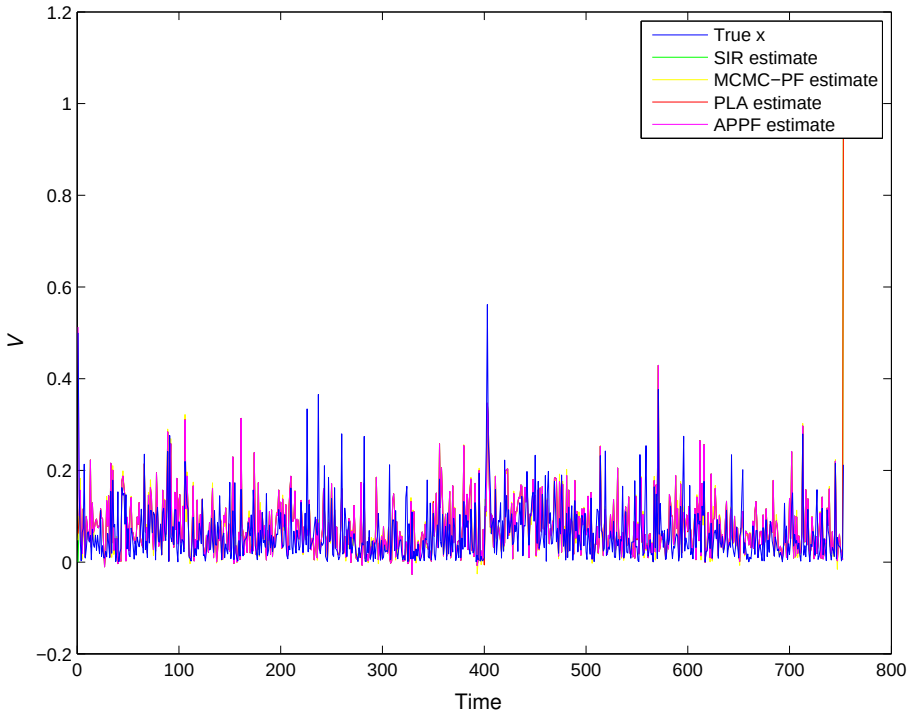

Figure 4.18: Heston model estimates for C - filter estimates (posterior means) vs. true state PF (RMSE=0.06162), MCMC-PF (RMSE=0.06233), PLA (RMSE=0.06175) and APPF (RMSE=0.05887)

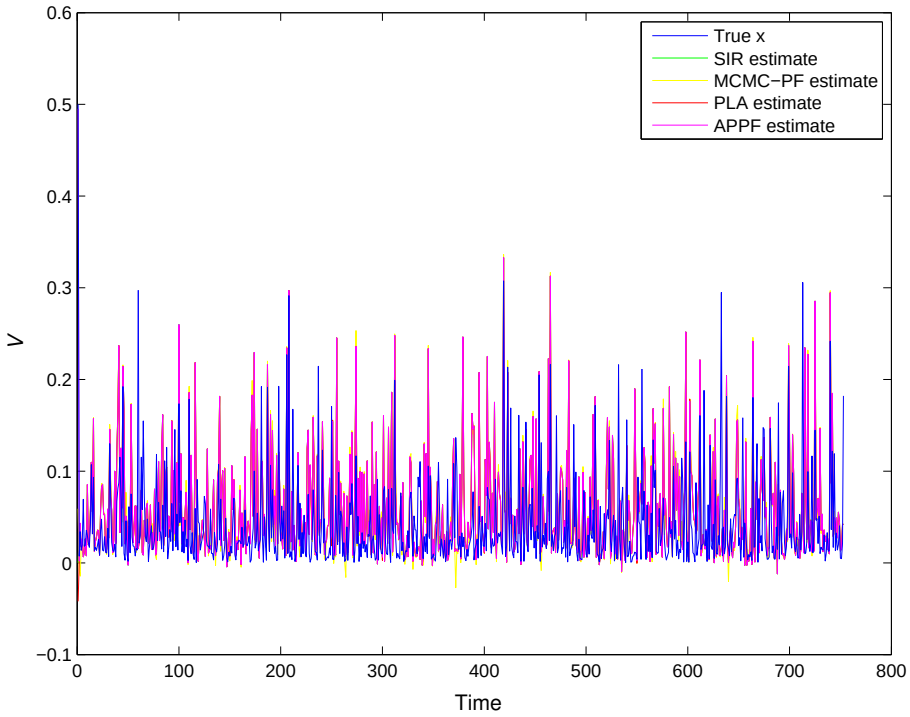

Figure 4.19: Heston model estimates for T - filter estimates (posterior means) vs. true state PF (RMSE=0.05976), MCMC-PF (RMSE=0.05940), PLA (RMSE=0.06025) and APPF (RMSE=0.05691)

4.5 Summary

We have proposed an evolutionary computation based particle filter for the estimation of stochastic volatility of indices and stocks. We have compared our method, the APPF with the traditional SIR particle filter, and the more advanced MCMC-PF and PLA. We find our filter to perform better in all state estimation experiments compared to these filters, with marked and statistically significant improvements in places.

Additionally, APPF performance does not increase when increasing the number of particles indicating we can get optimal results with a small number of particles. The APPF overcomes potential noise uncertainty in the weight update fitness evaluation. It is robust to changes in the signal once the optimal solution has been locked onto. Finally, the APPF overcomes uncertainty from the temporal fitness function.

These results go some way in showing that selective pressure from our generation-gap and distribution-recombination method does not lead to premature convergence. We have implicitly included a number of approaches to handling premature convergence in dynamic optimization problems with evolutionary computation (Jin & Branke 2005). Firstly, we generate diversity after a change by resampling. We maintain diversity throughout the run through the importance sampling diffusion of the current and past generation particle set. This generation based approach enables the learning algorithm to maintain a memory, which in turn is the base of Bayesian inference. And finally, our multi-population approach enables us to explore previously, possibly unexplored regions of the search space.

Chapter 5

Modeling & Estimation in Astrophysics

We continue our evaluation of APPF performance by modeling and estimating astrophysical time series. In aid of assessing efficacy and applicability of sequential Monte Carlos to astrophysical time series analysis we worked on modeling quasar time series and in understanding their latent dynamics. We introduce the domain and provide some background in Section 5.1. Our evaluation begins in Section 5.2 where we inspect the efficacy of sequential Monte Carlo methods on modeling regularized astrophysical time series. Building on these results we look towards modeling irregular astrophysical time series in Section 5.3. We summarize and conclude in Section 5.4.

5.1 Computational Statistics in Astrophysics

Owing to the proliferation of astrophysical data from deep-sky surveys past, present and future a key problem facing astronomy and the astrophysical community is with understanding the data. Such efforts are typically used to search for and/or predict phenomena within the sky. The usefulness of a survey is thus correlated to the analysis and understanding therewith. There has been significant research into the application of classification and traditional machine learning systems to astronomy alongside postulations of theoretical physical systems of the underlying dynamics. In this investigation we are concerned with understanding the latent dynamics of quasars, through the use of real-time learning techniques. Such an understanding can then be used to characterize light curves.

Quasars display high electromagnetic energy across all wavebands which comes from accretion onto a supermassive black hole. Such variability is typical of an active galaxy nucleus (AGN). The source of such variability is unclear though there have been efforts to describe the optical flux through thermal fluctuations (Kelly, et al. 2009), accretion disk instabilities (Kawaguchi, et al. 1998), supernovae (I., et al. 1997), microlensing (Hawkins 2004), and Poisson process models (Cid Fernandes, et al. 2000).

Results from reverberation mapping have shown that the broad emission lines respond to variations in the continuum emission after some lag, implying that continuum variations are dominated by the intrinsic process of the accretion disk (Peterson, et al. 2004). Kelly et al. (2009) built on these results and found that optical flux is driven by thermal fluctuations, on the base that optical emission was understood to be thermal emission from the accretion disk. Previous analyses of quasar optical flux were reduced to working with ensemble sets owing to the difficulty in obtaining high quality, well sampled light curve time series. Working with high quality light curves from across datasets, these results provides an important and powerful insight into the physics of the accretion disk and quasar dynamics.

Motivated by these results we conducted an investigation into the efficacy of online learning methods to astrophysical data - light curves from the MACHO dataset. The MACHO survey observed the sky from July 1992 through 1999 to detect microlensing events produced by massive compact halo objects (MACHO) in the Milky Way halo. Several tens of millions of stars were observed in the Large Magellanic Cloud (LMC), Small Magellanic Cloud (SMC) and Galactic bulge (Alcock, et al. 2000).

5.2 Statistical Modeling of Regularized Time Series

In principal recursive Bayesian estimation can be applied on any non-linear non-Gaussian dynamical system though typically it is prevalent in fields where problems are well known both theoretically and empirically. Astrophysics throws up a novel problem in not having mature models of the dynamical systems underlying observations, though we have promising theoretical and experimental models as those of Kelly et al. (2009) discussed in Section 5.1. Building on their insight we are encouraged to assess the dynamics from first principles.

Forecasters have been drawn to the benefits of parsimony (Hamilton 1994). Complex models perform well over periods on which the parameters have been trained though they often perform poorly on out-of-sample runs. It is noted that simpler models provide more robust forecasts. Box & Jenkins (1976) have been strong advocates of parsimonious modeling. Their approach to forecasting can be distilled into the following four steps:

(i) Transform the data, if necessary, so that it is covariance-stationary.

(ii) Make initial guesses of p and q which might describe an ARMA(p, q) model of the transformed series.

(iii) Estimate the autoregressive parameters $\phi(L)$ and the moving average parameters $\theta(L)$.

(iv) Perform diagnostic analysis to confirm model is consistent with observed features.

Traditional machine learning and artificial intelligence techniques assume regularized data (where the data is observed at regular time intervals). However, many systems (both natural and artificial) are largely irregular posing a major data problem prior to the main learning tasks. We proceed by regularizing our data through interpolation over a defined granularity to facilitate an investigation into the efficacy of sequential Monte Carlo methods for astrophysical time series. Thereafter we will relax the regularized sampling restriction and model irregular time series in Section 5.3. Employing the Box-Jenkins modeling philosophy we first consider the covariance-stationary case in Subsection 5.2.1 and then move onto assuming non-stationary data in Subsection 5.2.2.

5.2.1 Autoregressive Models

Autoregressive models leverage the statistical significance of their lagged autocorrelations in an attempt to predict the future. Consider an AR(p) autoregressive model of order p:

$$X_t = \sum_{i=1}^{p} \alpha_i X_{t-i} + \varepsilon_t \tag{5.1}$$

where $\alpha_1 \dots \alpha_i$ are the autoregressive parameters and ε_t is a white noise process with variance σ^2. It makes use of the predictive power of past observations with varying decay through the autoregressive parameters a_i. Given this feedback, trivially increasing the order p will lead to over-fitting, usually expressed in an unusually accurate fit on the current dataset but with little explanatory power to new datasets. We shall be considering a few AR(p) models which we proceed to define.

Given (5.1) we define an AR(1) process as:

$$X_t = \alpha X_{t-1} + \varepsilon_t \tag{5.2}$$

The state space dynamics are given by:

$$X_t = \alpha X_{t-1} + \varepsilon_t \tag{5.3}$$

$$Y_t = H X_t \tag{5.4}$$

where (5.3) is the state equation and (5.4) is the observation equation with $H = (1, 0, \dots, 0)$. Equivalence of an AR(p) and state space models can be found in Appendix A.

Given (5.1) we define an AR(2) process as:

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \varepsilon_t \tag{5.5}$$

The state space dynamics are given by:

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \varepsilon_t \tag{5.6}$$

$$Y_t = H X_t \tag{5.7}$$

where (5.6) is the state equation and (5.7) is the observation equation with $H = (1, 0, \ldots, 0)$.

5.2.2 Autoregressive Integrated Moving Average Models

Thus far we have assumed that our observed series is linear and stationary though given the unknown nature of the underlying dynamical system (it is entirely plausible that physical affects in the halo could introduce non-stationarity into our series) it would be beneficial to assess the stationarity of the series. To do so we shall use ARIMA models where we can transform a non-stationary series into a stationary series by considerings its differenced series (corresponding to step (i) of the Box-Jenkins modeling philosophy). The order of this differencing corresponds to the d of an ARIMA(p, d, q) model. Our prediction equations must now reflect the fact we are working on the differenced series. We define a number of linear growth ($d = 1$) ARIMA models that we shall be considering.

An ARIMA(*p,d,q*) process is given by:

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right)(1 - L)^d X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right)\varepsilon_t \tag{5.8}$$

where L is the lag operator, the α_i are the autoregressive parameters, the θ_i are the moving average parameters, and the ε_t are i.i.d. sample error terms from the normal distribution. Derivation of the ARIMA process can be found in Appendix B.

For an ARIMA$(1, 1, 0)$ of the form (5.8) the prediction states equation is:

$$X_t = \alpha X_{t-1} + \varepsilon_t + (X_{t-1} - \alpha X_{t-2}) \tag{5.9}$$

Full derivation of (5.9) can be found in Appendix B.1.

For an ARIMA$(2, 1, 0)$ of the form (5.8) the prediction states equation is:

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \varepsilon_t + (X_{t-1} - \alpha_1 X_{t-2} - \alpha_2 X_{t-3}) \tag{5.10}$$

Full derivation of (5.10) can be found in Appendix B.2.

5.2.3 Results

We tested regularized modeling on the MACHO QSO 63.7365.151 B-band light curve. The original, irregular light curve can be seen in Figure 5.1. Note the abnormal evolution within the series just after 4.95x10^4 MJD and 5x10^4 MJD, in addition to the massive spike just after 5.05x10^4 MJD. We remove bad measurements, such as the spike just after 5.05x10^4 MJD, and linearly interpolate using 1-dimensional piecewise cubic Hermite interpolation over 10 days with extrapolation to out-of-range values. Analysis of the underlying distribution pre and post-regularization does show a significant change, as is to be expected on such sparsely-sampled time series. The resultant regularized series can be seen in Figure 5.2.

We estimated the AR(1), AR(2), ARIMA(1,1,0) and ARIMA(2,1,0) models 100 times with the SIR PF, MCMC-PF and APPF using $N = 1,000$ particles and multinomial resampling. The parameters of the AR(p) models are estimated using the Burg method (an extended case of least-squares), using a set of forward and backward prediction equations to estimate the parameters. The ARIMA models were fitted using sequential quadratic programming multidimensional constrained linear optimization which allowed us to not only difference the data as per step (i) of the Box-Jenkins modeling philosophy but to also find the optimal parameters. Sample runs for each of the models can be seen in Figure 5.3 - Figure 5.6. Summary results across the runs are detailed in Table 5.1. We compare the RMSE, the variance of the RMSE and execution time.

Table 5.1: Regularized Time Series Experiment Results. RMSE mean, variance and execution time in seconds: 100 runs using 1,000 particles and multinomial resampling.

AR(1)

	RMSE	Var	Exec. (s)
PF	0.061409	(-9.84E-07)	5.19
MCMC-PF	0.064501	(-1.33E-05)	8.29
APPF	0.026691	(-4.67E-07)	7.54

AR(2)

	RMSE	Var	Exec. (s)
PF	0.046247	(-5.49E-07)	5.30
MCMC-PF	0.048789	(-2.64E-07)	16.43
APPF	0.027902	(-6.10E-07)	7.79

ARIMA(1,1,0)

	RMSE	Var	Exec. (s)
PF	0.044886	(-1.60E-06)	126.95
MCMC-PF	0.041431	(-4.84E-07)	137.49
APPF	0.027094	(-6.31E-07)	128.74

ARIMA(2,1,0)

	RMSE	Var	Exec. (s)
PF	0.038406	(-1.55E-06)	203.31
MCMC-PF	0.047284	(-8.38E-07)	213.24
APPF	0.028913	(-7.53E-07)	205.69

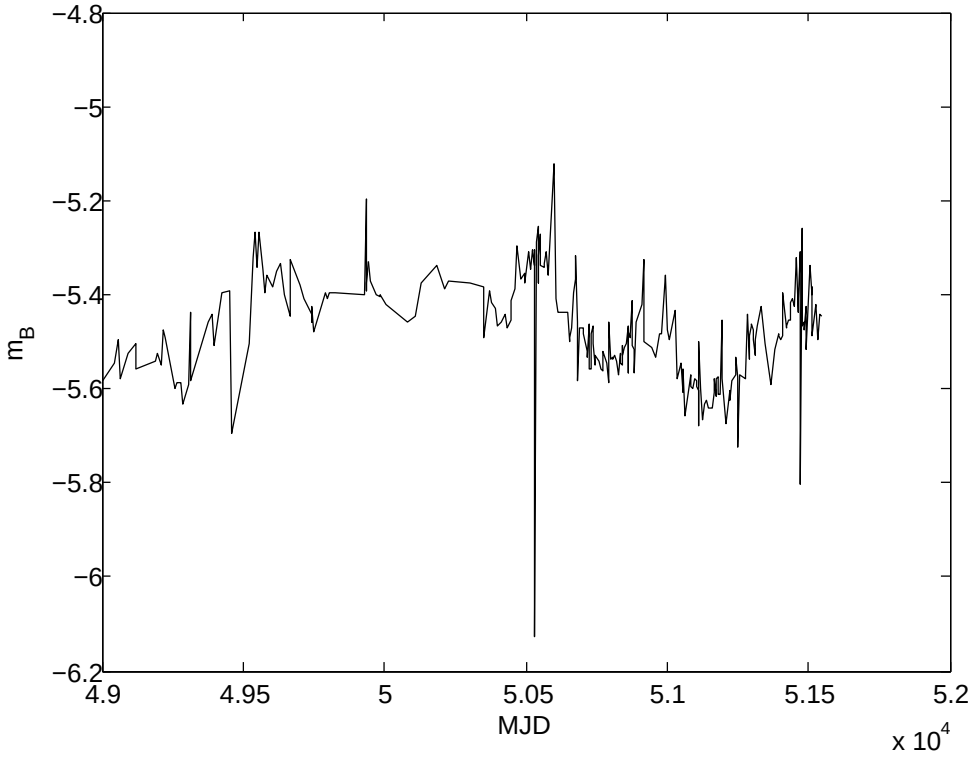

Figure 5.1: Source 63.7365.151 quasar light curve from the MACHO QSO candidates in the LMC fields dataset.

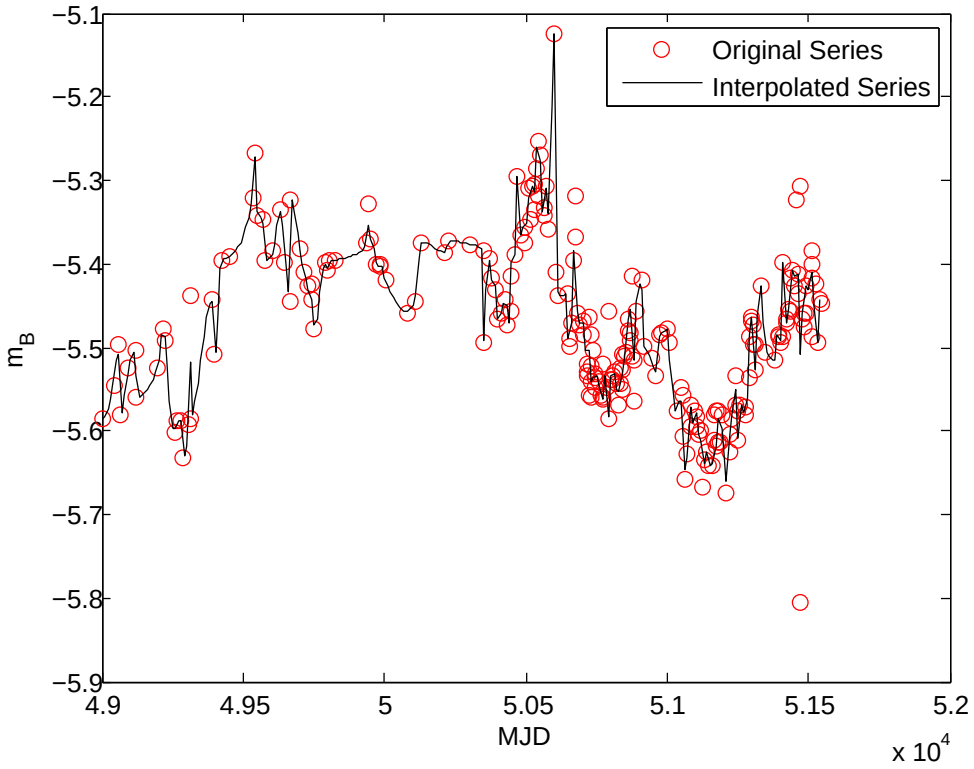

Figure 5.2: Source 63.7365.151 quasar light curve linearly interpolated using 1-dimensional piecewise cubic Hermite interpolation per 10 days.

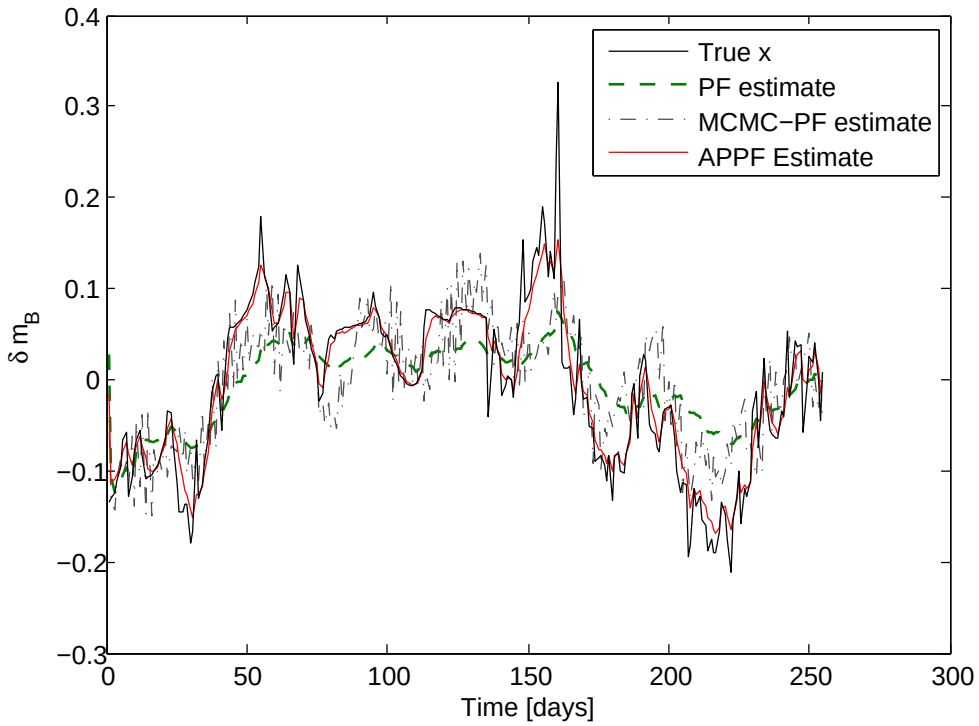

Figure 5.3: AR(1) - Filter Estimates (posterior means) vs. True State PF (RMSE=0.06108), MCMC-PF (RMSE=0.061934) and APPF (RMSE=0.026268)

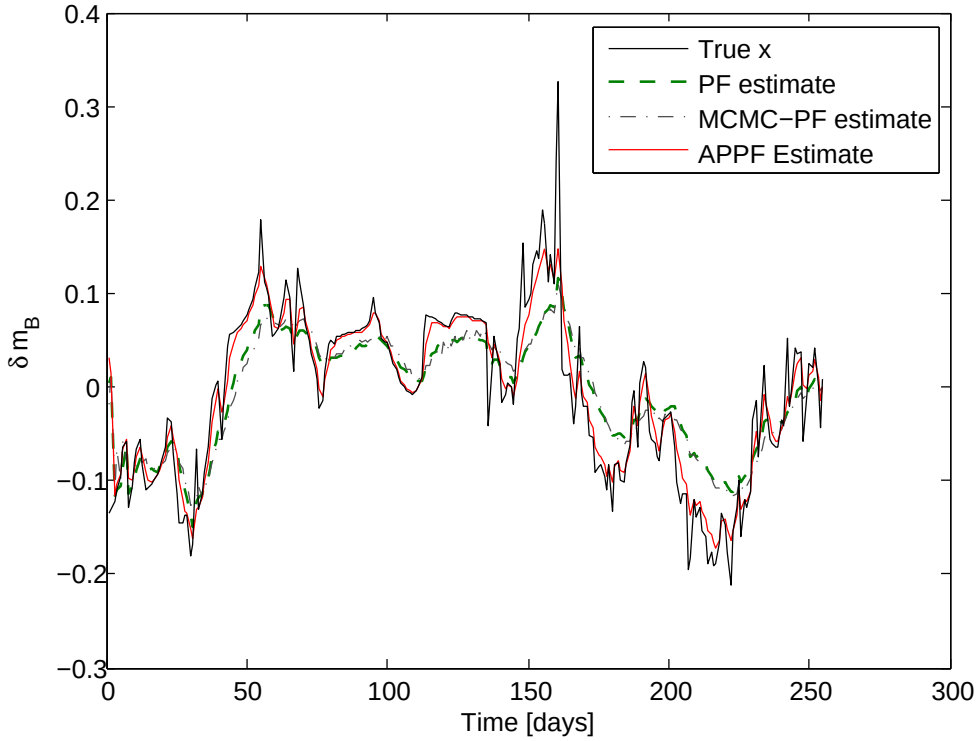

Figure 5.4: AR(2) - Filter Estimates (posterior means) vs. True State PF (RMSE=0.046927), MCMC-PF (RMSE=0.048948) and APPF (RMSE=0.028735)

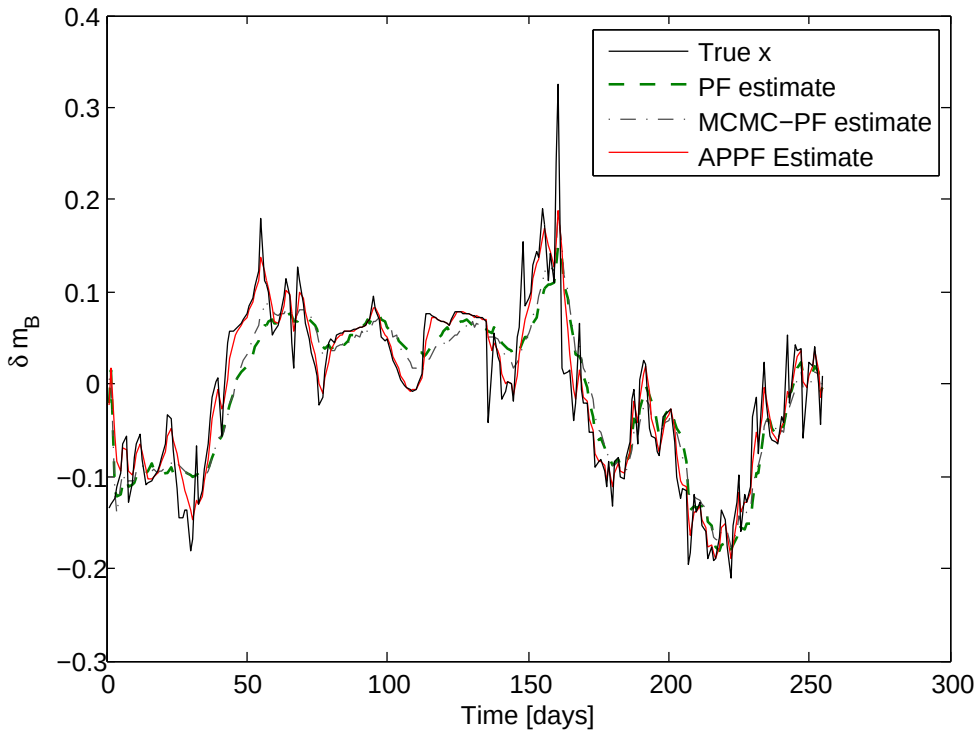

Figure 5.5: ARIMA(1,1,0) - Filter Estimates (posterior means) vs. True State PF (RMSE=0.043025), MCMC-PF (RMSE=0.04133) and APPF (RMSE=0.027015)

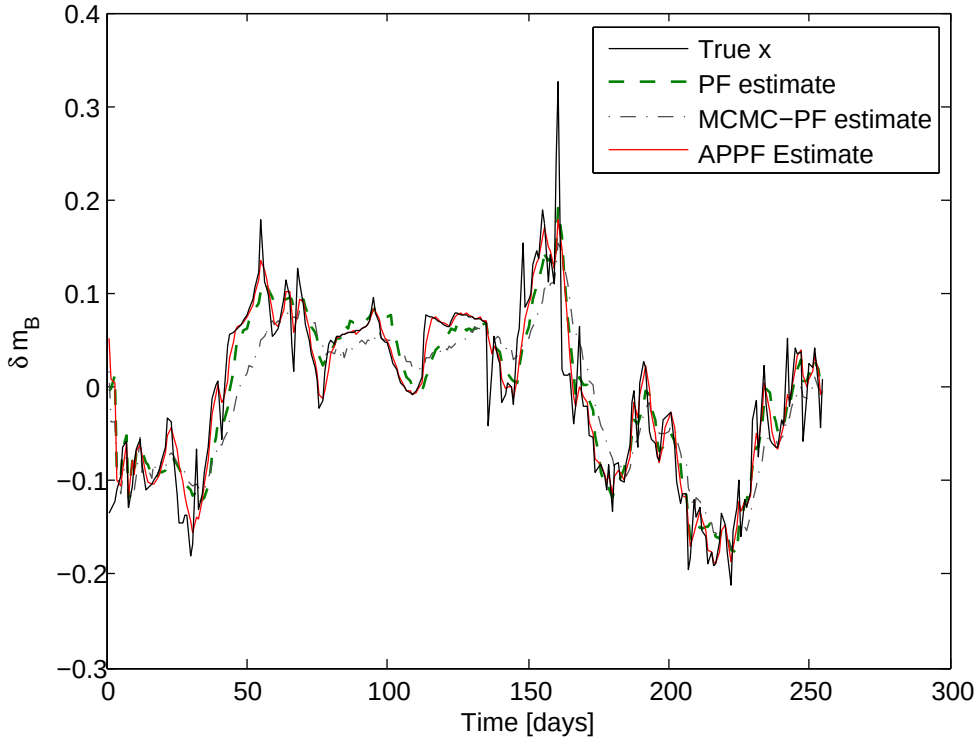

Figure 5.6: ARIMA(2,1,0) - Filter Estimates (posterior means) vs. True State PF (RMSE=0.036684), MCMC-PF (RMSE=0.046511) and APPF (RMSE=0.030455)

In analyzing the AR(p) models we see that though the performance of the AR(2) model increases for the PF and MCMC-PF it does not for the APPF. Closer analysis of Figure 5.3 for the AR(1) and Figure 5.4 for the AR(2) respectively show the PF and MCMC-PF to be significantly away from truly describing the latent signal. For the AR(1) we can see the MCMC-PF behaving quite erratically, it is tracking the signal through a lagged moving average window. Adding a further order to the AR expression enables the PF and MCMC-PF to estimate the state more accurately however not at the level of the APPF. The APPF tracking is fairly stable across the two models.

We can see that the AR(1) process describes the light curve well. Assessing the RMSEs of the APPF across the AR(1) and AR(2) we do not see a marked improvement. The PF shows a marked improvement with little increased computational overhead. The MCMC-PF estimation accuracy increases with the AR(2) but at egregious added computational expense. Most informative across these runs is the performance of the APPF. Here performance, across our experiments, as mentioned actually slightly deteriorates however still vastly outperforms both the traditional PF and MCMC-PF indicating the performance increase we are seeing for the PF and MCMC-PF is an artifact of over-fitting: an AR(1) process is sufficient to describe the light curve. Recollect naively adding autoregressive terms allows you to describe a series more accurately however limits its descriptive ability to out-of-sample series and points.

This thesis is further corroborated in analysis under assumptions of non-stationarity. See how for the ARIMA(1,1,0) in Figure 5.5 and the ARIMA(2,1,0) in Figure 5.6, performance of the PF and MCMC-PF increase substantially however, crucially APPF performance does not increase substantially, remaining within the bounds of the AR(p) model performance. In addition, there is a vast computational increase across all three filters which together with distinct lack of increase in estimation accuracy is indicative of limited-to-null non-stationarity within the series. We can thus say that the light curve can be characterized by a first-order autoregressive AR(1) process and does not have any significant non-stationarity present in the series.

The APPF tracks idiosyncrasies of the light curve across all the models. The PF and MCMC-PF go part way in achieving this in some of the models e.g. in the AR(2) model between $t = 0 - t = 40$ but quickly drop track as the series evolves and jumps. Here the APPF clearly displays its adaptability and robustness to noise and change in the underlying system. In these experiments we have shown that sequential Monte Carlos can be used to understand the latent dynamics of regularized astrophysical time series, and from amongst these the APPF provides the highest estimation accuracy.

5.3 Statistical Modeling of Irregular Time Series

Astrophysical time series are rarely regularly sampled. Discrete processes, as the AR(p) models above, are oftentimes obtained through observing a continuous process over a discrete sequence of time. It is natural thus to model discrete time observed flux as a continuous process: actual physical processes in the accretion disk are continuous and modeling as such allows us to handle our irregularly sampled light curves. Owing to these considerations we model the light curves as a first-order continuous-time autoregressive process (CAR(1)) (Brockwell & Davis 2002).

The CAR(1) process is described by the stochastic differential equation:

$$dX(t) = -\frac{1}{\tau}X(t)dt + \sigma\sqrt{dt}\epsilon(t) + bdt \tag{5.11}$$

where τ is the relaxation time of the process $X(t)$, and $\epsilon(t)$ is a Gaussian white-noise process with mean zero and variance one. Within the context of this work $X(t)$ is the quasar flux. The mean value of the light curve $X(t)$ is $b\tau$ and the variance is $\tau\sigma^2/2$. The relaxation time τ can be seen as describing the variability amplitude of the time series and from Kelly et al. (2009) has been shown to be significantly related to the black hole mass M_{BH}.

There have been a few previous attempts in using CAR(1) to model light curves. Kelly et al. (2009) show the CAR(1) process provides accurate descriptions of active galactic nuclei including the light curves under consideration in this paper. Pichara, et al. (2012) use calibrations of CAR(1) as inputs into classification methods for the EROS-2 and MACHO LMC datasets.

5.3.1 Parameter Estimation of the CAR(1) process

Parameters of CAR(1) processes are usually estimated through maximum likelihood of the observed series. Such an approach avoids cascading errors commonly found in non-parametric methods. Estimating characteristic parameters directly from the data gives us unbiased estimates though this requires one to assume a parametric model for the time series.

For a light curve with measured fluxes $\mathbf{x} = \{x_1, \ldots, x_n\}$ observed at times $\{t_1, \ldots, t_n\}$ with measurements error variances $\{\delta_1, \ldots, \delta_n\}$ the likelihood function of a CAR(1) process of the form (5.11) is a mixture of Gaussian functions:

$$p(\mathbf{x}|b, \sigma, \tau) = \prod_{i=1}^{n} \frac{1}{\sqrt{v_i}} f\left(\frac{x(t_i) - m_i}{\sqrt{v_i}}\right) \tag{5.12}$$

where $f(x) = n(x; 0, 1)$ is a standard normal, $\tau = 1/a$, $m_1 = b\tau$ and for $i > 1$:

$$m_i = e^{-\delta t/\tau}x(t_{i-1}) + b\tau\left(1 - e^{-\delta t/\tau}\right) \tag{5.13}$$

$$v_i = \frac{\sigma^2}{2}\tau\left[1 - e^{-2\delta t/\tau}\right] \tag{5.14}$$

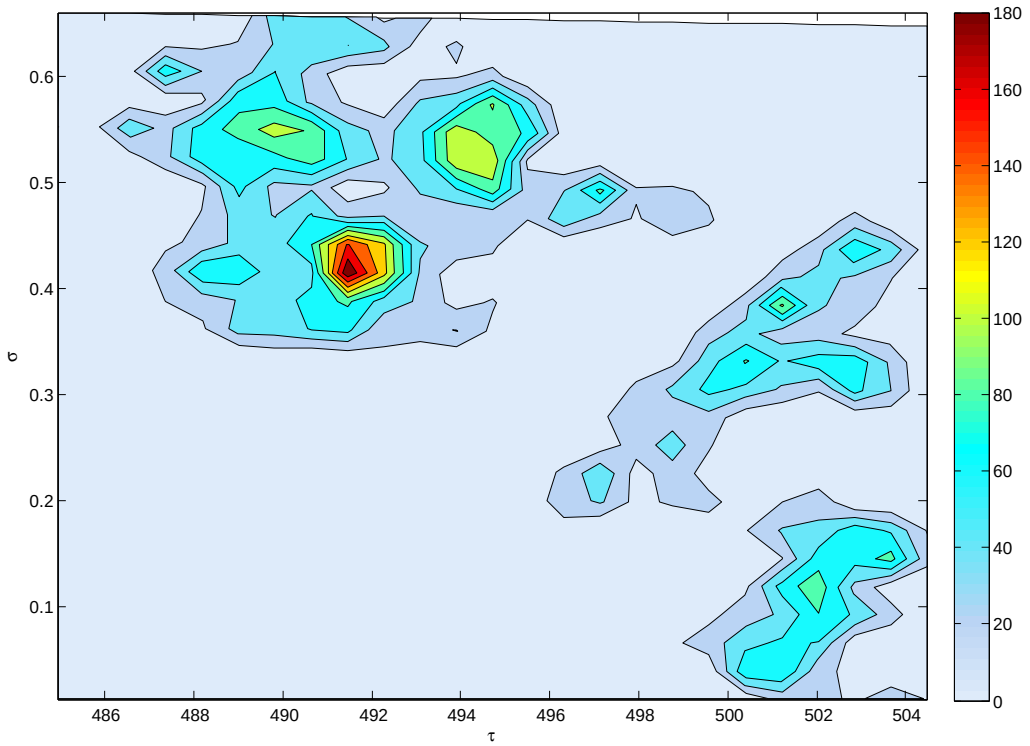

Figure 5.7: MAP estimation for MACHO source 63.7365.151 at $t = 220$ using MCMC with 10,000 samples. Here $\tau = 491.45$ and $\sigma = 0.42$.

Employing a Bayesian approach, we solve the likelihood using Metropolis-Hastings MCMC sampling. The probability distribution of the parameters (the posterior) is calculated as a product of the likelihood and a uniform prior. We propose a uniform prior as we assume any value of the autoregressive parameters a_i to be equally likely and do not a priori postulate any correlations between subsequent points. Such a prior is non-informative in the sense that all information comes from the data.

Upon parameter estimation, simply taking posterior median values would not be congruent with Bayesian analysis: the estimates would not significantly differ from a maximum likelihood fit (Sorenson 1980). We take the mode of the posterior distribution which corresponds to the MAP point estimation on the basis of the data. An example 2-dimensional MAP estimate of τ and σ for MACHO source 63.7365.151 at $t = 220$ can be seen in Figure 5.7. We can see that the highest peak (global maximum) of this surface corresponds to the point where $\tau = 491.45$ and $\sigma = 0.42$.

5.3.2 Results

We tested the CAR(1) model against 55 MACHO QSO B-band light curves. We remove bad measurements from the original, irregular light curves prior to a 10,000 iteration MCMC calibration. The filters are then run using MAP estimates from the calibration, 1,000 particles and

multinomial resampling. Summary results can be seen in Table 5.2 where τ is the characteristic timescale of the quasar light curve in days, calculated off of the last 100 points; σ_τ is the standard deviation of the characteristic timescale of the quasar light curve, calculated off of the last 100 points; τ_N is the standard deviation of the characteristic timescale of the quasar light curve, calculated off of the last 100 points; 95% CI τ_N is the conditional 95% confidence interval of the final characteristic timescale value; σ is the standard deviation of the quasar flux variations, calculated off of the last 100 points; σ_σ is the standard deviation of the standard deviation of the quasar flux variations, calculated off of the last 100 points; and σ_N is the standard deviation of the quasar flux variations of the final value of the quasar light curve in days.

From our calibration we notice a number of interesting items. The value of τ is tightly coupled across all quasars, with a tightly bounded variance, breaking from the understanding that τ is related to M_{BH} as posited by Kelly et al. (2009). Recent postulations within the astrophysical community suggest the presence of a unified underlying structure for accretion powered sources (Kazanas, et al. 2012). The appearance of such sources are different only because of differences in ionization (the spectrum of ionizing radiation) e.g. the wind density profile can accommodate several diverse and independent aspects of AGN phenomenology under one framework. This parsimonious framework describes AGNs using just two parameters: the dimensionless accretion rate $\dot{m}$ and the observers inclination angle θ. The M_{BH} it is argued by Boroson (2002) provides the overall scale of an object's luminosity and size. The characteristic timescale τ corresponds to the distance between x-rays and the galactic dust in the accretion disk (as related through the x-ray absorptions features of the QSO spectra) and should not, as indicated by the data, therefore have any correlation to the M_{BH}.

Significantly, we find that all filters perform well with the APPF outperforming both the traditional PF and the MCMC-PF in all cases. There are a number of scenarios where PF and MCMC-PF performance is similar, indicative of cases where the MCMC implementation as a PF is catching itself out in a pre and post-loop: the importance weights are reliant on the location before the MCMC move however the sample is reliant upon the location after the move which in essence causes removed samples to be replicated in resampling thus performing comparably to the generic PF. Assessing performance across the filters, we can see the heuristic step within the APPF (that being the only difference between it and the PF) imbues the filter with a level of performance far beyond the PF and MCMC-PF in most cases.

Examining some specific scenarios we see the filter run for MACHO source 63.7365.151 in Figure 5.8 and the associated MAP evolution of τ in Figure 5.9. Recollect this is the same quasar light curve we regularized and estimated using both AR(p) and ARIMA(p, d, q) models

Table 5.2: Irregular Time Series Experiments: CAR(1) calibration of 55 MACHO light curves.τ and σ were calibrated using a 10,000 run MCMC; RMSE means of the particles filters compared to the true observations using $N = 1,000$ particles and multinomial resampling.

Source	Points	τ [a]	σ_τ [b]	τ_N [c]	95% CI τ_N [d]	σ [e]	σ_σ [f]	σ_N [g]	PF	MCMC-PF	APPF
1.4418.1930	1033	497.75	11.92	500.53	[488.12 : 495.63]	0.27	0.21	0.3	0.04444	0.02392	0.01078
1.4537.1642	1186	496.4	14.06	499.12	[487.90 : 496.38]	0.27	0.18	0.24	0.02123	0.02068	0.01488
11.8988.1350	1014	499.12	14.16	503.68	[496.01 : 501.81]	0.31	0.25	0.28	0.02065	0.01893	0.01253
13.5717.178	937	501.46	12.71	508.49	[500.08 : 505.17]	0.29	0.24	0.35	0.02118	0.01923	0.01349
13.5962.237	930	498.99	14.92	471.90	[476.87 : 487.06]	0.32	0.28	0.19	0.02624	0.02546	0.01584
13.6805.324	1006	498.76	13.16	508.78	[511.28 : 519.04]	0.26	0.2	0.06	0.02173	0.02364	0.01364
13.6808.521	926	499.85	15.45	495.52	[473.37 : 486.78]	0.29	0.22	0.6	0.01920	0.02376	0.01242
14.8249.74	881	501.96	14.87	501.15	[509.70 : 518.41]	0.25	0.18	0.26	0.01661	0.01830	0.01402
17.2227.488	436	500.7	12.2	516.40	[501.79 : 510.64]	0.27	0.24	0.26	0.03330	0.03063	0.02554
17.3197.1182	428	502.42	12.72	498.03	[485.46 : 493.91]	0.26	0.21	0.03	0.03030	0.02991	0.01620
2.5873.82	896	499.53	13.46	495.90	[500.04 : 506.39]	0.28	0.22	0.12	0.02277	0.02052	0.01281
20.4678.600	355	501.14	17.01	500.81	[498.69 : 505.01]	0.27	0.27	0.46	0.02458	0.03042	0.01781
206.16653.987	882	501.1	14.74	504.97	[503.99 : 509.70]	0.28	0.23	0.05	0.02155	0.02142	0.01324
206.17052.388	901	499.45	13.08	503.07	[493.63 : 499.98]	0.28	0.22	0.24	0.01745	0.01933	0.01041
207.16310.1050	988	500.61	14.25	509.43	[502.16 : 508.44]	0.32	0.25	0.12	0.02730	0.02268	0.01091
207.16316.446	834	499.73	13.35	501.47	[499.12 : 503.83]	0.26	0.22	0.2	0.02213	0.01974	0.01226
208.15799.1085	967	500.56	14.71	531.80	[517.81 : 529.89]	0.3	0.23	0.5	0.02503	0.02348	0.00901
208.15920.619	992	499.79	13.64	511.06	[505.21 : 514.01]	0.3	0.22	0.04	0.01708	0.01939	0.01016
208.16034.100	973	500.35	13.39	504.55	[480.89 : 492.83]	0.28	0.22	0.3	0.01774	0.01688	0.01122
211.16703.311	863	500.1	14.3	506.99	[498.54 : 504.93]	0.29	0.22	0.47	0.01801	0.01906	0.01190
211.16765.212	856	498.48	14.9	492.84	[487.84 : 493.87]	0.31	0.24	0.46	0.02607	0.02070	0.00952
22.4990.462	551	498.29	13.93	498.19	[485.47 : 493.17]	0.24	0.23	0.34	0.02441	0.02711	0.01396
22.5595.1333	562	497.6	13.73	497.59	[501.07 : 506.39]	0.3	0.26	0.04	0.02819	0.03034	0.01651
25.3469.117	360	501.59	15.32	502.24	[488.32 : 495.35]	0.2	0.22	0.18	0.02261	0.02980	0.01317
25.3712.72	347	500.14	15.04	499.04	[507.27 : 517.35]	0.24	0.22	0.35	0.01656	0.01566	0.01561
28.11400.609	329	501.41	13.78	518.69	[510.74 : 518.74]	0.25	0.22	0.91	0.03785	0.03889	0.02985
30.11301.499	296	500.27	12.79	495.66	[491.45 : 495.69]	0.23	0.21	0.06	0.07567	0.04284	0.02564
37.5584.159	272	501.05	14.07	505.02	[496.67 : 502.98]	0.27	0.24	0.26	0.03282	0.03327	0.01798
48.2620.2719	345	498.87	14.93	485.52	[479.38 : 488.62]	0.26	0.21	0.07	0.03125	0.02685	0.01203
5.4643.149	896	501.48	12.96	497.99	[492.38 : 500.83]	0.26	0.22	0.07	0.01978	0.02187	0.01368
5.4892.1971	1002	499.35	14.6	495.40	[499.10 : 506.67]	0.29	0.24	0.81	0.02232	0.02312	0.01433
52.4565.356	249	500.33	12.2	516.17	[510.42 : 519.07]	0.19	0.19	0.23	0.03371	0.02845	0.01834
53.3360.344	250	500.63	13.78	499.59	[507.68 : 515.92]	0.21	0.21	0.59	0.05704	0.04432	0.03391
53.3725.29	238	500.89	12.34	496.74	[488.42 : 496.80]	0.23	0.23	0.06	0.03330	0.04460	0.01751
53.3970.140	266	501.37	12.67	514.41	[491.67 : 505.44]	0.27	0.25	0.37	0.01634	0.01346	0.01236
58.5903.69	236	500.95	13.79	510.93	[492.45 : 503.64]	0.23	0.21	0.31	0.04874	0.03388	0.02139
58.6272.729	312	499.17	13.62	488.49	[488.44 : 496.85]	0.23	0.2	0.16	0.05335	0.05251	0.04612
59.6398.185	268	499.05	14.29	509.79	[499.91 : 505.35]	0.22	0.21	0.32	0.03696	0.03839	0.01233
61.8072.358	374	500.69	15.42	519.16	[507.99 : 514.14]	0.26	0.23	0.35	0.02838	0.03421	0.01216
61.8199.302	405	499.26	13.67	505.95	[499.76 : 505.11]	0.2	0.18	0.07	0.02358	0.02297	0.01151
63.6643.393	241	499.41	14.97	509.57	[508.56 : 514.44]	0.26	0.24	0.02	0.04686	0.04094	0.02344
63.7365.151	229	497.92	14.38	494.29	[495.78 : 500.38]	0.21	0.19	0.04	0.03547	0.03674	0.01587
64.8088.215	260	501.02	13.47	500.46	[500.42 : 506.38]	0.26	0.24	0.53	0.05011	0.04561	0.02843
64.8092.454	260	501.77	14.88	498.97	[499.58 : 503.46]	0.23	0.19	0.24	0.03401	0.03380	0.02044
68.10968.235	258	499.88	11.76	506.65	[479.26 : 491.11]	0.24	0.22	0.22	0.02845	0.03869	0.01386
68.10972.36	264	500	10.92	500.70	[502.26 : 512.07]	0.23	0.21	0.46	0.03724	0.04330	0.02171
69.12549.21	233	499.03	13.1	499.93	[471.73 : 484.87]	0.25	0.24	0.11	0.02401	0.02641	0.02187
70.11469.82	246	499.9	11.87	502.89	[503.92 : 510.43]	0.21	0.19	0.2	0.04503	0.04859	0.04106
75.13376.66	236	500.77	10.33	507.23	[516.10 : 526.96]	0.21	0.18	0.4	0.03014	0.03601	0.02316
78.5855.788	873	498.13	11.4	488.00	[481.38 : 489.36]	0.23	0.21	0.08	0.02130	0.01843	0.01388
82.8403.551	877	501.96	13.25	497.21	[491.20 : 497.24]	0.28	0.22	0.04	0.02225	0.02557	0.01220
9.4641.568	1035	499.46	15.05	497.06	[501.96 : 507.96]	0.28	0.23	0.71	0.02085	0.02276	0.01336
9.4882.332	1070	496.53	15.24	505.57	[500.45 : 506.62]	0.29	0.22	0.22	0.02115	0.02286	0.01168
9.5239.505	997	499.72	14.05	510.25	[503.51 : 511.64]	0.3	0.24	0.09	0.01802	0.01892	0.01174
9.5484.258	1022	499.91	14.35	525.52	[511.15 : 522.02]	0.33	0.24	0.45	0.01891	0.01932	0.00955

[a] The characteristic timescale of the quasar light curve in days, calculated off of the last 100 points.

[b] The standard deviation of the characteristic timescale of the quasar light curve, calculated off of the last 100 points.

[c] The characteristic timescale of the final value of the quasar light curve in days.

[d] The conditional 95% confidence interval of the final characteristic timescale value.

[e] The standard deviation of the quasar flux variations, calculated off of the last 100 points.

[f] The standard deviation of the standard deviation of the quasar flux variations, calculated off of the last 100 points.

[g] The standard deviation of the quasar flux variations of the final value of the quasar light curve in days.

in Section 5.2. We are able to estimate the latent state accurately using all three filters with comparative RMSEs (from Table 5.2) PF RMSE = 0.03547, MCMC-PF RMSE = 0.03674 and APPF RMSE = 0.01587. The APPF provides a 55% improvement over the PF, a markedly significant performance increase. We can see how the APPF keeps track through the evolution of the series whilst the PF loses track on a number of occasions especially towards the beginning of the series. Analyzing the evolution of τ we can see it quickly jumps to 500 and fluctuates around that value with little deviation, adapting as and when necessary.

MACHO source 211.16703.311 exhibits far more erratic behavior. Observe the massive downward spikes in Figure 5.10 around 4.95x10^4 MJD and 5.1x10^4 MJD. All filters track the signal closely with the APPF (RMSE = 0.01190) showing a marked improvement over the PF (RMSE = 0.01801) and MCMC-PF (RMSE = 0.01906). Of note, we have more data points for this light curve compared to MACHO source 63.7365.151 (863 vs. 229) which is evident both in the evolution of the curve and in the evolution of τ in Figure 5.11. We can see it is extremely tightly coupled with a larger concentration around the long run mean of 500.

A similar period of turbulence, with a number of massive upward and downward spikes around 4.95x10^4 MJD, can be observed in MACHO source 1.4418.1930 in Figure 5.12. Across this period we can see the power of the IS diffusion in capturing spikes in the data. In estimation of this light curve, the APPF is statistically significantly more accurate with an RMSE = 0.01078 compared to PF RMSE = 0.04444 and MCMC-PF RMSE = 0.02392 exhibiting a 75% increase in performance. As with source 211.16703.311 we can see an extremely tight coupling and bounds in the evolution of τ in Figure 5.13. MACHO source 30.11301.499 is a sparsely observed light curve with only 296 data points. There are many periods of missing data within the series, easily seen by the sharpness of the evolution of the series in Figure 5.14. See for instance the two jagged troughs between 4.93x10^4 MJD and 4.94x10^4 MJD. This artifact is also visible in the evolution of the associated τ in Figure 5.15. Here again, the APPF (RMSE = 0.02564) significantly outperforms the PF (RMSE = 0.07567) and MCMC-PF (RMSE = 0.04284) in estimation accuracy providing a 66% increase in performance.

Across these four sample curves we have observed the advantage provided by the APPF in estimation of irregular quasar time series. For both sparse and densely sampled series, the APPF provides statistically significant increases in estimation accuracy when compared to the PF and MCMC-PF. It is able to adapt to changes in the latent state quicker and more accurately than the other filters building on the IS diffusion inherent to the original particle filtering recursions. These results are replicated across all quasar light curves we have estimated.

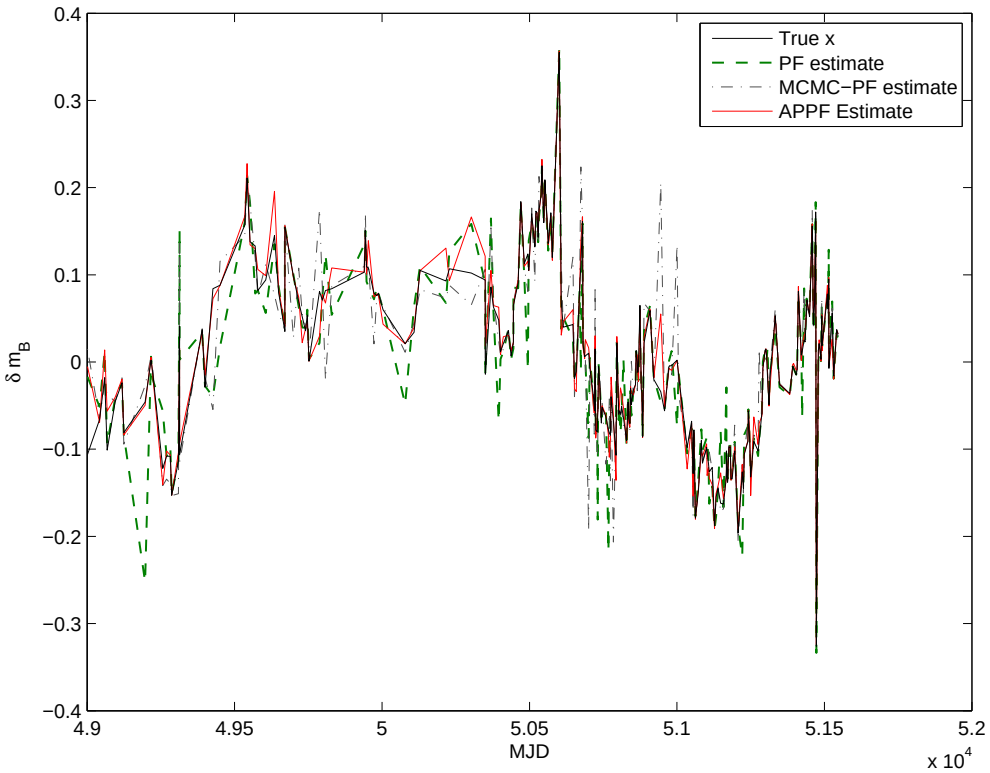

Figure 5.8: Source 63.7365.151 quasar light curve - filter estimates (posterior means) vs. true state PF (RMSE=0.03547), MCMC-PF (RMSE=0.03674) and APPF (RMSE=0.01587)

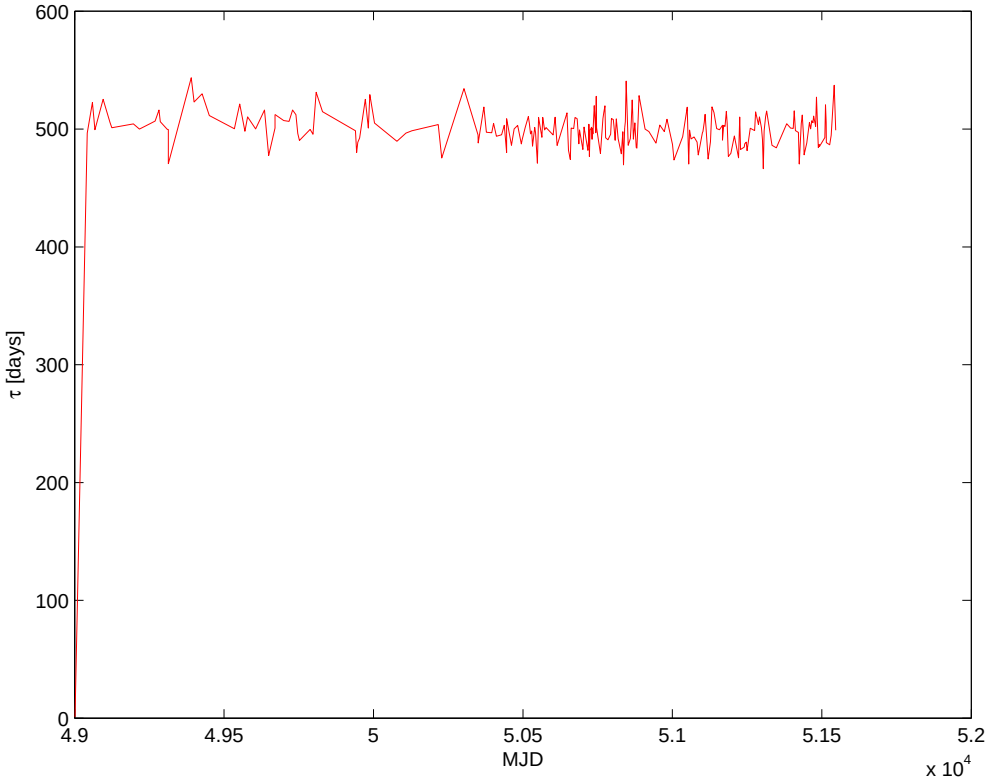

Figure 5.9: Source 63.7365.151 τ MAP estimates

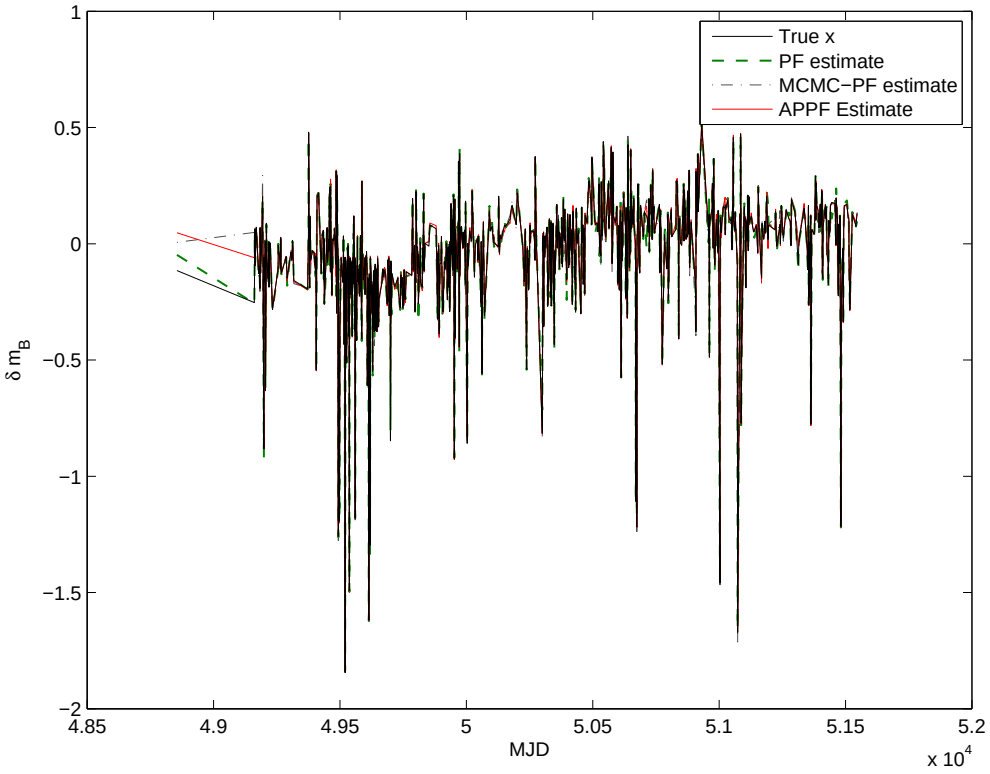

Figure 5.10: Source 211.16703.311 quasar light curve - filter estimates (posterior means) vs. true state PF (RMSE=0.01801), MCMC-PF (RMSE=0.01906) and APPF (RMSE=0.01190)

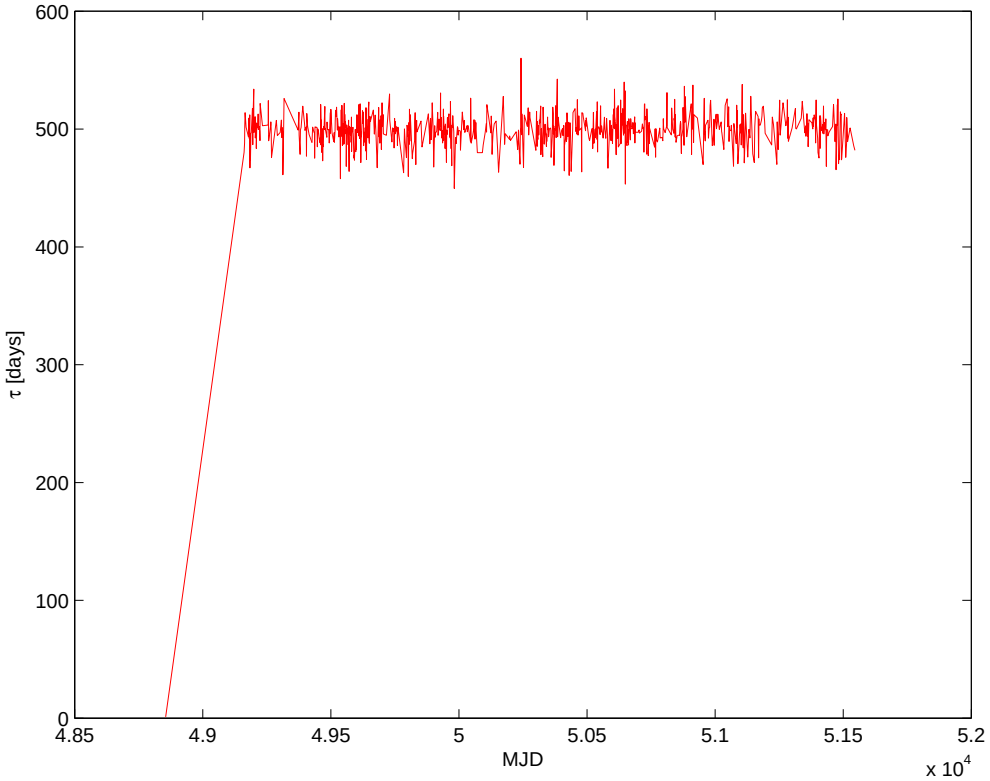

Figure 5.11: Source 211.16703.311 τ MAP estimates

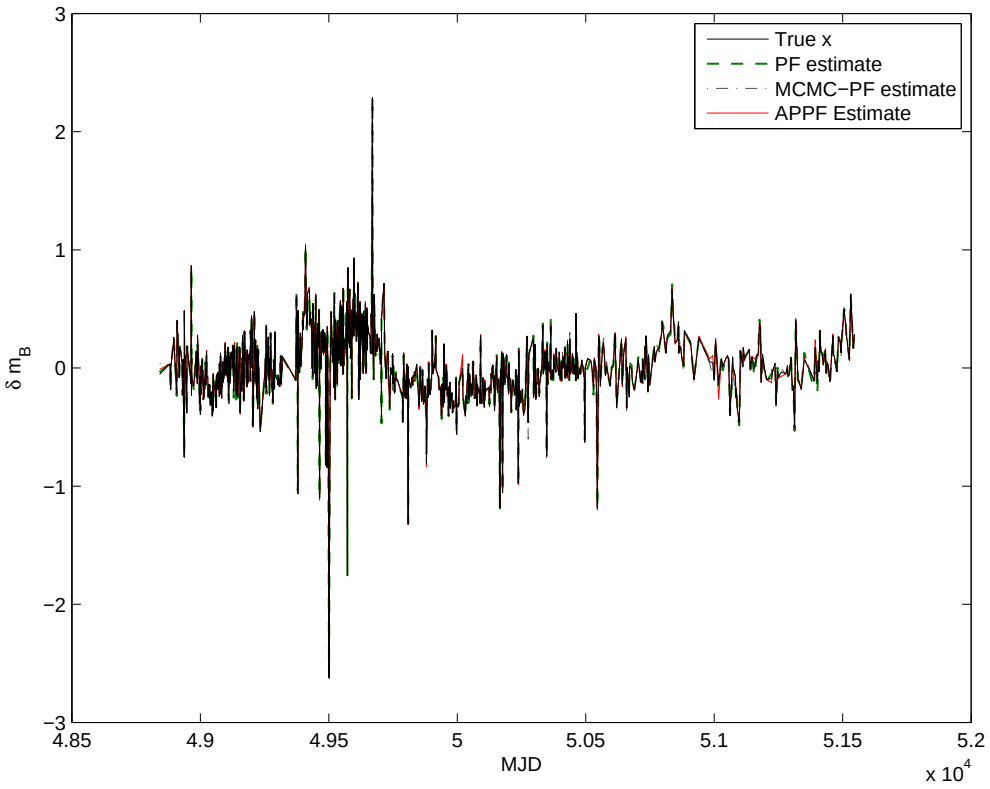

Figure 5.12: Source 1.4418.1930 quasar light curve - filter estimates (posterior means) vs. true state PF (RMSE=0.04444), MCMC-PF (RMSE=0.02392) and APPF (RMSE=0.01078)

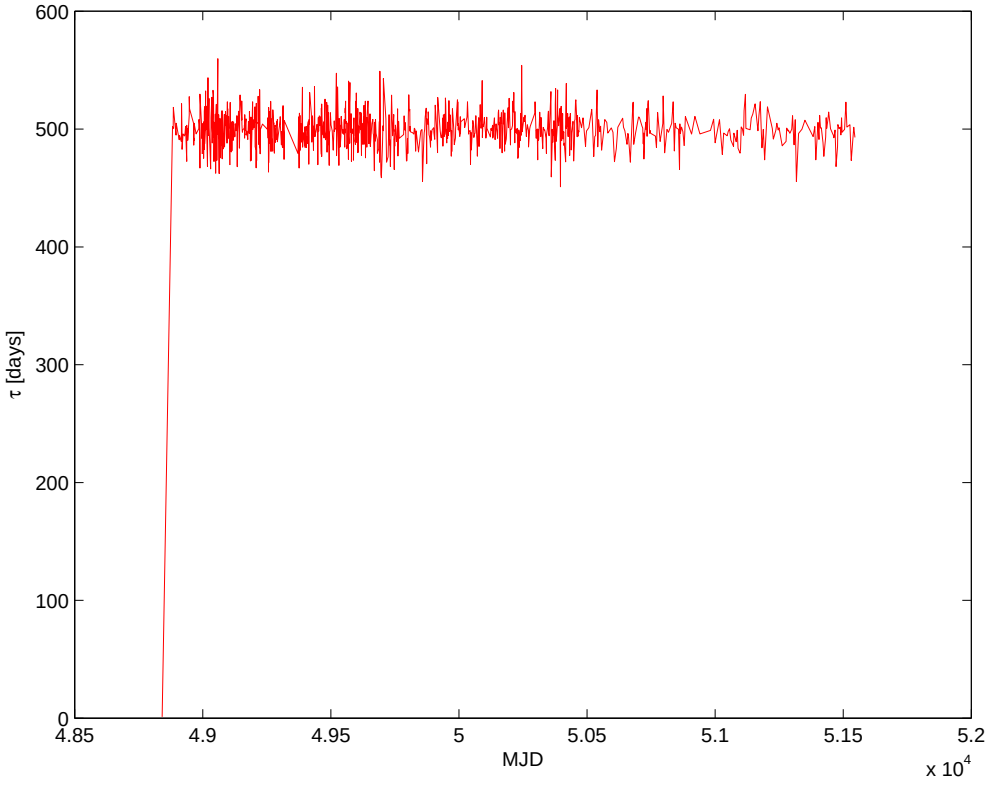

Figure 5.13: Source 1.4418.1930 τ MAP estimates

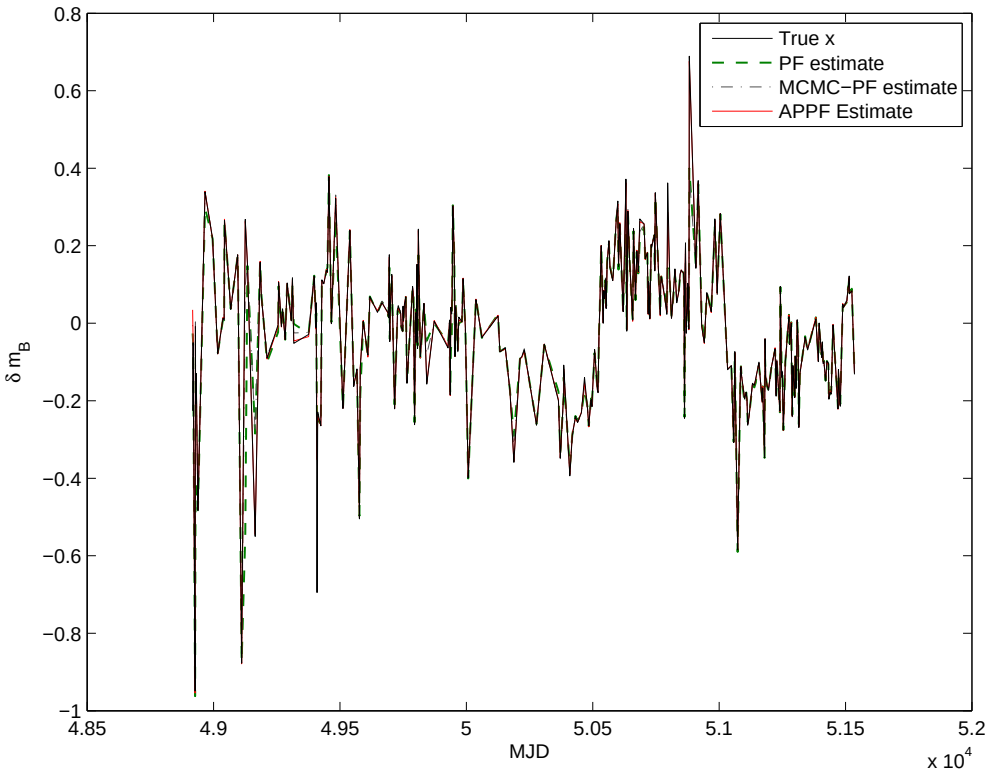

Figure 5.14: Source 30.11301.499 quasar light curve - filter estimates (posterior means) vs. true state PF (RMSE=0.07567), MCMC-PF (RMSE=0.04284) and APPF (RMSE=0.02564)

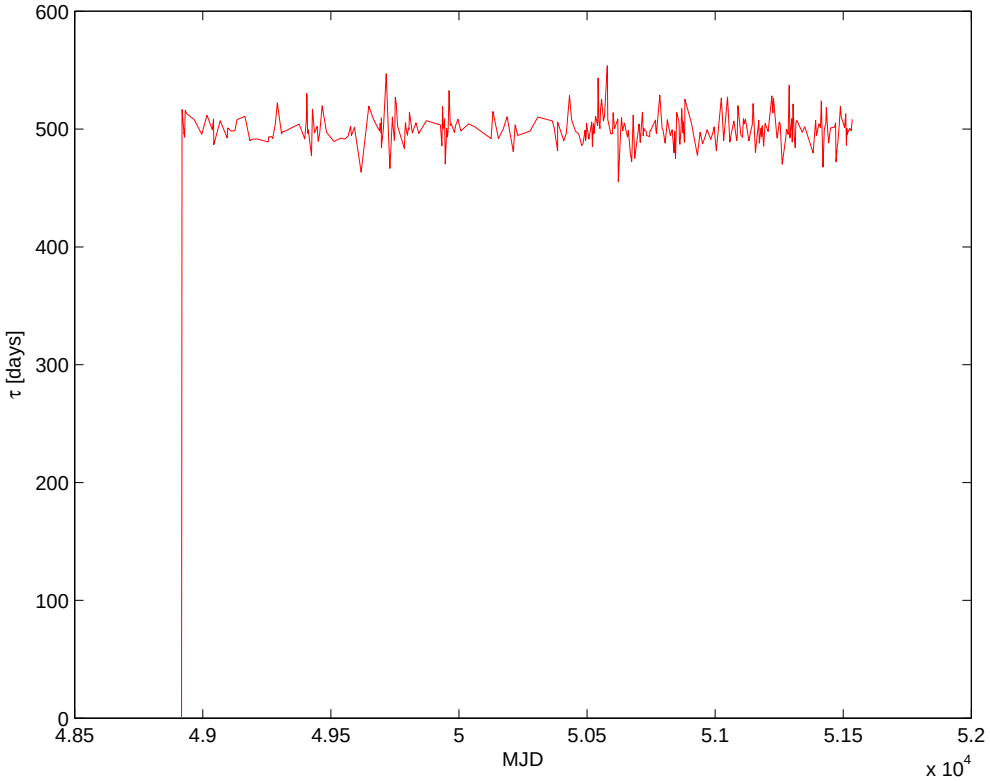

Figure 5.15: Source 30.11301.499 τ MAP estimates

Our MCMC MAP calibration of τ across the light curves shows a level of stability in contradiction to the literature. From Table 5.2, we can quite clearly see it oscillating around $\tau = 500$, with quite significant jumps from 0 to this rest level from the beginning of the calibration in the sample evolutions of τ provided in Figure 5.9, Figure 5.11, Figure 5.13 and Figure 5.15. From this it is self-evident that equivocating M_{BH} to τ, as the masses across the quasars vary, is not valid and is in essence indicative of a more unified AGN structure.

5.4 Summary

In this chapter we have presented a method for understanding the latent dynamics of astrophysical time series in real-time. Firstly we show that regularized astrophysical time series can be learned using sequential Monte Carlo methods through experiments based on autoregressive models. Secondly, we build on these results to model irregular astrophysical times series, successfully capturing the idiosyncrasies of 55 MACHO QSO light curves.

We calibrate the CAR(1) using an MCMC run by taking MAP estimates. The calibration shows τ to be first-order stable across the light curves which is at odds with current understandings of τ in the literature, leading us to discussions of τ's relation to x-rays in the galactic dust and towards a more unified AGN structure. Upon analysis, the evolution of τ is found to rapidly converge to a stable state which coupled with the efficacy of the fully Bayesian framework employed leads to extremely accurate state estimation, particularly with the APPF, excelling and outperforming the SIR PF and MCMC-PF.

Chapter 6

Discussion

In this chapter we discuss and evaluate our work in aid of highlighting what we have contributed to the field. We concisely answer our research objective in Section 6.1, providing detailed elaboration of our results in defining our conclusions. We thoroughly analyze our methods in Section 6.2. We articulate our findings into the body of knowledge in Section 6.3, highlighting significance and novelty, alongside a discussion on contributions. We conclude the chapter in Section 6.4.

6.1 Introduction

Embedding a computational intelligence step (a heuristic selection scheme) of adaptive path switching between generations based on maximal likelihood as a fitness function into the new APPF has yielded increased estimation accuracy for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems compared to contemporary filters. We have observed the APPF outperforming the SIR PF, MCMC-PF and PLA in estimation of the Heston stochastic volatility model for three stocks and three indices. In addition, we have observed the APPF being used for estimation of fifty-five quasar time series, outperforming both the SIR PF and MCMC-PF. Within and across both these domains the APPF provides statistically significant increases in estimation accuracy by addressing the weight degeneracy and sample impoverishment problem inherent to traditional sequential Monte Carlo methods.

Mathematical filtering is signal modeling and state inference given noisy observations. Wiener provided solutions for the stationary underlying distribution, with Kalman catering for non-stationary underlying distributions. Extensions of the KF try to overcome limitations of assumptions of linearity and Gaussianity but do not provide closed-form solutions to the distribution approximations required. As such the KF and its extensions have severe limitations when applied to complex distributions frequently encountered in real life.

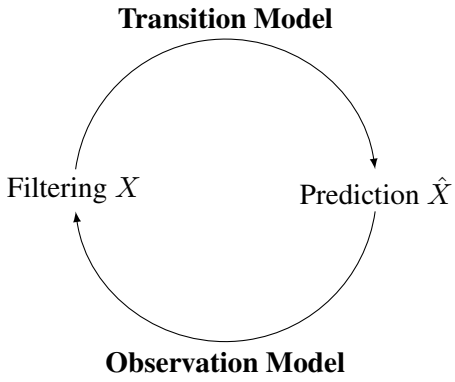

Figure 6.1: Recursive Bayesian Filtering

When filtering, we maintain an explicit representation of the current distribution over the state of the world. Filtering leverages state space models in a data processing algorithm to estimate latent state variables through the observation of large quantities of data. It focuses on removing observation errors and computing the pdf over the most recent state. This is an inverse learning problem: you want to find inputs as you are given outputs. Recursive Bayesian filtering, Figure 6.1, builds on the assumptions that the state follows a first-order Markov process and that the observations and states are independent to reason about beliefs under conditions of uncertainty. Given an initial understanding we predict (estimate) the latent state a priori by:

$$p(x_t|y_{0:t-1}) = \int p(x_t|x_{t-1}, y_{0:t-1})p(x_{t-1}|y_{0:t-1})dx_{t-1}$$

We update our understanding by using the prediction and new observation to obtain an a posteriori estimate:

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

Monte Carlo approximation using particle methods calculates this expectation of the pdf by sampling from the importance distribution π which is proportional to the true posterior at every point. We thus are able to sample sequentially i.i.d. draws, enabling us to estimate the state as in Equation (6.1). Given a prior understanding, particles are re-weighted as an expectation with respect to the dominating importance density.

$$\underbrace{\mathbb{E}_{p_\theta(x_0|y_{1:t})}[h(x_{0:t})]}_{\text{Expectation wrt Target Density}} \propto \int \underbrace{h(x_{0:t})}_{\text{Function we're interested in}} \cdot \underbrace{\frac{p_\theta(x_{0:t-1}|y_{1:t-1})}{\pi_\theta(x_{0:t-1}|y_{1:t-1})}}_{\text{Prior Weight}} \cdot \underbrace{\frac{f_\theta(x_t|x_{t-1}) \cdot g_\theta(y_t|x_t)}{\pi_\theta(x_t|x_{0:t-1}, y_{1:t})}}_{\text{Re-weight}} \cdot \underbrace{\pi_\theta(x_{0:t}|y_{1:t})dx_{0:t}}_{\text{Expectation wrt Dominating Density}} \tag{6.1}$$

Latent Transition Density — Observation Density — Marginal Importance Function

The primary problem with sequential Monte Carlo methods is weight degeneracy, where the particle system collapses onto a few non-zero points. The distribution of importance weights becomes more and more skewed over time with fewer and fewer non-zero importance weights. This is traditionally tackled with resampling the importance weights however this leads to a lack of diversity amongst particles, where the resultant sample has many repeated points. Traditional sequential Monte Carlo methods attempt to address this problem however introduce further data processing problems, which results in minimal to comparable performance improvements over the SIR PF. There have been a number of successful attempts to address this problem by hybridizing the original SIR algorithm with ideas from EC and GA, with application to a variety of fields and problems.

A new computational intelligence sequential Monte Carlo method, the APPF, is proposed for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems. The APPF is a recombinatory evolutionary algorithm which embeds a generation based adaptive particle switching step into the weight update. This step codifies the objective function as a maximal likelihood weight update fitness assessment through inter-generational competition. Initialization follows from SIR however upon IS we draw from both the current particle set and the past resampled out particle set. These two particle sets are recombined using a maximal likelihood weight update. This update follows from the fundamental reasoning of Monte Carlo methods: the APPF converges by a central limit theorem onto an invariant and thus the correct distribution. The core idea of the APPF is to leverage the descriptive ability of naively discarded particles in an adaptive evolutionary environment.

Initial tests were performed on a synthetic scalar estimation problem and on the univariate log-stochastic volatility model. The scalar estimation problem tackled is a common benchmark problem in sequential Monte Carlo literature. This problem has severe non-linearity in both the system and measurement equations with additive Gamma noise in the process equation. The observation model is non-stationary, changing at $t = 30$. Across 100 runs there was a marked and statistically significant increase in performance by the APPF in comparison to the PF and MCMC-PF, providing a 40% increase in performance. This result is interesting in firstly, displaying the theorized performance increase however, secondly, and more importantly we are able to analyze some key empirical features of the APPF. Though we are some way off perfect tracking, the APPF reacts and adapts quicker to spikes in the signal, especially when the generating distribution changes. This key adaptive feature is further observed in our univariate log-stochastic volatility model experiment. Assessed over 100 runs, there is a notable increase

in performance by the APPF in comparison to the PF and MCMC-PF. Closer examination shows the APPF converging to the generating signal quicker than the comparison filters.

This key feature led us in naming our novel filter the APPF. It is patent that the adaptive particle switching step, this being the only difference between it and SIR, is addressing the weight degeneracy and sample impoverishment problem. This step enables the APPF to switch between competing particles to best understand the current signal, addressing weight degeneracy. In addition this step coupled with resampling maintains diversity amongst the particle system. This is self-evident from the results observed.

6.1.1 Stochastic Volatility Estimation

Estimation of the Heston stochastic volatility model built on our preliminary results. Three indices and three common stocks (3 years' daily closing prices) were calibrated using an MCMC and estimated using the PF, MCMC-PF, PLA and APPF. The APPF outperformed all comparison filters in all SV estimation experiments. By taking the MAP estimate from our calibration of the models, the parameter estimation problem is reduced to a deterministic 2-dimensional control function, enabling us to marginalize out parameter estimation effects on state estimation. The results from state estimation are thus easily compared and benchmarked against one another as deviations from the MCMC calibration.

In estimation of the three indices we find the APPF providing a 4% - 6% increase in performance accuracy over the SIR PF. In estimation of the three common stocks we find the APPF providing a 4% - 16% increase in performance accuracy over the SIR PF. A zoomed window of $t = 0$ - $t = 160$ for GE SV estimation in Figure 4.17 is provided in Figure 6.2. We observe the APPF converging onto the latent signal quicker than the other PFs. See the variation in filter performance through $t = 0$ - $t = 5$. Thereafter all filters either lag or over emphasize changes in the latent signal though the APPF provides most stability in this respect. In addition filters other than the APPF frequently lag and overshoot the latent signal. This leads to the observed increase in performance of the APPF. This result is prototypical of the SV estimation results across all securities.

As observed in our preliminary results, the APPF significantly outperforms the other filters through the race-to-convergence. The APPF is more accurate in its response to changes in the latent state and subsequent tracking. When testing against influences of particle size, there is no notable increase in filter performance for the APPF, though there is a small increase in performance for the PF, MCMC-PF and PLA in congruence with literature. This result displays the robustness of the APPF which we find attributable to the APPF maintaining a population of particles which when diffused through IS and selected through inter-generational competition,

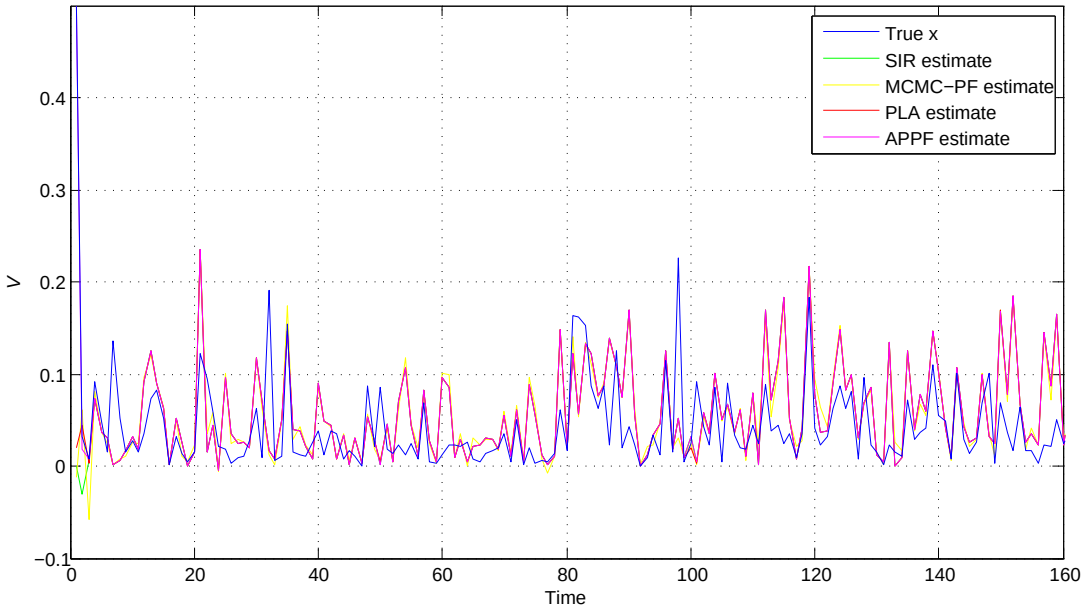

Figure 6.2: Heston model estimates for GE - filter estimates (posterior means) vs. true state: zoom $t = 0 - t = 160$

selects informative particles from pertinent areas of the search space. This is the cause of increase in performance observed in the PF, MCMC-PF and PLA. More particles allow the IS diffusion to search a wider area in the search space.

6.1.2 Astrophysical Time Series Analysis

Estimating astrophysical time series using sequential Monte Carlo methods buttressed previous results. On regularized time series the APPF outperformed both the PF and the MCMC-PF, providing statistically significant increases in estimation accuracy. Over 100 runs for the AR(1) model the APPF provided a 56% increase in estimation accuracy, a 40% increase for the AR(2) model, a 40% increase for the ARIMA(1,1,0) model and a 25% increase for the ARIMA(2,1,0) model.

Adding additional autoregressive terms self-evidently allows us to further specify the latent dynamics. Trivial increases in the autoregressive order leads to overfitting of the data and needs to be carefully monitored. This increased performance of the PF and MCMC-PF however does not increase performance of the APPF. Positing stationarity within the series again increases performance of the PF and MCMC-PF, without significantly affecting APPF performance. From this it is clear that the APPF is providing robust estimates without further specification of the underlying astrophysical dynamics. This can be seen quite clearly in Figure 6.3 which provides a zoom onto $t = 100$ - $t = 200$ on the sample runs found in Figure 5.3 and Figure 5.4.

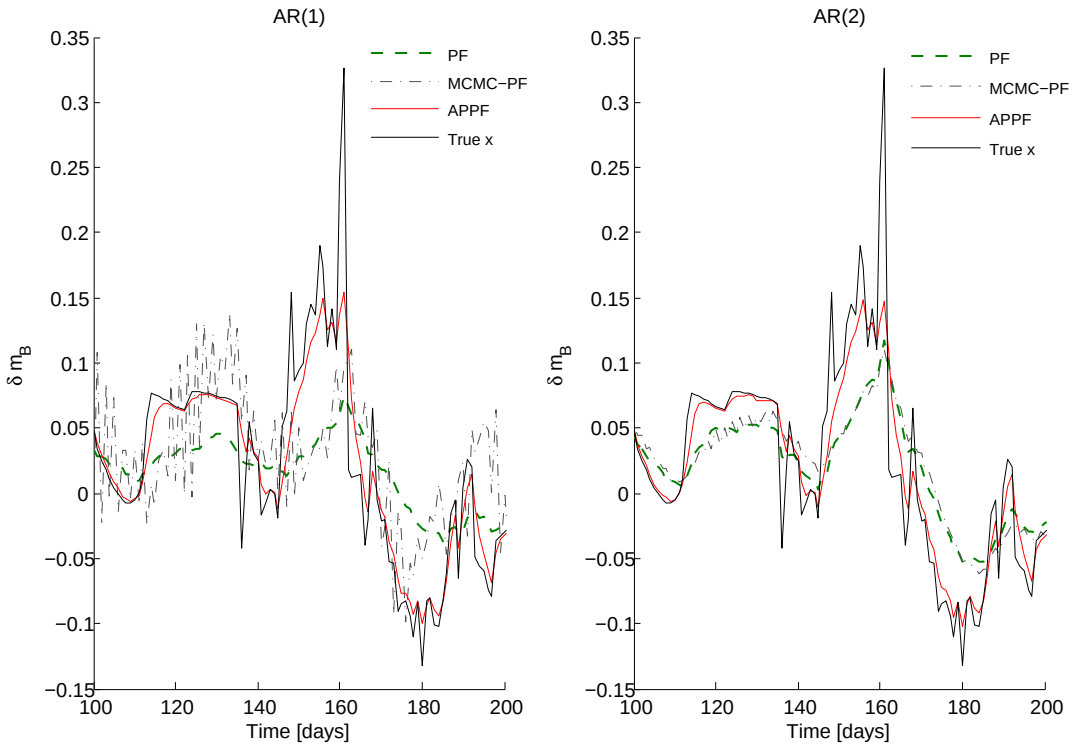

Figure 6.3: Regularized source 63.7365.151 quasar light curve AR(1) & AR(2) - filter estimates (posterior means) vs. true state: zoom $t = 100 - t = 200$

Firstly, note how accurately the APPF tracks the true latent state. It is able to pick up and adapt to changes in the underlying signal quicker and more accurately than both the PF and MCMC-PF. For AR(1) the MCMC-PF is very erratic: it picks up the broader trend but with a high level of variance (2 orders of magnitude higher than both the PF and APPF - see Table 5.1). Here, the PF is far off tracking the signal closely. Adding an autoregressive term and positing AR(2) dynamics results in minimal change in APPF performance however observe the tightening of variance in the MCMC-PF with both it and the PF improving significantly in their track of the latent state. Tracing through, the PF and MCMC-PF seem to be providing lagged estimates to the true state. The APPF is able to both track the state more accurately and adapt to changing dynamics whilst not over specifying the problem.

Fifty-five MACHO B-band light curves were modeled using the CAR(1) model after some pre-processing to remove bad measurements. The APPF outperformed both the PF and MCMC-PF in estimation of every light curve. The percentage increase in performance accuracy can be seen in the histogram and complementary cumulative distribution function in Figure 6.4. Interestingly the density of increase in estimation accuracy is skewed with mean at 40% and a fat left tail (indicating a larger tendency to significantly outperform). In addition we can see from the complementary cumulative distribution function that for ∼80% we get at least a 35%

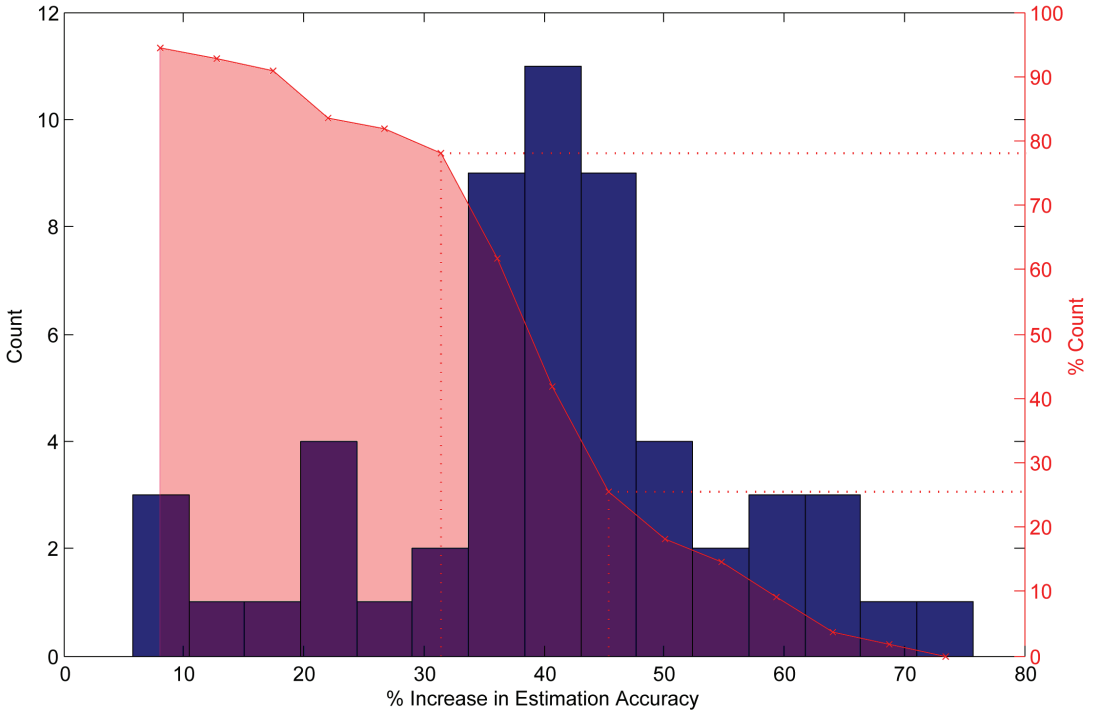

Figure 6.4: CAR(1) model of 55 MACHO light curves increase in APPF estimation accuracy with respect to SIR PF - histogram and complementary cumulative distribution function. There is a large tendency for a 40% increase in estimation accuracy (where the histogram is centered), with ∼80% gaining 35% and ∼30% gaining at least 45%.

increase in estimation accuracy, whilst ∼30% are gaining at least a 45% advantage. The APPF has significantly outperformed the PF in estimation accuracy, with the adaptive path switching step being the only change between the APPF and the SIR PF making patent our hypothesis that embedding a computational intelligence step of adaptive path switching between generations based on maximal likelihood as a fitness function into the new APPF yields increased estimation accuracy for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems compared to contemporary filters.

6.2 Methods and Performance Analysis

A representation of the posterior distribution for parameter estimation was built using MCMC simulation techniques. This is the modus operandi for calibrating stochastic volatility models (Sandmann & Koopman 1998). Shephard & Pitt (1997), Durbin & Koopman (1997) and Durbin & Koopman (2000) have designed methods for constructing likelihoods of general state space models using Monte Carlo methods which were shown to be efficient implementations for standard linear SV models and for non-linear extensions by Sandmann & Koopman (1998). The key feature of the Monte Carlo method is the formulation of the SV as a linear dynamical

system enabling powerful filtering and smoothing algorithms to be used and to draw on the vast spectrum of research on structural time series models (Harvey 1991).

There are a number of advantages to using such adapted Monte Carlo methods (Sandmann & Koopman 1998). Firstly, computational effort is reduced dramatically when compared to naive Monte Carlo grid search space exploration, while attaining finite sample efficiency. Secondly, sampling variation is reduced to give arbitrarily close approximations to the posited true likelihood density. Thirdly, standard likelihood tests can be used for comparison to other methods. And finally, a number of increasingly complex extensions can be posited with minimal changes to the estimation procedure as the state space form is retained. Several well-known and widely used extensions of the basic SV model can thus be treated. Leveraging this retention of state space form, calibration of the CAR(1) model for astrophysical time series analysis was performed using an MCMC.

There are a number of criticisms of using MAP estimates. Primarily, it is argued that MAP is not very representative of Bayesian methods. This criticism is false by assertion: Bayesian methods summarize data by characteristic distributions from which either the median or mean is taken, and in the case of MAP the mode is taken. In addition, Sorenson (1980) has shown that taking posterior median values does not differ significantly from a maximum likelihood fit. Secondly, it is argued that where mixture models are used the posterior may be multi-modal which by taking MAP would necessitate finding the highest mode. This criticism is more of a computational challenge which is overcome by using higher-dimensional histogram analysis as can be seen in the contour plot in Figure 5.7.

The calibration of τ across the astrophysical time series in Section 5.3 (details in Table 5.2) shows a level of stability in direct contradiction with literature. The value of τ is first-order stable and coupled across all quasars (note the characteristic timescale of the quasar light curve calculated off the last 100 points and the characteristic timescale of the final value of the quasar light curve), with a tightly bounded variance (note 95% CI of final characteristic timescale values) breaking from the understanding that τ is related to M_{BH} as posited by Kelly et al. (2009). Recent postulations in the astrophysics community suggest a more unified structure underlying accretion powered sources (Kazanas et al. 2012): the characteristic timescale τ corresponds to the distance between x-rays and the galactic dust in the accretion disk (as related through the x-ray absorptions features of the QSO spectra) and should not, as indicated by the data, therefore have any correlation to the M_{BH}: a thesis our results corroborate.

Calibration of the regularized astrophysical time series models followed the Box-Jenkins methodology. This is a parsimonious modeling technique which provides robust forecasts

(Hamilton 1994). The methodology is fully defined in Section 5.2. Estimates of the autoregressive parameters were provided by the Burg method which fits the input data by minimizing the least squares fit to forward and backward prediction errors with constraints satisfying the Levinson-Durbin recursion, which calculates solution to equations involving Toeplitz matrices (Kay 1988). Empirically, the Burg method was found to provide stable estimates in comparison to the method-of-moments calculations through the Yule-Walker equations.

Given parameter estimates, state estimation performance was assessed using RMSEs. The RMSE is lower-bounded by the CRLB which is a computationally intensive calculation of estimation accuracy. Furthermore, Quang et al. (2010) have shown the RMSE converges under an upper bound which does not show explicit dependency on the dimension of the hidden state. This result enables assignment of the RMSE as the MC error in discrete and continuous dynamical systems, the ultimate benchmark which is to be minimized. Hendeby (2008) argue that RMSE evaluation only captures one aspect of filter performance as distribution estimates differ. They advocate the use of Kullback-Leibler (KL) divergence to capture the true difference between distributions however, KL divergence can only be used in a simulation setting. Working on real data institutes using the prior as a substitute for the real distribution from which we diverge. As such entropy measures such as KL divergence, though enticing theoretically, have limited practical application.

Preliminary results indicated towards the performance increase achieved by the APPF in comparison to both the PF and MCMC-PF. These were built built upon by taking a number of representative securities, benchmarking them through an MCMC calibration of a common SV model and estimated using sequential Monte Carlos. RMSE tracking comparison found the APPF to offer statistically significant increase in estimation accuracy compared to the PF, MCMC-PF and PLA. Subsequently, the APPF was applied to astrophysical time series were fifty-five light curves were modeled both as regularized and irregular series. The APPF excelled in comparison to the PF and MCMC-PF in both scenarios, again providing statistically significant improvements in estimation accuracy. Across both problem domains the APPF excelled in providing robust and accurate estimates of the latent state.

Whilst not the primary concern of this work, there is a body of research on the use of evolutionary computation for parameter estimation (Pantrigo & Sanchez 2005, Han et al. 2011, Park et al. 2007, Kwok et al. 2005, Li & Honglei 2011, Duan & Cai 2008, Yang et al. 2010, Zhang et al. 2010, Zheng & Meng 2008) which has been reviewed in Section 2.4. The assertions made here are sensitive to parameter settings. As such, benchmarking methods i.e. MCMC, need to be undertaken carefully to ensure no extraneous factors affect final results. The results

have focused primarily on state estimation on two distinct datasets from which we have drawn firm conclusions as to the efficacy of computational intelligence synergism in sequential Monte Carlo methods in comparison to the PF, MCMC-PF and PLA.

6.3 Significance & Contributions

An effective new sequential Monte Carlo method, the APPF, is realized for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems, to achieve increased estimation accuracy through adapting to changes in the latent process and addressing the weight degeneracy and sample impoverishment problem within sequential Monte Carlo methods. As postulated, embedding a computational intelligence step of adaptive path switching between generations based on maximal likelihood as a fitness function into the new APPF, has yielded increased estimation accuracy compared to contemporary filters.

Computational intelligence techniques have long been applied to computational finance though stochastic volatility estimation has never previously been accomplished through the synergy between computational intelligence and sequential Monte Carlo methods. By building on work from within both the sequential Monte Carlo and computational intelligence community we were able to address a major problem with sequential Monte Carlo methods. The APPF was successfully applied to the stochastic volatility estimation problem where it outperformed a number of contemporary filters.

Sequential Monte Carlo methods, to the best of our knowledge, have not been applied to modeling astrophysical time series. We pioneered research into the application of sequential Monte Carlo methods to astrophysical time series analysis, in turn postulating latent dynamics and measuring estimation accuracy. Starting with regularized quasar time series we proposed a number of autoregressive and autoregressive integrated moving average models to describe the latent dynamics. Upon recursive Bayesian estimation using sequential Monte Carlo methods, we found the APPF to outperform the PF and MCMC-PF, providing robust results with generalized dynamical system dynamics. Building on this result, we attempted to model irregular astrophysical time series using the CAR(1) model. Fifty-five quasar time series were calibrated using MCMC, after which they were estimated using sequential Monte Carlos. In all cases we found the APPF outperforming the PF and MCMC-PF providing statistically significant results. In addition, we found τ - the characteristic timescale of the quasar in days - to be first-order stable in contradiction to literature and in line with current postulations within the astrophysics community.

This thesis provides a novel sequential Monte Carlo method which leverages a computational intelligence step of adaptive path switching between generations based on maximal likelihood as a fitness function to yield enhanced estimation accuracy for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems compared to contemporary filters, and an assessment of the efficacy of the use of sequential Monte Carlo methods for modeling astrophysical time series. Our novel method, the APPF, has been successfully applied to the stochastic volatility estimation problem, outperforming contemporary sequential Monte Carlo methods and to modeling astrophysical times series, again, outperforming contemporary filters. Admittedly there are many ideas which can be explored in the combinatorial space between computational intelligence (and more general metaheuristic techniques) and sequential Monte Carlo methods however we are looking at a particular problem - weight degeneracy and sample impoverishment - using a really specific technique in two distinct application domains with extremely positive results. There are an uncountably large number of modifications which could be made to sequential Monte Carlos which this work does not intend to address.

6.4 Summary

Recursive Bayesian filtering aims to maintain an explicit representation of the current distribution over the state of the world. It leverages state space models in a data processing algorithm to estimate latent state variables through observation. Sequential Monte Carlo approximation suffers from a number of problems, primary of which is the weight degeneracy and sample impoverishment problem. We have described a novel sequential Monte Carlo method which leverages ideas and concepts from computational intelligence to tackle the weight degeneracy and sample impoverishment problem.

Preliminary results on a scalar estimation problem and the log-stochastic volatility estimation problem shows the APPF outperforming contemporary particle filters. Application into two distinct problem domains shows the APPF statistically significantly outperforming contemporary particle filters. We have examined our methodology and have provided reasoning and justification for the process model we have used, elaborating and discussing apparent discrepancies, range of applicability and breadth of validity. We conclude by discussing the significance and novelty of our work, whilst setting out our contributions.

Chapter 7

Conclusions and Future Work

The main objective of this thesis was to address the weight degeneracy and sample impoverishment problem in sequential Monte Carlo methods. It was proposed to be addressed by embedding a computational intelligence step, a heuristic selection scheme, of adaptive path switching between generations based on maximal likelihood as a fitness function into a new adaptive path particle filter (APPF). It was posited that our APPF will yield increased accuracy for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems compared to contemporary filters. Preliminary tests on a scalar estimation problem and the log-stochastic volatility model proved that the APPF was outperforming contemporary sequential Monte Carlo methods. Further tested on stochastic volatility estimation and modeling of astrophysical time series, the APPF statistically significantly outperformed contemporary sequential Monte Carlo methods, providing robust estimates with enhanced recursive Bayesian estimation accuracy and an admissible increase in computational effort.

Recursive Bayesian estimation accuracy has been increased using a heuristic selection scheme which addresses a key problem within sequential Monte Carlo methods and applied into two distinct application domains with extremely positive results. This thesis has shown the advantages of such a synthesis. This development naturally lends itself to applications to non-linear non-Gaussian dynamical systems in general, offering further avenues for refinement along the way. At the level of the representation of the probability distribution, the APPF is using a fitness-based recombination of a past representation with a current representation and owing to the Markov nature of the state process draws upon the body of evolutionary computation experience and theory. The observed gains in estimation accuracy are directly related to evolutionary computation theory and practice and are justified through the fundamental reasoning of Monte Carlo methods - their convergence by a central limit theorem onto an invariant and thus the correct distribution by the law of large numbers. Owing to this relation the APPF can be further built upon using ideas from computational intelligence.

This novel approach could prove to be of great commercial and proprietary import, in aiding accurate pricing of derivative products and both proprietary and market-making activities. Here, the smallest increase in modeling and estimation accuracy affords real competitive advantages and capitalization opportunities. Furthermore, the approach has the potential to be used in further understanding the astrophysical dynamics of quasi-stellar radio objects and in classification models of stellar object type. Our work in this field pioneered the use of sequential Monte Carlo methods for modeling astrophysical time series. It has helped in both highlighting the efficacy of their use in astrophysical time series analysis but also in a wider understanding of the astrophysical dynamics of such time series.

7.1 Contributions

This thesis provides a novel sequential Monte Carlo method which leverages a computational intelligence step of adaptive path switching between generations based on maximal likelihood as a fitness function to yield enhanced estimation accuracy for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems compared to contemporary filters, and an assessment of the efficacy of the use of sequential Monte Carlo methods for modeling astrophysical time series.

This thesis makes the following contributions:

1. The development of a new sequential Monte Carlo method based on computational intelligence for recursive Bayesian estimation of non-linear non-Gaussian dynamical systems.
    (a) Outperformed contemporary filters in a scalar estimation problem and the univariate log-stochastic volatility estimation problem.
    (b) Successfully addresses weight degeneracy and sample impoverishment problem of traditional sequential Monte Carlo methods.
2. The application of a new sequential Monte Carlo method to the stochastic volatility problem.
    (a) Calibrated on Heston stochastic volatility model.
    (b) Outperformed contemporary filters in estimation of six securities.
3. The pioneering application of sequential Monte Carlo methods to astrophysical time series analysis.
    (a) Postulated latent dynamics of regularized and irregular quasar time series.

(b) Our novel sequential Monte Carlo method outperformed contemporary filters in estimation of regularized quasar time series.

(c) Our novel sequential Monte Carlo method outperformed contemporary filters in estimation of irregular quasar time series.

i. Calibrated CAR(1) model on fifty-five quasar time series.

ii. Found the characteristic timescale τ of quasars to be first-order stable.

7.2 Future Work

We have successfully displayed the enhanced estimation accuracy from the heuristic selection scheme in the APPF. There are a number of interesting avenues for future exploration, detailed below, which provide ideas for broadening our methodology and its application. These shall help further refine our results, allowing focus on both theoretical and empirical aspects.

7.2.1 Dual Estimation & Particle Learning

Our work has focused on state estimation using an MCMC calibration to drive the sequential Monte Carlos. This has enabled us to focus efforts on refining our algorithm for this purpose, marginalizing the parameter estimation problem. It would be interesting to inspect estimation performance when we combine state and parameter estimation as the dual estimation problem (also referred to as particle learning in the literature). This could, similarly as above, be benchmarked off an MCMC calibration for comparison.

The dual estimation problem is a prohibitively hard problem with an active research community (Doucet & Johansen 2008, Saha 2009). First proposed by Berzuini et al. (1997), a common technique is to include the parameters as part of the state vector. Particle filters perform poorly when the dimension increases (Bengtsson, et al. 2008, Crisan & Doucet 2002, Daum & Huang 2003, Snyder, et al. 2008), and as such any such amalgamation of parameters into the particle system, leads to a slower exploration of the search space. This added computational cost is a severe hindrance to the efficacy of particle learning in practice. It would be interesting to inspect the effectiveness of the computational intelligence step introduced into SIR as the APPF in overcoming learning difficulties in particle learning.

7.2.2 Convergence Analysis

The generation gap inspired adaptive particle switching step introduces a direct, fitness based (specifically, descriptive power based) competition between generational elements into the particle filter, creating a more evolutionary computation like algorithm, which is yielding superior

results. It also suggests advancing particle filters in other ways that are inspired by genetic algorithm theory and practice.

For instance, generation gap methods, as the ones adopted in the APPF, increase selective pressure which can potentially lead to premature convergence: a problem analogous to the very one we are addressing in sequential Monte Carlos. However, there are various schemes in evolutionary computation i.e. fitness sharing (Arulampalam et al. 2002, Goldberg & Richardson 1987), crowding (Mengshoel & Goldberg 2008, Eiben & Smith 2008), niche specialization (Ashlock 2006) and triggered hypermutation (Morrison & De Jong 2000) which aim to preserve diversity in populations and across generations providing further avenues of potential enhancement of the APPF. Within the framework outlined in the APPF, we can draw on past genetic algorithm investigations of these techniques which should fully bring to bear the theoretical advantages of genetic algorithms and evolutionary computation in particle filters.

7.2.3 Derivative Pricing and Systematic Volatility Trading

Typically, stochastic volatility estimation is used in derivative pricing. The advantages of the APPF over contemporary filters can be used to more accurately price derivatives. However, this is not the only avenue available to capitalize on accurate forward knowledge of volatility. Volatility trading strategies have been around for some time though very little effort has been placed in investigating systematic trading models for volatility arbitrage (where quoted volatility and our estimates provide an arbitrage opportunity). A simple such strategy would look at bounded ranges of futures positions to maximize on upside and minimize downside risk. Similarly one could enact vanilla and exotic trading strategies using long/short straddles, long/short strangles, ratio call/put spreads and call/put ratio backspreads.

7.2.4 Predictive Power for Astrophysical Time Series Analysis

Our work in astrophysical time series analysis, and specifically the estimation accuracy achieved leads us to ponder the predictive power of the posited astrophysical dynamics and associated sequential Monte Carlo methods. Traditionally, sequential Monte Carlo methods are used for one-step ahead prediction though it would be interesting to assess the predictive-decay in comparison to classical astrophysical time series analysis techniques. For instance we could compare the tracking degeneracy between static and dynamic τ's. Furthermore, this analysis would feed into our discussion of the latent dynamics and for instance could include an exogenous noise term in the observation equation. Such an extension would enable us to investigate effects away and aside from the state dynamics currently under consideration, considering more accurate dynamics.

Bibliography

C. Alcock, et al. (2000). 'The MACHO Project: Microlensing Results from 5.7 Years of Large Magellanic Cloud Observations'. *The Astrophysical Journal* **542**:281–307.

L. Andersen (2008). 'Simple and efficient simulation of the Heston stochastic volatility model'. *Journal of Computational Finance* **11**(3):42.

B. Anderson & J. Moore (1979). 'Optimal filtering'. *Prentice-Hall Information and System Sciences Series, Englewood Cliffs: Prentice-Hall, 1979* **1**.

M. Arulampalam, et al. (2002). 'A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking'. *Signal Processing, IEEE Transactions on* **50**(2):174–188.

D. Ashlock (2006). *Evolutionary computation for modeling and optimization*, vol. 200. Springer-Verlag New York Inc.

L. Bauwens, et al. (1999). *Bayesian inference in dynamic econometric models*. Oxford University Press, USA.

T. Bengtsson, et al. (2008). 'Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems'. *Probability and statistics: Essays in honor of David A. Freedman* **2**:316–334.

C. Berzuini, et al. (1997). 'Dynamic conditional independence models and Markov chain Monte Carlo methods'. *Journal of the American Statistical Association* **92**(440):1403–1412.

C. Berzuini & W. Gilks (2001). 'RESAMPLE-MOVE filtering with cross-model jumps'. *Sequential Monte Carlo Methods in Practice* pp. 117–138.

J. Besag (1974). 'Spatial interaction and the statistical analysis of lattice systems'. *Journal of the Royal Statistical Society. Series B (Methodological)* **36**(2):192–236.

E. Bonabeau, et al. (1999). *Swarm intelligence: from natural to artificial systems*. No. 1. Oxford University Press, USA.

T. A. Boroson (2002). 'Black hole mass and eddington ratio as drivers for the observable properties of radio-loud and radio-quiet QSOs'. *The Astrophysical Journal* **565**(1):78.

G. Box & G. Jenkins (1976). 'Time series analysis. Forecasting and control'. In *Holden-Day Series in Time Series Analysis, Revised ed., San Francisco: Holden-Day, 1976*, vol. 1.

P. Brockwell & R. Davis (2002). *Introduction to time series and forecasting*. Springer.

R. Cabido, et al. (2012). 'High performance memetic algorithm particle filter for multiple object tracking on modern GPUs'. *Soft Computing-A Fusion of Foundations, Methodologies and Applications* **16**(2):217–230.

S. Camazine, et al. (2003). *Self-organization in biological systems*. Princeton University Press.

J. Carpenter, et al. (1999). 'Improved particle filter for nonlinear problems'. In *Radar, Sonar and Navigation, IEE Proceedings-*, vol. 146, pp. 2–7. IET.

P. Carr & D. Madan (1999). 'Option valuation using the fast Fourier transform'. *Journal of Computational Finance* **2**(4):61–73.

Z. Chen (2003). 'Bayesian filtering: From Kalman filters to particle filters'. Tech. rep., and beyond. Technical report, Adaptive Systems Lab, McMaster University.

S. Chib, et al. (2002). 'Markov chain Monte Carlo methods for stochastic volatility models'. *Journal of Econometrics* **108**(2):281–316.

Y. Chien & K. Fu (1967). 'On Bayesian learning and stochastic approximation'. *Systems Science and Cybernetics, IEEE Transactions on* **3**(1):28–38.

N. Chopin (2004). 'Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference'. *The Annals of Statistics* **32**(6):2385–2411.

R. Cid Fernandes, et al. (2000). 'Quasar Variability in the Framework of Poissonian Models'. *The Astrophysical Journal* **544**:123–141.

T. Clapp (2000). 'Statistical methods for the processing of communications data'. *Signal Processing Group Department of Engineering St. Johns College, University of Cambridge* .

R. Cont (2001). 'Empirical properties of asset returns: stylized facts and statistical issues'. *Quantitative Finance* **1**(2):223–236.

D. Crisan & A. Doucet (2002). 'A survey of convergence results on particle filtering methods for practitioners'. *Signal Processing, IEEE Transactions on* **50**(3):736–746.

F. Daum & J. Huang (2003). 'Curse of dimensionality and particle filters'. In *Aerospace Conference, 2003. Proceedings. 2003 IEEE*, vol. 4, pp. 1979–1993. IEEE.

R. Douc & O. Cappé (2005). 'Comparison of resampling schemes for particle filtering'. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pp. 64–69. IEEE.

A. Doucet, et al. (2006). 'Efficient block sampling strategies for sequential Monte Carlo methods'. *Journal of Computational and Graphical Statistics* **15**(3):693–711.

A. Doucet, et al. (2001). *Sequential Monte Carlo methods in practice*. Springer Verlag.

A. Doucet & A. Johansen (2008). 'A tutorial on particle filtering and smoothing: Fifteen years later'.

Z. Duan & Z. Cai (2008). 'Evolutionary particle filter for robust simultaneous localization and map building with laser range finder'. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, vol. 1, pp. 443–447. IEEE.

J. Durbin & S. J. Koopman (1997). 'Monte Carlo maximum likelihood estimation for non-Gaussian state space models'. *Biometrika* **84**(3):669–684.

J. Durbin & S. J. Koopman (2000). 'Time series analysis of non-Gaussian observations based on state space models from both classical and Bayesian perspectives'. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **62**(1):3–56.

A. Eiben & J. Smith (2008). 'Introduction to evolutionary computing (Natural computing series)' .

P. Fearnhead (2004). 'Particle filters for mixture models with an unknown number of components'. *Statistics and Computing* **14**(1):11–21.

P. Fearnhead & P. Clifford (2003). 'Online inference for well-log data'. *Journal of the Royal Statistical Society* **65**:887–899.

J. Freitas, et al. (2000). 'Sequential Monte Carlo methods to train neural network models'. *Neural computation* **12**(4):955–993.

X. Ge, et al. (2000). 'Markov Chain Monte Carlo Calibration of Stochastic Volatility Models' .

S. Geman, et al. (1984). 'Gibbs distributions, and the Bayesian restoration of images'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(2):721–741.

M. Gen & L. Lin (2004). 'Multiobjective hybrid genetic algorithm for bicriteria network design problem'. In *The 8th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, pp. 73–82.

J. Geweke (1989). 'Bayesian inference in econometric models using Monte Carlo integration'. *Econometrica: Journal of the Econometric Society* pp. 1317–1339.

E. Ghysels, et al. (1996). *Stochastic volatility*. Université de Montréal, Centre de recherche et développement en économique.

W. Gilks & C. Berzuini (2001). 'Following a moving target - Monte Carlo inference for dynamic Bayesian models'. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(1):127–146.

W. Gilks & P. Wild (1992). 'Adaptive rejection sampling for Gibbs sampling'. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **41**(2):337–348.

D. Goldberg & J. Richardson (1987). 'Genetic algorithms with sharing for multimodal function optimization'. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pp. 41–49. L. Erlbaum Associates Inc.

D. E. Goldberg & J. H. Holland (1988). 'Genetic algorithms and machine learning'. *Machine Learning* **3**(2):95–99.

N. Gordon, et al. (1993). 'Novel approach to nonlinear/non-Gaussian Bayesian state estimation'. In *Radar and Signal Processing, IEE Proceedings F*, vol. 140, pp. 107–113. IET.

F. Gustafsson (2010). 'Particle filter theory and practice with positioning applications'. *Aerospace and Electronic Systems Magazine, IEEE* **25**(7):53–82.

F. Gustafsson, et al. (2002). 'Particle filters for positioning, navigation, and tracking'. *Signal Processing, IEEE Transactions on* **50**(2):425–437.

J. Hamilton (1994). 'Time series analysis' .

J. Hammersley & P. Clifford (1968). 'Markov fields on finite graphs and lattices' .

H. Han, et al. (2011). 'An evolutionary particle filter with the immune genetic algorithm for intelligent video target tracking'. *Computers & Mathematics with Applications* **62**(7):2685–2695.

A. C. Harvey (1991). *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press.

M. Hawkins (2004). 'Naked active galactic nuclei'. *Astronomy and Astrophysics* **424**(2):519–529.

S. Haykin (2001). 'Kalman Filtering and Neural Networks' .

G. Hendeby (2008). *Performance and implementation aspects of nonlinear filtering*. Ph.D. thesis, Södertörn University.

S. Heston (1993). 'A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options'. *Review of Financial Studies* **6**(2):327–343.

A. I., et al. (1997). 'QSO variability: probing the Starburst model'. *Monthly Notices of the Royal Astronomical Society* **286**.

H. Ishibuchi & T. Murata (1998). 'A multi-objective genetic local search algorithm and its application to flowshop scheduling'. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **28**(3):392–403.

E. Jacquier, et al. (2004). 'Bayesian analysis of stochastic volatility models with fat-tails and correlated errors'. *Journal of Econometrics* **122**(1):185–212.

A. Jaszkiewicz (2002). 'Genetic local search for multi-objective combinatorial optimization'. *European Journal of Operational Research* **137**(1):50–71.

Y. Jin & J. Branke (2005). 'Evolutionary optimization in uncertain environments-a survey'. *Evolutionary Computation, IEEE Transactions on* **9**(3):303–317.

M. Johannes & N. Polson (2009). 'MCMC methods for continuous-time financial econometrics'. *Handbook of Financial Econometrics* **2**:1–72.

T. Kawaguchi, et al. (1998). 'Optical variability in active galactic nuclei: starbursts or disk instabilities?'. *The Astrophysical Journal* **504**(2):671.

S. M. Kay (1988). 'Modern spectral estimation: Theory and application'.

D. Kazanas, et al. (2012). 'Toward a Unified AGN Structure'. *arXiv preprint arXiv:1206.5022* .

B. Kelly, et al. (2009). 'Are the variations in quasar optical flux driven by thermal fluctuations?'. *The Astrophysical Journal* **698**(1):895.

J. Kennedy & R. Eberhart (1995). 'Particle swarm optimization'. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948. IEEE.

G. Kitagawa (1996). 'Monte Carlo filter and smoother for non-Gaussian nonlinear state space models'. *Journal of computational and graphical statistics* **5**(1):1–25.

A. Klamargias, et al. (2008). 'Particle filtering with particle swarm optimization in systems with multiplicative noise'. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pp. 57–62. ACM.

A. Kong, et al. (1994). 'Sequential imputations and Bayesian missing data problems'. *Journal of the American Statistical Association* **89**(425):278–288.

J. Kotecha & P. Djuric (2003a). 'Gaussian particle filtering'. *Signal Processing, IEEE Transactions on* **51**(10):2592–2601.

J. Kotecha & P. Djuric (2003b). 'Gaussian sum particle filtering'. *Signal Processing, IEEE Transactions on* **51**(10):2602–2612.

N. Kwok, et al. (2005). 'Evolutionary particle filter: re-sampling from the genetic algorithm perspective'. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2935–2940. IEEE.

R. Lee (2004). 'Option pricing by transform methods: extensions, unification and error control'. *Journal of Computational Finance* **7**(3):51–86.

A. Lewis (2000). 'Option valuation under stochastic volatility'. *Option Valuation under Stochastic Volatility* .

C. Li & Q. Honglei (2011). 'Parallel genetic unscented particle filter algorithm'. *Chinese Journal of Electronics* **20**(4):755–760.

L. Lima & R. Krohling (2011). 'Particle Filter with Differential Evolution for Trajectory Tracking'. In A. Gaspar-Cunha, R. Takahashi, G. Schaefer, & L. Costa (eds.), *Soft Computing in Industrial Applications*, vol. 96 of *Advances in Intelligent and Soft Computing*, pp. 209–219. Springer Berlin Heidelberg.

A. Lipton (2002). 'The vol smile problem'. *Risk* **15**(2):61–66.

J. Liu & R. Chen (1998). 'Sequential Monte Carlo methods for dynamic systems'. *Journal of the American statistical association* **93**(443):1032–1044.

J. Liu & M. West (1999). *Combined parameter and state estimation in simulation-based filtering*. Institute of Statistics and Decision Sciences, Duke University.

R. Lord, et al. (2006). 'A comparison of biased simulation schemes for stochastic volatility models'. *Quantitative Finance* **10**(2):177–194.

S. MacEachern, et al. (1999). 'Sequential importance sampling for nonparametric Bayes models: The next generation'. *Canadian Journal of Statistics* **27**(2):251–267.

S. Maskell (2004). 'An introduction to particle filters'. *State Space and Unobserved Component Models, Theory and Applications, editors Harvey, AC and Koopman, SJ and Shephard, N* .

P. Maybeck (1979). *Stochastic models, estimation and control*, vol. 141. Academic press.

O. Mengshoel & D. Goldberg (2008). 'The crowding approach to niching in genetic algorithms'. *Evolutionary computation* **16**(3):315–354.

N. Metropolis, et al. (1953). 'Equation of state calculations by fast computing machines'. *The journal of chemical physics* **21**:1087.

R. Morrison & K. De Jong (2000). 'Triggered hypermutation revisited'. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 2, pp. 1025–1032. IEEE.

N. Nikolaev & E. Smirnov (2007). 'Stochastic Volatility Inference with Monte Carlo Filters'. *Wilmott Magazine, John Wiley and Sons, July* pp. 72–81.

J. Pantrigo & A. Sanchez (2005). 'Hybridizing particle filters and population-based metaheuristics for dynamic optimization problems'. In *Hybrid Intelligent Systems, 2005. HIS'05. Fifth International Conference on*, pp. 6–pp. IEEE.

J. J. Pantrigo, et al. (2011). 'Heuristic particle filter: applying abstraction techniques to the design of visual tracking algorithms'. *Expert Systems* **28**(1):49–69.

S. Park, et al. (2007). 'A new particle filter inspired by biological evolution: genetic filter'. *International Journal of Applied Science Engineering and Technology* **4**(1):459–463.

B. Peterson, et al. (2004). 'Central masses and broad-line region sizes of active galactic nuclei. II. A homogeneous analysis of a large reverberation-mapping database'. *The Astrophysical journal* **613**(2):682–699.

K. Pichara, et al. (2012). 'An improved quasar detection method in EROS-2 and MACHO LMC datasets'. *Monthly Notices of the Royal Astronomical Society* **401**(19).

P. Pinto, et al. (2005). 'Wasp swarm optimization of logistic systems'. *Adaptive and Natural Computing Algorithms* pp. 264–267.

M. Pitt & N. Shephard (1999). 'Filtering via simulation: Auxiliary particle filters'. *Journal of the American Statistical Association* **94**(446):590–599.

P. B. Quang, et al. (2010). 'An insight into the issue of dimensionality in particle filtering'. In *Information Fusion (FUSION), 2010 13th Conference on*, pp. 1–8. IEEE.

T. Runkler (2008). 'Wasp swarm optimization of the c-means clustering model'. *International Journal of Intelligent Systems* **23**(3):269–285.

S. Russell, et al. (2010). *Artificial intelligence: a modern approach*. Prentice hall.

S. Saha (2009). 'Topics in particle filtering and smoothing' .

G. Sandmann & S. J. Koopman (1998). 'Estimation of stochastic volatility models via Monte Carlo maximum likelihood'. *Journal of Econometrics* **87**(2):271–301.

J. Sarma & K. De Jong (2000). 'Generation gap methods'. *Evolutionary Computation: Basic algorithms and operators* **1**:205.

D. Sengupta & S. Kay (1989). 'Efficient estimation of parameters for non-Gaussian autoregressive processes'. *Acoustics, Speech and Signal Processing, IEEE Transactions on* **37**(6):785–794.

N. Shephard & M. K. Pitt (1997). 'Likelihood analysis of non-Gaussian measurement time series'. *Biometrika* **84**(3):653–667.

D. Sivia & J. Skilling (2006). 'Data analysis: a Bayesian tutorial' .

A. Smith & A. Gelfand (1992). 'Bayesian statistics without tears: a sampling-resampling perspective'. *American statistician* pp. 84–88.

R. E. Smith & M. S. Hussain (2012). 'Hybrid metaheuristic particle filters for stochastic volatility estimation'. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, pp. 1167–1174. ACM.

C. Snyder, et al. (2008). 'Obstacles to high-dimensional particle filtering'. *Monthly Weather Review* **136**(12):4629–4640.

H. Sorenson (1980). *Parameter estimation: principles and problems*. Control and systems theory. M. Dekker.

R. Storn & K. Price (1997). 'Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces'. *Journal of global optimization* **11**(4):341–359.

J. Stroud, et al. (2004). 'Practical filtering for stochastic volatility models'. *State space and unobserved component models: theory and applications* p. 236.

E. Talbi (2009). *Metaheuristics: from design to implementation*.

E.-G. Talbi, et al. (2001). 'A hybrid evolutionary approach for multicriteria optimization problems: Application to the flow shop'. In *Evolutionary Multi-Criterion Optimization*, pp. 416–428. Springer.

R. Tsay (2010). *Analysis of financial time series*. Wiley-Interscience.

K. Uosaki, et al. (2005). 'Nonlinear state estimation by evolution strategies based particle filters'. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 3, pp. 2102–2109. IEEE.

R. Van Der Merwe, et al. (2001). 'The unscented particle filter'. *Advances in Neural Information Processing Systems* pp. 584–590.

E. Wan & R. Van Der Merwe (2001). 'The Unscented Kalman Filter' **5**(2007):221–280.

Q. Wang, et al. (2006). 'Enhancing particle swarm optimization based particle filter tracker'. *Computational Intelligence* pp. 1216–1221.

Y. Wang & Y. Li (2010). 'Multi-Agent Co-Evolutionary Particle Filter for Robust Moving Object Tracking'. *Journal of Computational Information Systems* **6**(8):2511–2519.

P. Wilmott (2007). *Paul Wilmott on Quantitative Finance, 3 Volume Set*. Wiley.

Z. Xiaowei, et al. (2013). 'Object Tracking with an Evolutionary Particle Filter Based on Self-Adaptive Multi-Features Fusion'. *Int J Adv Robotic Sy* **10**(61).

X. Yang, et al. (2010). 'Particle swarm optimization particle filtering for dual estimation'. In *Intelligent Control and Information Processing (ICICIP), 2010 International Conference on*, pp. 184–187. IEEE.

J. H. Yoo, et al. (2012). 'Evolutionary particle filtering for sequential dependency learning from video data'. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pp. 1–8. IEEE.

J. Zhang, et al. (2011). 'A Parallel Hybrid Evolutionary Particle Filter for Nonlinear State Estimation'. In *Robot, Vision and Signal Processing (RVSP), 2011 First International Conference on*, pp. 308–312. IEEE.

X. Zhang, et al. (2010). 'A smarter particle filter'. *Computer Vision–ACCV 2009* pp. 236–246.

Y. Zheng & Y. Meng (2008). 'Swarming particles with multi-feature model for free-selected object tracking'. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 2553–2558. IEEE.

Appendix A

Equivalence of AR(p) and State Space Models

Given any scalar AR(p) model in vector form:

$$\mathbf{X}_{t+1} = F\mathbf{X}_t + V_t$$
$$Y_t = H\mathbf{X}_t$$

Assuming F has dimension p, back-substitute:

$$\begin{aligned} y_{t+p} &= H(F^p\mathbf{X}_t + V_{t+p-1} + FV_{t+p-2} + \dots F^{p-1}V_t) \\ y_{t+p-1} &= H(F^{p-1}\mathbf{X}_t + V_{t+p-2} + FV_{t+p-3} + \dots F^{p-2}V_t) \\ &\dots \\ y_{t+1} &= H(F\mathbf{X}_{t-1} + V_t) \\ y_t &= H\mathbf{X}_t \end{aligned}$$

Thus the observed series y_t satisfies a difference equation of AR form with error terms V_t.

Corollary. AR models in state-space form: in the scalar AR(p) model as a vector AR(1) process with:

$$\mathbf{X}_t = F\mathbf{X}_{t-1} + V_t$$
$$y_t = H\mathbf{X}_t$$

the observation equation lifts the first element so we let $H = (1, 0, \dots, 0)$.

Appendix B

Forecasts using ARIMA(*p,d,q*) models

For a time series X_t an ARMA(p,q) model is given by:

$$\left(1 - \sum_{i=1}^{p} \alpha_i L^i\right) X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t$$

where L is the lag operator, the α_i are the autoregressive parameters, the θ_i are the moving average parameters, and the ε_t are i.i.d. sample error terms from the normal distribution.

Assume now that $\left(1 - \sum_{i=1}^{p} \alpha_i L^i\right)$ has a unitary root multiplicity of d:

$$\left(1 - \sum_{i=1}^{p} \alpha_i L^i\right) = \left(1 - \sum_{i=1}^{p-d} \phi_i L^i\right) (1 - L)^d$$

An ARIMA(*p,d,q*) process expresses the polynomial factorization process and is given by:

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) (1 - L)^d X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t \tag{B.1}$$

and thus can be thought of as an ARMA(*p+d,q*) process having the autoregressive polynomial with some roots of unity.

B.1 ARIMA(1, 1, 0)

For an ARIMA$(1, 1, 0)$ of the form (B.1), given the lag operator

$$LX_t = X_{t-1} \tag{B.2}$$

and the factorization

$$\begin{aligned} Y_t &= (1 - L)^d X_t \\ &= X_t - X_{t-1} \end{aligned} \tag{B.3}$$

the prediction states equation for an ARIMA$(1, 1, 0)$ is given by:

$$\begin{aligned}\varepsilon_t &= (1 - \alpha L)Y_t \\ &= (1 - \alpha L)(X_t - X_{t-1}) \\ &= X_t - X_{t-1} - \alpha X_{t-1} + \alpha X_{t-2}\end{aligned} \tag{B.4}$$

rearranged:

$$X_t = \alpha X_{t-1} + \varepsilon_t + (X_{t-1} - \alpha X_{t-2}) \tag{B.5}$$

B.2 ARIMA(2, 1, 0)

For an ARIMA$(2, 1, 0)$ of the form (B.1), given the definition of an AR(p):

$$\begin{aligned}\varepsilon_t &= X_t - \sum_{i=1}^{p} \alpha_i X_{t-1} \\ &= \left(1 - \sum_{i=1}^{p} \alpha_i L^i\right) X_t,\end{aligned} \tag{B.6}$$

the lag operator

$$LX_t = X_{t-1} \tag{B.7}$$

and the factorization

$$\begin{aligned}Y_t &= (1 - L)^d X_t \\ &= X_t - X_{t-1}\end{aligned} \tag{B.8}$$

the prediction states equation for an ARIMA$(2, 1, 0)$ is given by:

$$\begin{aligned}\varepsilon_t &= \left(1 - \left[\sum_{i=1}^{p} \alpha_i L^i\right]\right) Y_t \\ &= \left(1 - \left[\alpha_1 L + \alpha_2 L^2\right]\right) Y_t \\ &= \left(1 - \left[\alpha_1 L + \alpha_2 L^2\right]\right) (X_t - X_{t-1}) \\ &= X_t - X_{t-1} - \alpha_1 X_{t-1} + \alpha_1 X_{t-2} - \alpha_2 X_{t-2} + \alpha_2 X_{t-3}\end{aligned} \tag{B.9}$$

rearranged:

$$\begin{aligned}X_t &= X_{t-1} + \alpha_1 X_{t-1} - \alpha_1 X_{t-2} + \alpha_2 X_{t-2} - \alpha_2 X_{t-3} + \varepsilon_t \\ &= \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \varepsilon_t + (X_{t-1} - \alpha_1 X_{t-2} - \alpha_2 X_{t-3})\end{aligned} \tag{B.10}$$