Stony Brook University

Simulation-Based Sequential Bayesian Filtering with Rao-Blackwellization applied to Nonlinear Dynamic State Space Models

A Dissertation Presented

by

Mahsiul Khan

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

August 2009

Stony Brook University

The Graduate School

Mahsiul Khan

We, the dissertation committee for the above candidate for the **Doctor of Philosophy** degree, hereby recommend acceptance of this dissertation.

Dr. **Petar M. Djuric**, Dissertation Advisor
Professor, Department of Electrical & Computer Engineering

Dr. **Nancy Mendell**, Co-Advisor
Professor, Department of Applied Mathematics & Statistics

Dr. **Eugene Feinberg**, Chairperson of Defense
Professor, Department of Applied Mathematics & Statistics

Dr. **Haipeng Xing**, Member of the Committee
Assistant Professor, Department of Applied Mathematics & Statistics

Dr. **Jiaqiao Hu**, Member of the Committee
Assistant Professor, Department of Applied Mathematics & Statistics

Dr. **Monica F. Bugallo**, Outside Member
Assistant Professor, Department of Electrical & Computer Engineering

This dissertation is accepted by the Graduate School.

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Simulation-Based Sequential Bayesian Filtering with Rao-Blackwellization applied to Nonlinear Dynamic State Space Models

by

Mahsiul Khan

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

2009

Stochastic models are used to describe many real world random processes which necessitate the extraction of hidden (unobserved) states (signals) from noisy observable (measured) outputs. We consider a class of nonlinear dynamic state space models which contain conditionally linear and unknown static parameters. For tracking the a posteriori distribution of the hidden states of this type of models, one can apply particle filtering, which is an increasingly popular method in many fields of science and engineering. It is based on the Bayesian methodology and approximations of the distributions of interest with random measures composed of samples (particles) from the space of the states and weights associated to the particles. Particle filtering performs tracking of the desired distributions as new observations are made by modifying the random measure, that is, the particles and the weights. We address the appli-

cation of particle filtering with the use of the Rao-Blackwellization principle. Rao-Blackwellization reduces the variance of estimators, and it is based on the Rao-Blackwell theorem. In the context of particle filtering, Rao-Blackwellization allows for integration of the conditionally linear unknowns thereby decreasing the dimension of the sampling space of the particles. One novelty in this dissertation is the implementation of Rao-Blackwellization by employing the implied integration method. Another novelty is the use of the approach on the standard stochastic volatility model and regime-switching stochastic volatility model. The latter model generalizes the former by allowing changes of the model parameters at unknown instants of time. All the models are nonlinear and contain conditionally linear parameters. Simulated datasets are used to compare the performances of our algorithms with popular ones based on the Liu and West method. Both the classical and auxiliary particle filtering algorithms are applied. We demonstrate that our particle filtering algorithms outperform the ones based on the Liu and West method.

To my late Aunt, who raised me with love

and

To my daughter Jememah, who is the joy in my life.

Contents

List of Figures

Acknowledgements

It has been a great privilege to have Dr. Petar Djuric as my mentor through this long and difficult journey. As if finding a mentor who brings the best in student's academic life was not enough, I was lucky to have one who developed personal relationship, and provided support and guidance in many real life situations. He has provided me direction and inspiration, while giving me the freedom to choose and develop my own ideas in research. He gave me the opportunity to study my cherished subjects *Bayesian inference* and *stochastic modelling*. Without his support and guidance, I could not have accomplished this body of work under the very difficult circumstances, and would like to thank him for his time and endurance.

I am deeply honored to have Dr. Nancy Mendell as my mentor and also as my spiritual counselor. It would not be possible for me to overcome the obstacles I faced without her active counsel, support and caring. She inspired and strengthen me to move on during my most difficult time, and merely thanking her is not enough for her contributions for completing this work. I would like to thank Dr. Eugene Feinberg for providing me encouragement and assistance to complete this work. I must also thank to Dr. Haipeng Xing, Dr. Jiaqiao Hu and Dr. Monica Bugallo for their valuable time. My special thanks to Dr. Xiaolin Li for his cooperation and inspiration. I am grateful to Dr. Robert Frey for contributing his valuable time, who first introduced me concepts in *financial modeling*.

Finally, my family have provided me love and mental strength to overcome difficult circumstances. My wife Sonia has been an active partner of my life during the entire period, ups and downs of my graduate student life, and it would not be possible to complete this

work without her contributions. My newly born daughter Jememah is the only joy in my life when I am down. I would like to thank Fazle Sobhan and Sabiha Sobhan for their support during my difficult time. My dear friends Sajjad, Girish, Joondong and Donghyung was the source of energy during the time at Stony Brook, and I will miss their company as well as productive and stimulating discussions.

Chapter 1

Introduction

1.1 Preliminaries

Understanding of many natural phenomena could only be possible with the use of stochastic processes. The mathematical appeal of the theory of stochastic processes and its range of applications in science and engineering has made it one of the cornerstones for solving many difficult problems. In deterministic processes, there is no uncertainty in their future behavior, and they are well defined by mathematical expressions such as differential equations. On the other hand, the future of observed realizations of stochastic processes is uncertain, which is why it is modeled by probabilistic functions. Most real-world processes are stochastic in nature, and frequent objective in their study has been the analysis of their statistical properties so that one can perform estimation or prediction of the processes. The past two decades have seen an outstanding development of methods for studying stochastic processes. Many of these methods are based on computer simulations, which is due to the unprecedented growth of computing power in that period.

Real world processes such as natural systems, biological systems, communication systems, and economic systems generate observable outputs or data which can be characterized as signals. The signals can be categorized as functions of time (such as communication signals) or as functions of space (such as pixels of an image). The signals can be discrete in nature (such as binary coded bits) or continuous in nature (such as speech signals), and the

signals can be stationary or nonstationary. Most often, the signals of interest are not observed directly, and they are treated as stochastic in nature. Therefore, for their extraction, estimation, and prediction, one employs stochastic models. One big class of such models is known as dynamic state-space (DSS) models, which is also known as a class of Hidden Markov Models (HMMs) (Rabiner, 1989; Rabiner and Juang, 1993).

DSS models are composed of two sets of equations, a state equation and an observation equation. The state refers to the unobserved component that dynamically evolves over time, whereas the observation is a random quantity that depends on the current state. In a filtering problem, the objective is to have sequential inference of the hidden states based on the available observations. Initially, DSS models were introduced as linear-Gaussian state-space models (Kalman, 1960; Anderson and Moore, 1979; Harvey, 1981; West, 1997). For the linear-Gaussian models with analytic (closed form) solutions, the *Kalman filter* (Kalman, 1960) provides the optimal solution. The solution is in the form of a recursive calculation of the means and covariances of the posterior distributions of the unknowns. Most complex systems are nonlinear and non-Gaussian in nature, and they do not have analytic solutions, and therefore the Kalman filter is not suitable for them. For these problems, traditional solutions have been based on extensions of the Kalman filter, such as the *extended Kalman filter* (Anderson and Moore, 1979) and the *Gaussian sum filter* (Sorenson and Alspach, 1971; Alspach and Sorenson, 1972). Often they have yielded poor results especially in presence of high nonlinearities. An alternative is to apply grid-based methods with deterministic numerical integration schemes, but they are difficult to implement when the dimension of the unknowns is high and computationally too expensive. Yet another group of methods relies on Monte Carlo sampling, and they can be traced back to the early 1950s (Metropolis and Ulam, 1949) (later generalized in Hastings (1970)). Due to the advances of computing power, and with the work presented in (Gelfand and Smith, 1990), and the rediscovery of the contributions of Metropolis and Hastings (Metropolis-Hastings algorithm), *Markov Chain Monte Carlo* (MCMC) methods have been widely applied in the mainstream statistics, and

in almost every applied area of science. The MCMC "revolution" in the last 20 years has allowed for penetration of Bayesian statistics to new areas because the new methods virtually provide a universal tool for dealing with high dimensional integration and optimization (Cappe and Robert, 2000). The MCMC, in general, is considered quite efficient for batch data (non-sequential) processing, and the Gibbs sampler (Geman and Geman, 1984), one of the popular schemes of MCMC sampling, is viewed as very efficient for estimating posterior distributions.

Many real world problems can be formulated by using DSS models including problems in signal processing, radar, communications, computer vision, and economics. In these problems it is assumed that observations arrive sequentially in time and one requires real-time estimation and prediction of the signals. For this kind of dynamic processes, which may be highly nonlinear and non-Gaussian, the states are unobserved. The MCMC methods are not well suited for sequential estimation and/or prediction of states due to their possibly slow convergence.

In the past 15 years, for studying nonlinear and/or non-Gaussian systems, a new class of filters has gained high popularity. It is due to their efficiency and wide range of applications. This class of filters is known as *Particle Filtering* (PF) or *Sequential Monte Carlo* (SMC) methods. These algorithms allow for sequential estimation of the posterior distributions of the unknowns at each time instant. Under the Bayes' rule, the algorithms sequentially approximate the posterior distributions by discrete random measures generated by Monte Carlo (MC) sampling. The PF is well suited for sequential real-time estimation/prediction of dynamic states, where the parameters are often assumed known or evolve dynamically as states. However, when the parameters are unknown and are static (that is, they are constants), PF becomes a challenging problem. Jane Liu and Mike West have introduced a PF algorithm which sequentially estimates dynamic states and constant parameters simultaneously (Liu and West, 2001). The algorithm has been widely used and is based on an adaptive importance sampling technique combined with a weighted kernel density method

for approximating the posterior distributions (West, 1992, 1993a,b).

As pointed out previously, for the PF, it is quite challenging to estimate the posterior/predictive distributions of the unknowns when the unknowns contain static (constant) parameters. Hence, there has been necessity for the development and implementation of efficient sequential filtering algorithms for this type of models. In this dissertation, we address the DSS models with unknown constant parameters by exploiting the concept of Rao-Blackwellization (RB) (Casella and Robert, 1996; Liu and Chen, 1998; Doucet et al., 2000; Storvik, 2002; Maskell, 2004). RB is a neat scheme that is based on analytically integrating out some of the unknowns conditioned on the remaining unknowns. It is based on Rao-Blackwell theorem, which maintains that there is a reduction of the variance of an estimator on application of RB. Its implementation in the context of PF contributes to the reduction of the dimension of the sampling space and therefore to an improved accuracy of the estimation. The objective of the proposed work is to establish an efficient method for implementing sequential filtering algorithms. In our work, we implement the RB by invoking the *implied integration method*. As application examples, we have investigated a class of stochastic volatility (SV) models with static parameters, where the reduction of the sampling spaces contributes to significant reduction of the particle size (sample size), and consequently the reduction in the variance of the estimates, and hence, improved overall performance. Recently, these models have been studied by PF, but to the best of our knowledge, the RB method has never been applied to them.

1.2 State Space Models with Constant Parameters

In most problems with DSS models, the interest is in extracting the dynamic hidden states of the model, and if it has any constant parameters, they are assumed *known*. However, there are many real-world problems where the parameters are unknown. In addition, the observations arrive sequentially in time, and one needs real-time sequential estimation and/or prediction of the hidden states. As already pointed out, the widely used and popular

MCMC algorithms have been successfully applied for estimation of the hidden states and unknown static parameters for off-line setup (Liu, 2001), and in general, they are not suitable for sequential real-time estimation. For these types of problems, there are not many sequential filtering algorithms. A PF algorithm that has been popular among researchers are based on the ideas from (Gordon et al., 1993) (West, 1992, 1993a,b), and is presented in (Liu and West, 2001). That algorithm provides simultaneous and sequential estimates of the hidden states and the static parameters.

The generic form of the DSS models can be presented by two probability distribution functions (PDFs) as follows:

$$\boldsymbol{x}_t \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) \qquad (state \quad distribution) \tag{1.1}$$

$$\boldsymbol{y}_t \sim p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}) \qquad (observation \quad distribution) \tag{1.2}$$

where t is a discrete time index, $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$ is the hidden state at time instant t, $\boldsymbol{y}_t \in \mathbb{R}^{n_y}$ is the observation at time instant t, and $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is an unknown parameter vector, which is static and often considered as nuisance. In addition, $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})$ is the state transition distribution, and $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$ is the observation distribution, both often being non-Gaussian. An alternative way of describing the DSS model is by using the following sets of equations:

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t, \boldsymbol{\theta}) \qquad (state \quad equation) \tag{1.3}$$

$$\boldsymbol{y}_t = g(\boldsymbol{x}_t, \boldsymbol{v}_t, \boldsymbol{\theta}) \qquad (observation \quad equation) \tag{1.4}$$

where t, $\boldsymbol{x}_t$, $\boldsymbol{y}_t$, and $\boldsymbol{\theta}$ have the same meaning, $\boldsymbol{u}_t \in \mathbb{R}^{n_u}$ and $\boldsymbol{v}_t \in \mathbb{R}^{n_v}$ are the state noise and observation noise processes, respectively, and $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_y}$ are the state transition and observation functions, respectively. In this dissertation, we assume that at least one of the above functions are nonlinear. Our main objective is to obtain sequential inference of the hidden states $\boldsymbol{x}_t$ from the marginal filtering distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t})$,[1] in the presence of unknown static parameters $\boldsymbol{\theta}$.

[1]The notation $\boldsymbol{x}_{0:t-1}$ signifies $\boldsymbol{x}_{0:t} \equiv \{\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_t\}$.

The DSS models as described above have become very powerful for modeling dynamic systems. They have been used in a wide ranging of applications including communication and signal processing, econometrics and financial modeling, target tracking and missile guidance, terrain navigation, computer vision, robotics, industrial process control, neural networks, machine learning and biology.

1.3 Particle Filtering

We reiterate that if the state-space models are linear and Gaussian with analytic solution, the well known Kalman filter (Kalman, 1960) is the optimal solution for tracking the unknown states. However, most real world dynamical systems are nonlinear and possibly non-Gaussian. Approximating such systems with linear-Gaussian models, and the use of Kalman filter often provide poor estimates. The various extensions of the Kalman filter, such as the extended Kalman filter (Anderson and Moore, 1979), and the Gaussian sum filter (Sorenson and Alspach, 1971; Alspach and Sorenson, 1972) have often been used for studying nonlinear and non-Gaussian systems. The extended Kalman filter linearizes the nonlinear functions by using Taylor series expansion, and then follows up with the use of the standard Kalman filtering method, and the Gaussian sum filter approximates the non-Gaussian distribution by the sum of Gaussian distributions. These two methods often provide poor results if the systems are highly non-Gaussian and nonlinear. The Grid-based methods (Kitagawa, 1987) with deterministic numerical integration can give good estimate, but they are computationally very expensive and difficult to implement for higher dimension.

In the last two decades, there has been an explosion of scientific papers on MC-based simulation methods that employ the Bayesian methodology. The Bayesian methods have proved to be quite effective for recursive estimation of hidden states of DSS models.[2] In most problems of interest, prior knowledge about the unknowns of the model is available.

[2]The Kalman filter can also be interpreted as a Bayesian method.

The data modify the prior knowledge through the likelihood function, and the result of the modification is the posterior PDF of the unknowns.

Thus, all information of the unknowns is captured by the posterior PDF, and therefore, our main objective is to get an estimate of it (Box and Tiao, 1992; Doucet et al., 2001). The transition from the prior to the posterior PDF is given by

$$\underbrace{p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}, M)}_{posterior} = \frac{\overbrace{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t}, M)}^{likelihood}\overbrace{p(\boldsymbol{x}_{0:t}|M)}^{prior}}{\underbrace{p(\boldsymbol{y}_{1:t}|M)}_{evidence}} \tag{1.5}$$

Where the notation $\boldsymbol{y}_{1:t} \equiv \{y_1 \cdots y_t\}$ is the observation series, $\boldsymbol{x}_{0:t}$ is the unknown hidden state sequence and M denotes the assumed model. Often, observations arrive sequentially in time, and it is our interest to obtain on-line (real-time) sequential inference about the unknowns. Therefore, it is necessary to update the posterior PDF as the new observations become available. In mathematical terms, our objective is to get an estimate of the posterior PDF $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ recursively in time, and obtain one of the important marginal PDF, the *filtering distribution* $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ from this posterior PDF. PF or SMC (Gordon et al., 1993; Doucet et al., 2001; Arulampalam et al., 2002; Maskell, 2004; Djuric and Bugallo, 2009) methods are a set of MC-based simulation algorithms, which sequentially estimate the posterior PDF. For nonlinear and/or non-Gaussian DSS models, PF has shown to outperform other existing methods. The central idea of the PF is to represent the posterior PDF with a *random measure*, composed of a set of random samples, also called particles, and their associated weights (Carlin et al., 1992; West, 1992; Muller, 1991). In generating the random samples and computing their weights, one exploits the concept of *importance sampling* (Rubin, 1988; Geweke, 1989).

The full posterior distribution is expressed with the random measure as follows:

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta(\boldsymbol{x}_{0:t} - \boldsymbol{x}_{0:t}^i) \tag{1.6}$$

where $\boldsymbol{x}_{0:t}^i$, and w_t^i, $i = 1, 2 \cdots, N$ are the particle streams and their weights, respectively, and $\delta(\cdot)$ is the Dirac delta function. We note that the particles $\boldsymbol{x}_t^i$ are generated by an

importance function. As the number of particles tends to infinity, the random measure, under some given conditions, tends to almost surely to the posterior PDF (Crisan and Doucet, 2002). With a recursive importance function, one can show that the weights are updated according to

$$w_t^i \propto w_{t-1}^i \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^i)p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{0:t-1}^i, \boldsymbol{y}_{1:t})} \tag{1.7}$$

where $p(\cdot)$ is a generic posterior distribution and $q(\cdot)$ is a generic importance function. The function $q(\cdot)$ is chosen such that the support of $q(\cdot)$ contain the support of $p(\cdot)$ i.e., $supp(q) \supset supp(p)$.

1.4 Rao-Blackwellization on State Space Models

RB (Casella and Robert, 1996; Liu and Chen, 1998; Doucet et al., 2000; Storvik, 2002; Maskell, 2004) allows for improved estimation of unknowns based on the Rao-Blackwell theorem (Lehmann, 1991). The application of RB reduces the variance or the uncertainty of the estimate, that is, it improves the accuracy of the estimates. For stochastic modeling, most of the MC-based simulation methods approximate a posterior PDF of a state vector, which may be of high dimension. Updating the posterior PDF, the necessary integrations which cannot be carried out analytically are performed by the MC method. Clearly, if we had analytic integrations, there would be no errors in the integration, whereas any stochastic integration by its nature introduces errors. We consider a class of DSS models with dynamic states and with unknown static parameters, which are conditionally linear given the nonlinear states. In particular, we work with a model where the state equation is linear in the static parameters given the state of the model.

Our objective is to develop a filtering algorithm for Rao-Blackwellized state space models for sequential estimation. Since the parameters are nuisance to the models, the application of RB integrates out the parameters, which consequently *reduces* the dimension of the sampling space. With the reduction of the unknown sampling space, the approximation of the posterior PDF improves considerably for a fixed number of particles (Evans and Swartz,

2000). The latter implies that we would need a smaller set of particles to approximate the posterior PDF for good accuracy, implying the reduction in computations and their cost. Hence, RB leads to many favorable outcomes, and should be applied whenever possible.

In implementing the RB on the parameter space of the DSS models as defined in the previous section (1.1-1.2), we need to integrate out the conditionally linear parameters $\boldsymbol{\theta}$.[3] Formally, the marginalization in the state equation is carried out according to

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}) &= \int_{\boldsymbol{\theta}} p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{x}_{0:t-1})\mathrm{d}\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\theta}} p(\boldsymbol{x}_t|\boldsymbol{\theta}, \boldsymbol{x}_{0:t-1})p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t-1})\mathrm{d}\boldsymbol{\theta} \end{aligned} \tag{1.8}$$

and likewise, the marginalization of the likelihood by

$$\begin{aligned} p(\boldsymbol{y}_t|\boldsymbol{x}_{0:t}) &= \int_{\boldsymbol{\theta}} p(\boldsymbol{y}_t, \boldsymbol{\theta}|\boldsymbol{x}_{0:t})\mathrm{d}\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\theta}} p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{x}_{0:t})p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t})\mathrm{d}\boldsymbol{\theta}. \end{aligned} \tag{1.9}$$

The implementation of the RB method requires that the integrals obtained from the state and observation equations are analytically solvable. In our approach, we propose the implementation RB using the *implied integration method*, which avoids direct integration. This implied integration technique is also known as the candidate's formula (Besag, 1889). Thus, the desired marginalized filtering PDF of $\boldsymbol{x}_t$ has the form

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t}) &= \int_{\Theta} p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t})\mathrm{d}\boldsymbol{\theta} \\ &\propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}) \end{aligned} \tag{1.10}$$

The marginalization allows us to generate particles only from the space of $\boldsymbol{x}_t$, without generating particles from the space of the parameter vector $\boldsymbol{\theta}$. This entails significant advantage with respect to numerical computation and accuracy of the estimates.

[3] In the sequel, $\boldsymbol{\theta}$ denotes a vector of conditionally linear parameters.

1.5 Application to Stochastic Volatility Models

Even though there are various interpretations of *volatility*, it mainly represents a measure of asset price variability over some period of time. It is the standard deviation of the asset returns, and is denoted as σ. Another interpretation of volatility is the changes in the logarithm of variance of the asset price over some time period. Empirical studies suggest that volatility or variance of the financial return series is not constant, i.e., it changes over time. The source of the volatility change is not well known. The changing phenomena of variance over time is called *heteroscedasticity*, and this assumption leads to model the volatility as a stochastic process. The standard volatility model of excess return series y_t is defined by (Taylor, 2005),

$$y_t = \sigma_t v_t \tag{1.11}$$

where $y_t = r_t - \mu$ with r_t and μ being the standard return series and expected return, respectively, σ_t is the volatility, and v_t is a zero mean white Gaussian noise process with unit variance.[4] The processes σ_t and v_t are stochastically independent. The above implies that

$$y_t \sim \mathcal{N}(0, \sigma_t^2). \tag{1.12}$$

Forecasting and understanding the *volatility* is the prime objective of many researchers in statistics, finance and economics.

There are various volatility models. One of them is the *ARCH (AutoRegressive Conditional Heteroscedasticity)* model, which was first introduced by Robert Engle (Engle, 1982), and later developed by Bollerslev (Bollerslev, 1986) as *GARCH (generalized ARCH)* model. The simple ARCH model of the excess return series is an autoregressive process of order one, AR(1), i.e.,

$$p(y_t|y_{t-1}, y_{t-2}, \ldots) \sim \mathcal{N}(0, \sigma_t^2) \tag{1.13}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 \tag{1.14}$$

[4]The standard return series r_t is defined as $r_t = \mu + \sigma_t v_t$.

where α_0 and α_1 are constant parameters of the model and y_t represents the excess return series. The model is used for *volatility* forecasting, and the estimation is usually performed by the maximum likelihood (ML) method.

An alternative hypothesis is that the volatility is a latent or hidden process, and cannot be observed directly from the return series. Hence, the formation of the stochastic volatility (SV) models (Clark, 1973; Taylor, 1982; Jacquier et al., 1994; Kim et al., 1998; Shephard, 2005) as a class of discrete time dynamic state-space models was considered. The continuous time models have also been studied for pricing of financial derivatives such as *options*. The SV models specify a stochastic process for modeling the volatility, where the log-volatility is considered as a hidden process. The SV models differ from the ARCH models that specify the volatility process modeled with return series. SV and ARCH models have similarities with respect to describing the volatility measure. However, some researchers (Shephard, 1996) have concluded that estimation of SV models is easier than ARCH models when the series become very large. Moreover, the properties of SV models are easier to find and understand. They are also easier to generalize to multivariate models. There is a surge of interests by researchers in finance and in statistics for the development of various algorithms as a solutions to the SV models. Bayesian inference of SV models using various Markov Chain Monte Carlo (MCMC) schemes (Jacquier et al., 1994; Kim et al., 1998) proved to be quiet efficient. The MCMC sampling schemes are used to estimate the unobserved log-volatility and unknown parameters.

The standard and most used form of SV model is as folllows:

$$x_t = \alpha + \beta x_{t-1} + \sigma_u u_t \quad (state \quad equation) \tag{1.15}$$

$$y_t = e^{x_t/2} v_t \quad (observation \quad equation) \tag{1.16}$$

where, $x_t \in \mathbb{R}$ is a hidden state, and represents the logarithm of variance, and $u_t \in \mathbb{R}$ and $v_t \in \mathbb{R}$ are uncorrelated Gaussian noises with zero means and unit variances. The static parameters $\boldsymbol{\theta} = (\alpha \ \beta \ \sigma_u^2)^\top$ are unknown, but nuisance. Our objective is to implement a filtering algorithm to sequentially estimate the hidden state x_t based on the available

observations $y_{1:t}$, and the model (1.15) – (1.16). The state equation is linear in the unknown parameters α and β, and it is Gaussian, whereas the observation equation is nonlinear in the state x_t. We want to apply RB to the state equation to integrate out the unknown static parameters $\boldsymbol{\theta}$, and then implement PF for sequential estimation of the hidden state (the log-volatility). As a wider application example, we want to extend the standard SV model to regime-switching stochastic volatility (RSSV) models, where switching among regimes are based on Markov process, and again apply RB to reduce the sampling dimension and thereby improve the estimation accuracy of the method.

1.6 Contributions

The main contributions of this dissertation are the application of RB in the context of PF by using the implied integration technique, and the use of the procedure on the standard SV and RSSV models. The simulation results of the proposed procedures on these models show significant advantages over the standard PF methods that have been applied to these models before.

1.7 Organization of the Thesis

In Chapter 2, we introduce the DSS models and the filtering, predictive and smoothing PDFs of the state. We also review algorithms for PF including the standard PF (SPF) and the auxiliary particle filtering (APF).

In Chapter 3, we describe DSS models with unknown static parameters and filtering algorithms for sequential estimation of the states and the parameters. A PF algorithm developed by Liu & West is presented in detail. It is shown how the algorithm is applied to stochastic volatility (SV) models with simulated data, and some results are presented.

In Chapter 4, we describe the concept of RB and its application to DSS models with unknown and static parameters. We develop a PF algorithm for the standard SV model. We, then test its performance on simulated data. We compare the performance with that

of the algorithm of Liu & West. As a metrics for comparison we use the root mean squared error (RMSE), and the effect of sampling size on convergence. The performances of the SPF and APF are also provided.

In Chapter 5, we introduce the general regime-switching state space model, where switching occurs stochastically in time in accordance with a first-order stationary Markov process. As an application, we extend the standard SV model to a RSSV model with two discrete regimes. As before, we apply RB to reduce the dimensionality of the unknowns. We present our PF algorithm in detail and show its performance on simulated data. We also compare its performance to the performance of the APF algorithm by Carvalho & Lopes.

Finally, in Chapter 6 we provide the conclusions of our work and a list of possible directions for extending this effort to related challenging problems.

Chapter 2

DSS Models and Filtering

2.1 DSS Models: An Overview

DSS models were originally developed by control engineers, and they are applied to a wide range of dynamical systems where we have unobserved states (or signals) and observations that are functions of the states. Many problems in control theory, signal processing, speech processing, seismology, radar, time series, economics, and biostatistics can be expressed in the form of a DSS models. In general, they are described by a *state equation* and an *observation equation*. The state randomly evolves over time, and the observation is a quantity that is a function of the state and is perturbed by a random noise process. The objective is to estimate the hidden states based on the observations in a sequential manner.

A DSS model describes a dynamical systems of which measurements are available by a set of equations that is aimed to capture the properties and behavior of the systems. The system functions in real time, where the time is considered as an independent variable in the equations. The system is causal, i.e., the output at time $t = t_0$ does not depend on the input at time $t > t_0$. The system can be linear or non-linear, and can have discrete or continuous time outputs, or can be discrete- or continuous-time system. Accordingly, the system can be described either by difference or differential equations (Anderson and Moore, 1979). In this dissertation, we work with discrete-time DSS models.

The observations are *noisy* or *stochastic* in nature. The uncertainty in the system

can arise in various ways: the inputs signals are *random*, and the observations contain measurement errors or noise. The DSS models are described by two equations: a *transition or state equation* and an *observation or measurement equation.*

State Equation: This equation describes the dynamics of the state variables over time. The evolution equation can be linear or non-linear and usually includes a noise process that is usually modeled as a Gaussian process.

Observation Equation: This equation provides the relationship between the observations and the state variables, and it includes another random noise process. The functional relationship can be linear or non-linear, and the noise process can be Gaussian or non-Gaussian.

The discrete-time DSS model is written in mathematical terms as follows:

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) \qquad (state \quad equation) \tag{2.1}$$

$$\boldsymbol{y}_t = g(\boldsymbol{x}_t, \boldsymbol{v}_t) \qquad (observation \quad equation) \tag{2.2}$$

where $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$ is the hidden state, $\boldsymbol{y}_t \in \mathbb{R}^{n_y}$ is the observation at time t, and $\boldsymbol{u}_t \in \mathbb{R}^{n_u}$ and $\boldsymbol{v}_t \in \mathbb{R}^{n_v}$ are the state noise and observation noise processes respectively. The functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_y}$ are known as state transition and observation functions respectively.

The $\boldsymbol{u}_t$ is a zero mean white noise process, and is independent of $\boldsymbol{v}_t$ and the states, $\boldsymbol{x}_t$. The $\boldsymbol{v}_t$ is also another zero mean white noise process, and is independent of the states, $\boldsymbol{x}_t$. The PDFs of $\boldsymbol{u}_t$, $\boldsymbol{v}_t$, and the initial distribution of $\boldsymbol{x}_0$ are assumed known. The state process has the Markov property, and the observation process has the conditional independent property. If the functions $f(\cdot)$ and $g(\cdot)$ are linear, and the noise processes $\boldsymbol{u}_t$ and $\boldsymbol{v}_t$ are Gaussian, the DSS model is known as linear-Gaussian state space models. For this type of models, when the PDF of $\boldsymbol{x}_0$ is Gaussian, the posterior PDF remains Gaussian at all times, and its exact analytical expression can be obtained. The solution to this models is the well known *Kalman filter* (Kalman, 1960), which recursively calculates the means and covariances of the states.

If the functions $f(\cdot)$ and $g(\cdot)$ are non-linear, and/or the noise processes are non-Gaussian, in most cases there is no analytic or closed form (optimal) solution. For over 30 years, various extensions of the Kalman filter have been developed (see Chapter 1). As already pointed out, they are inadequate in many important situations. Recently, for the complex nonlinear and non-Gaussian problems, a new class of filters has reemerged, and has become very popular, and it is known as *particle filters*. We reiterate that PF approximates posterior PDFs of interest by discrete random measures, and to that end, it employs the Bayesian theory and MC simulations method(Gordon et al., 1993; Arulampalam et al., 2002; Doucet et al., 2001; Maskell, 2004; Djuric and Bugallo, 2009).

2.1.1 Filtering

Filtering is a generic term, and it is being used in everyday lives for many centuries. Since ancient times, men has tried to remove visible impurities from water by a method called *filtering*. The first meaning of a noun filter by a dictionary is *" a contrivance for freeing liquids from suspended impurities, especially by passing them through strata of sand, charcoal, etc"* . However, modern usage of filtering has more elaborate and abstract meaning than just removing impurities from liquids. Filtering is a mechanism by which some unwanted element(s) or quantity is removed by passing the object(s) or information through a system in order to obtain the desired element(s) or quantity (Anderson and Moore, 1979). Examples include news filtering out of war zones, filtering adult material from the Internet, and filtering ultraviolet ray in the lens. In communications, the received signals are usually corrupted with noise, and the objective is to recover the transmitted signals based on the received ones. Suppose, $\boldsymbol{y}_t$ is the received signal at time t, which is the noisy version of the transmitted signal $\boldsymbol{x}_t$. Then, a model of the received signal could have the following form:

$$\underbrace{\boldsymbol{y}_t}_{observed\ signal} = \underbrace{\boldsymbol{x}_t}_{true\ signal} + \underbrace{\boldsymbol{\epsilon}_t}_{noise\ process} . \tag{2.3}$$

The filtering process is the recovery of $\boldsymbol{x}_t$ from $\boldsymbol{y}_t$ or the extraction of information about $\boldsymbol{x}_t$ from $\boldsymbol{y}_t$. From a probabilistic point of view, our interest is to obtain the inferences of

posterior PDF of $\boldsymbol{x}_t$ given all the available information up to time instant t, i.e., $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$. This PDF contains all the information about $\boldsymbol{x}_t$ given $\boldsymbol{y}_{1:t}$, and it is known as *filtering* PDF.

2.1.2 Prediction

The objective of prediction is to estimate the value of a state at future time, for example, $(t+k)$ for $k > 0$, given the information available up to time t, i.e., we want to predict $\boldsymbol{x}_{t+k}$, based on the available information in $\boldsymbol{y}_{1:t}$. In general, we want to obtain the inference of the predictive PDF $p(\boldsymbol{x}_{t+k}|\boldsymbol{y}_{1:t})$.

2.1.3 Smoothing

Smoothing is usually used for retrospective analysis or delayed inference about the states $\boldsymbol{x}_t$. Suppose, the observations $\boldsymbol{y}_{1:T}$ are available, and we want to find the estimate of $\boldsymbol{x}_t$, $t < T$, i.e., we are interested in obtaining the inference of smoothing PDF $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T})$.

2.2 The PF Methodology

PF methods are a set of MC-based simulation methods that are developed by using the Bayesian methodology. They provide sequential estimation of posterior PDFs of nonlinear and non-Gaussian states of DSS models. The literature suggests that the performance of PF can be superior over the existing methods. Various filters based on PF appear under different names in the literature including bootstrap filter, condensation algorithm, Monte Carlo filter, interacting particle filter, and survival of the fittest filter. PF is a very flexible methodology, easy to implement, and has wide range of applications in science and engineering, including target tracking and missile guidance, terrain navigation, computer vision, neural networks, population biology, financial modeling, and time series analysis and forecasting. Before we explain the procedure of the generic PF, we briefly review the concepts of *Bayesian inference* and *importance sampling (IS)*.

2.2.1 Bayesian Inference

Under Bayesian formulation, the inference about the unknown states is contained in the posterior PDF $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$, where $\boldsymbol{x}_{0:t} \equiv \{\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_t\}$ and $\boldsymbol{y}_{1:t} \equiv \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_t\}$ are the states and observations up to time t, respectively. The full posterior PDF is defined as

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})}{p(\boldsymbol{y}_{1:t})} \tag{2.4}$$

where the normalizing constant is given by

$$p(\boldsymbol{y}_{1:t}) = \int p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})\mathrm{d}\boldsymbol{x}_{0:t} \tag{2.5}$$

This joint posterior PDF of $\boldsymbol{x}_{0:t}$ can be expressed recursively as

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})\frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} \tag{2.6}$$

The posterior PDF $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$, and its various features are the main object in *Bayesian inference* (Box and Tiao, 1992; Doucet et al., 2001). In many applications, we are interested in estimating the posterior PDF sequentially in time, and often our interest on its marginal, the so called *filtering* PDF, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$. The filtering PDF is defined as

$$\begin{aligned} p(\boldsymbol{x}_t|y_{1:t}) &= \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} \\ &= \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{\int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})\mathrm{d}\boldsymbol{x}_t}. \end{aligned} \tag{2.7}$$

PF methods obtain the distributions of interest by implementing *Bayesian filtering* method recursively in time with MC simulations. The key idea is to represent the posterior distributions by a *random measure* or a set of random samples, also known as *particles* with their associated *weights*. These weighted particles approximate the posterior PDF of interest $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$, by the random measure, $\chi_{0:t} = \{\boldsymbol{x}^i_{0:t}, w^i_t\}^N_{i=1}$, where $\{\boldsymbol{x}^i_{0:t}\}^N_{i=1}$ are the set of support points with associated weights $\{w^i_t\}^N_{i=1}$, and $\{\boldsymbol{x}^i_{0:t}\}^N_{i=1}$ are the possible trajectories or realizations of the state up to time instant t. The weights are normalized to unity, i.e., $\sum^N_{i=1} w^i_t = 1$. Mathematically, we express

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N} w^i_t \delta(\boldsymbol{x}_{0:t} - \boldsymbol{x}^i_{0:t}) \tag{2.8}$$

where $\delta(\cdot)$ is the *Dirac delta* function. These weighted particles constitute the discrete approximation of the true posterior PDF. The weights are computed by using the principle of *importance sampling (IS)* (Geweke, 1989). If one is interested in sequential estimation of the filtering PDF of $\boldsymbol{x}_t$, only $\boldsymbol{x}_t^i$ needs to be stored, and the history of the states $\boldsymbol{x}_{0:t-1}^i$ and the observations $\boldsymbol{y}_{0:t-1}$ can be discarded. With the discrete random measure, the filtering PDF is approximated as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i). \tag{2.9}$$

As the number of particles tends to infinity, the random measure, under given some conditions, tends almost surely to the true posterior PDF (Crisan and Doucet, 2002).

2.2.2 Importance Sampling (IS)

Ideally, we would like to sample directly from the posterior distribution itself. However, direct sampling from the posterior distributions is not feasible with the models with high dimensionality, and also with nonlinearity and non-Gaussianity. In that scenario, the concept of *importance sampling (IS)* (Rubin, 1988; Geweke, 1989) comes in very handy. With the importance sampling, an arbitrary and easy to sample importance function $q(\cdot)$ is chosen such that the *support* of $q(\cdot)$ contain the support of $p(\cdot)$, i.e., $supp(q) \supset supp(p)$, where $p(\cdot)$ is the generic target posterior PDF . Research indicate that estimates based on importance sampling can be super-efficient. That is, it is possible to find a distribution $q(\cdot)$ that can be used for generating samples, which yield estimates with lower variance than the direct MC-based sampling method. High sampling efficiency can be achieved by sampling from the important regions, i.e., from the region of *high probability density* (HPD), and hence the name importance sampling.

Suppose, one is interested in the expectation of an arbitrary function $h(\boldsymbol{x}_{0:t})$,

$$I(h) = E_{p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})}[h(\boldsymbol{x}_{0:t})] = \int h(\boldsymbol{x}_{0:t}) p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) \mathrm{d}\boldsymbol{x}_{0:t}$$

Based on the *IS* principle, we get the identity (Geweke, 1989),

$$I(h) = \frac{\int h(\boldsymbol{x}_{0:t}) w(\boldsymbol{x}_{0:t}) q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) \mathrm{d}\boldsymbol{x}_{0:t}}{\int w(\boldsymbol{x}_{0:t}) q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) \mathrm{d}\boldsymbol{x}_{0:t}}$$

where we assume that the *support* of the importance function $q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \{\boldsymbol{x}_{0:t} \in \mathbb{R}^{n_x \times (t+1)} : q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) > 0\}$ contain the *support* of $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \{\boldsymbol{x}_{0:t} \in \mathbb{R}^{n_x \times (t+1)} : p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) > 0\}$ The *unnormalized* importance weight is defined as

$$\begin{aligned} w(\boldsymbol{x}_{0:t}) &\propto \frac{p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})}{q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})} \\ &= \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t}) p(\boldsymbol{x}_{0:t})}{q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})} \end{aligned} \tag{2.10}$$

Hence, if one simulates i.i.d. particles $\{\boldsymbol{x}_{0:t}^i\}_{i=1}^N$ from $q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$, an estimate of $I(h)$ is obtained as

$$\begin{aligned} \widehat{I}_N(h) &= \frac{\frac{1}{N}\sum_{i=1}^N h(\boldsymbol{x}_{0:t}^i) w(\boldsymbol{x}_{0:t}^i)}{\frac{1}{N}\sum_{i=1}^N w(\boldsymbol{x}_{0:t}^i)} \\ &= \sum_{i=1}^N h(\boldsymbol{x}_{0:t}^i) \widetilde{w}_t^i \end{aligned} \tag{2.11}$$

where the *normalized* importance weights $\widetilde{w}_t^i$ are given by,

$$\widetilde{w}_t^i = \frac{w(\boldsymbol{x}_{0:t}^i)}{\sum_{i=1}^N w(\boldsymbol{x}_{0:t}^i)} \tag{2.12}$$

Under weak assumptions, as $N \to \infty$, the *strong law of large numbers (SLLN)* applies, that is, $\widehat{I}_N(h) \xrightarrow{a.s} I(h)$.

2.2.3 Sequential Importance Sampling (SIS)

The above *IS* technique is not recursive, i.e., it does not allow for sequential estimation in time. At each time instant, when a new observation $\boldsymbol{y}_t$ arrives, one needs to recompute the *importance weights* over the entire state sequence, which increases the computational complexity as time increases. Here, we consider a *sequential importance sampling (SIS)* technique which has a substantial advantage in computational complexity for sequential estimation. The *IS* method can be applied to obtain sequential inferences of $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ at

time t, without modifying the past trajectories $\{\boldsymbol{x}^i_{0:t-1}\}_{i=1}^N$ along with the weights $\{w^i_{t-1}\}_{i=1}^N$ (which approximate the posterior $p(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$ at time $t-1$). If the *importance function* can be factorized as,

$$q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \underbrace{q(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})}_{keep\quad existing\quad path} \underbrace{q(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t})}_{extend\quad path} \tag{2.13}$$

then it is possible to obtain samples (trajectories) $\boldsymbol{x}^i_{0:t} \sim q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ up to time t, by augmenting the existing trajectories $\boldsymbol{x}^i_{0:t-1} \sim q(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$ with the current state samples $\boldsymbol{x}^i_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t})$ at time t. Since the full posterior distribution at time t can be expressed recursively as in (2.6), the factorization of the importance function in (2.13) allows us to evaluate the importance weight recursively in time. Thus, recursive importance function allows us to express the weight update equation as,

$$\begin{aligned} w^i_t &\propto \frac{p(\boldsymbol{x}^i_{0:t}|\boldsymbol{y}_{1:t})}{q(\boldsymbol{x}^i_{0:t}|\boldsymbol{y}_{1:t})} \\ &\propto w^i_{t-1}\frac{p(\boldsymbol{y}_t|\boldsymbol{x}^i_t)p(\boldsymbol{x}^i_t|\boldsymbol{x}^i_{t-1})}{q(\boldsymbol{x}^i_t|\boldsymbol{x}^i_{0:t-1},\boldsymbol{y}_{1:t})} \end{aligned} \tag{2.14}$$

Furthermore, if $q(\boldsymbol{x}^i_t|\boldsymbol{x}^i_{0:t-1},\boldsymbol{y}_{1:t}) = q(\boldsymbol{x}^i_t|\boldsymbol{x}^i_{t-1},\boldsymbol{y}_t)$, then the importance function depends only on $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, instead of the entire history, and it is suitable for estimating the filtering PDF. The modified weight update equation then becomes

$$w^i_t \propto w^i_{t-1}\frac{p(\boldsymbol{y}_t|\boldsymbol{x}^i_t)p(\boldsymbol{x}^i_t|\boldsymbol{x}^i_{t-1})}{q(\boldsymbol{x}^i_t|\boldsymbol{x}^i_{t-1},\boldsymbol{y}_t)} \tag{2.15}$$

2.2.4 Degeneracy

One of the biggest problem of PF is that after a few recursions a very few particles will have significant weights, while the weights of the rest of the particles will be negligible. This phenomenon is called *degeneracy*. The particles with negligible weights will have no contribution to the estimation/approximation of the posterior distribution, and carrying them forward is a mere waste of computational power. This degeneracy problem cannot be avoided, since it can be shown that the variance of the weights increases over time (Kong

et al., 1994; Doucet et al., 2000). A good measure of degeneracy is introduced by Liu (Liu and Chen, 1998), and it is called the *effective sample size*, N_{eff}, which is defined as

$$N_{eff} = \frac{N}{1 + var(w_t^i)} \tag{2.16}$$

where w_t^i is the "true weight" given by

$$w_t^i = \frac{p(\boldsymbol{x}_t^i|\boldsymbol{y}_{1:t})}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i, \boldsymbol{y}_t)} \tag{2.17}$$

Since (2.16) cannot be evaluated exactly, an estimate for practical use can be obtained by

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N}(\widetilde{w}_t^i)^2} \tag{2.18}$$

where $\widetilde{w}_t^i$ is the *normalized* weight computed according to (2.15). If $N_{eff} \leq N_{tresh}$, i.e., when N_{eff} falls below some threshold, it is an indication of degeneracy. The degeneracy problem cannot be eliminated, but can be reduced by: (1) a good choice of importance function, and (2) applying resampling algorithm. In the next two subsections, we discuss the choice of importance functions and resampling algorithms.

2.2.5 Choice of Importance Function

To reduce the degeneracy of PF, a natural choice consists of selecting a *importance function* which minimizes the variance of the importance weights conditioned on the past trajectory $\boldsymbol{x}_{0:t-1}^i$, and the observation up to time t, $\boldsymbol{y}_{1:t}$. It has been shown in (Doucet et al., 2000) that

$$var_{q(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}^i, \boldsymbol{y}_{1:t})}[w_t^i] = 0 \tag{2.19}$$

when the importance function is chosen as,

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

and substituting (2.20) into (2.15), we get

$$\begin{aligned} w_t^i &\propto w_{t-1}^i p(\boldsymbol{y}_t|\boldsymbol{x}_{t-1}^i) \\ &= w_{t-1}^i \int p(\boldsymbol{y}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i) \mathrm{d}\boldsymbol{x}_t \end{aligned} \tag{2.21}$$

The importance function $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i, \boldsymbol{y}_t)$ is considered to be the optimal importance function. However, it suffers from two major setbacks, (1) its use requires to be able to sample from $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i, \boldsymbol{y}_t)$, and (2) evaluate $p(\boldsymbol{y}_t|\boldsymbol{x}_{t-1}^i) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i) \mathrm{d}\boldsymbol{x}_t$. Besides some limited cases, in general, this integral does not have a closed form solution.

In practice, the importance function is chosen based on the ability to draw samples from it. The most popular importance function is the prior density and is defined as,

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}^i, \boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i) \tag{2.22}$$

and the corresponding weight update equation by substituting (2.22) into (2.15) becomes,

$$w_t^i \propto w_{t-1}^i p(\boldsymbol{y}_t|\boldsymbol{x}_t^i) \tag{2.23}$$

The prior importance density may cause degeneracy if the likelihood function is much narrower than the prior, or the prior and the likelihood are well separated.

2.2.6 Resampling

Resampling also reduces degeneracy in PF algorithm. The resampling step is applied when significant degeneracy is observed, i.e., when N_{eff} is less than some threshold N_T. The main idea of resampling is to discard the trajectories with insignificant weights and to replicate the trajectories with significant weights. The resampling step transforms the random measure with unequal weights $\{\boldsymbol{x}_t^i, w_t\}_{i=1}^N$ to equal weights, $\{\boldsymbol{x}_t^j, \frac{1}{N}\}_{j=1}^N$. There are several resampling algorithms in use, such as multinomial resampling (random resampling), residual resampling, and systematic resampling. It has been observed that there is no significant performance difference of PF when different resampling algorithms are used. There is, however, difference in computational cost. In our work, we used the systematic resampling (Kitagawa, 1996) algorithm.

The systematic resampling algorithm (Arulampalam et al., 2002) was first introduced by Kitagawa (Kitagawa, 1996). We gave preferences to this algorithm because it is relatively easy to implement, has complexity of the order of *O(N)*, and introduces *minimum Monte Carlo variation.*

The steps of systematic resampling algorithm are as follows:

$[\{\boldsymbol{x}_t^{j*}, w_t^j, i^j\}_{j=1}^N]$=systematic resample $[\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^N]$

- initialize : $c_1 = 0$
- for $i = 2 : N$
  - $c_i = c_{i-1} + w_t^i$
- end
- Draw a starting point : $u_1 \sim U[0, \frac{1}{N}]$
- for $j = 1 : N$
  - $u_j = u_1 + \frac{j-1}{N}$
  - while $u_j > c_i$
    - $i = i + 1$
  - end
  - Assign sample : $\boldsymbol{x}_t^{j*} = \boldsymbol{x}_t^i$
  - Assign weight : $w_t^j = \frac{1}{N}$
  - Assign parent : $i^j = i$
- end

Although the resampling technique reduces the degeneracy problem, it introduces other problems. Besides, adding additional computational cost, it has two major drawbacks- the first one is the lack of *parallelizability*, and the second one is lack of *diversity.* The first problem is of interest typically for hardware implementation of PF. The second is always the interest of all. The resampling algorithm statistically selects particles with higher weights (w_t^i) more times than particles with lower weights. As a result, the resampled trajectories contain many more repeated ones, which causes loss of diversity. This phenomenon is called *sample impoverishment.* The sample impoverishment is dominant for DSS models

with small state noise variance. There have been some systematic methods to counter the effects of sample impoverishment. One of them is the resample move algorithm (Gilks and Berzuini, 2001), which combines the PF with MCMC sampling. There, the MCMC sampling is used to increase the diversity after resampling. Another method that reduces the sample impoverishment is called *regularization* (Musso et al., 2001).

2.3 A Generic PF Algorithm

This algorithm is based on choosing prior distribution as importance function, i.e., $q(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1})$ (Doucet et al., 2001), and with the use of *SIS* method. We refer to it as SPF algorithm and its outline is as follows:

1. *Initialization*, $t = 0$
   - for $i = 1, \ldots, N$, sample $x_0^i \sim p(x_0)$ and set $t = 1$.
2. *Importance sampling step*
   - for $i = 1, \ldots, N$, sample $x_t^i \sim p(x_t|x_{t-1}^i)$ and set $x_{0:t}^i = (\widetilde{x}_{0:t-1}^i, x_t^i)$.
   - for $i = 1, \ldots, N$, evaluate the importance weights, $w_t^i = w_{t-1}^i p(y_t|x_t^i)$.
   - normalize the importance weights: $\widetilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}, \quad i = 1, \ldots, N$.
3. *Selection step*
   - resample with replacement N particles$\{\widetilde{x}_{0:t}^i\}_{i=1}^N$ from the set $\{x_{0:t}^i\}_{i=1}^N$ with the importance weights according to some resampling algorithm.
   - If resampling takes place, set $\widetilde{w}_t = 1/N$.
   - set $t \leftarrow t + 1$ and go to step 2.

There are various PF algorithms of which three have been most popular. They are the sampling importance resampling (SIR) filter (which is the SPF algorithm described above), the auxiliary particle filter (APF) and the regularized particle filter (RPF).

2.4 Auxiliary Particle Filtering (APF) Method

APF was introduced by Pitt and Shephard (Pitt and Shepard, 1999). This method introduces an *IS* density with a special form. It also assumes that sampling can be performed from the prior density $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. The main idea of this algorithm is that at time t, we choose to propagate particles which are from the regions of HPD of a predictive PDF of the states. The method samples the pair $\{\boldsymbol{x}_t^j, i^j\}_{j=1}^N$ from the joint importance function $q(\boldsymbol{x}_t, i|\boldsymbol{y}_{1:t})$, where the index i^j represent the index of the particles at time $t-1$. With the application of Bayes' theorem, the joint posterior can be expressed as,

$$\begin{aligned} q(\boldsymbol{x}_t, i|\boldsymbol{y}_{1:t}) &\propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t, i|\boldsymbol{y}_{1:t-1}) \\ &\propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|i, \boldsymbol{y}_{1:t-1})p(i|\boldsymbol{y}_{1:t-1}) \\ &\propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)w_{t-1}^i \\ &\propto p(\boldsymbol{y}_t|\boldsymbol{\mu}_t^i)p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)w_{t-1}^i \end{aligned} \tag{2.24}$$

where $\boldsymbol{\mu}_t^i$ represents some characteristic of $\boldsymbol{x}_t^i$ given $\boldsymbol{x}_{t-1}^i$. This characteristic can be the mode, the mean, $\boldsymbol{\mu}_t^i = E[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i]$ or a sample, $\boldsymbol{\mu}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$. By factoring this joint importance function as,

$$q(\boldsymbol{x}_t, i|\boldsymbol{y}_{1:t}) = q(i|\boldsymbol{y}_{1:t})q(\boldsymbol{x}_t|i, \boldsymbol{y}_{1:t}) \tag{2.25}$$

and by defining,

$$q(\boldsymbol{x}_t|i, \boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i) \tag{2.26}$$

and

$$q(i|\boldsymbol{y}_{1:t}) = p(\boldsymbol{y}_t|\boldsymbol{\mu}_t^i)w_{t-1}^i \tag{2.27}$$

the second stage weight update equation can be obtained as,

$$\begin{aligned} w_t^j &\propto w_{t-1}^i \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^j)p(\boldsymbol{x}_t^j|\boldsymbol{x}_{t-1}^{i^j})}{q(\boldsymbol{x}_t^j, i^j|\boldsymbol{y}_{1:t})} \\ &= \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^j)}{p(\boldsymbol{y}_t|\boldsymbol{\mu}_t^{i^j})} \end{aligned} \tag{2.28}$$

where the used sample pair $\{\boldsymbol{x}_t^j, i^j\}_{j=1}^N$ represents the propagated state $\boldsymbol{x}_t^j$ at time t, and the index i^j represents the particle trajectory at time $t-1$.

The steps of APF algorithm are as follows:

- for $i = 1 : N$
  - Calculate $\boldsymbol{\mu}_t^i$
  - Evaluate the first stage weights : $w_t^i \propto w_{t-1}^i p(\boldsymbol{y}_t|\boldsymbol{\mu}_t^i)$
- end
- Normalize the weights: $\widetilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}$
- Sample the indices: $[\{-,-,i^j\}_{j=1}^N] = resample[\{\boldsymbol{x}_t^i, \widetilde{w}_t^i\}_{i=1}^N]$
- Propagate and compute the second stage weights using the indices:
- for $j = 1 : N$
  - Draw $\boldsymbol{x}_t^j \sim p(\boldsymbol{x}_t|\boldsymbol{x}_t^{i^j})$
  - Compute the second stage weights: $\widetilde{w}_t^j = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^j)}{p(\boldsymbol{y}_t|\boldsymbol{\mu}_t^{i^j})}$
- end

The advantage of the APF method may be that, it generates particles from the trajectories at $t-1$ which are conditioned on the current observation. Therefore, they are more likely to be close to the true value of the state. We note that, the resampling in APF algorithm occurs earlier than in SPF algorithm, that is, it takes place before the propagation of the state (as opposed to resampling after the propagation of the state, for all other resampling algorithms). It also uses the point estimate $\boldsymbol{\mu}_t^i$, which characterizes the transition density $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$. This approach may improve the samples at the propagation step, and may generate particles from the high probability region of the posterior PDF. APF has been shown to have superior performance when the state noise is small.

Chapter 3

Particle Filtering and Static Parameters

3.1 DSS Models with Static Parameters

For the nonlinear and non-Gaussian DSS models with known static parameters $\boldsymbol{\theta}$, the PF algorithms are quite effective for sequential estimation of the dynamic state $\boldsymbol{x}_t$. If the parameter vector is also dynamic, the usual PF algorithm can be applied by including $\boldsymbol{\theta}_t$ as part of the state $\boldsymbol{x}_t$. However, when $\boldsymbol{\theta}$ is constant, there is no effective PF algorithm that can estimate simultaneously the joint posterior PDF of the state and parameter vector. The problem is that the lack of dynamics of $\boldsymbol{\theta}$ entails sample degeneracy of the random measures, which leads to poor performance of the particle filters. In statistical literature, static parameters have been successfully estimated by MCMC algorithms (Liu, 2001). However, the MCMC algorithms operate on batch data (off-line), and due to its slow convergence, they may not be suitable for sequential online estimation. In many real world problems, where observations are high frequency data which arrives sequentially in real-time, and the real-time estimation is desired, where the MCMC algorithms are not suitable.

There are several approaches for implementing PF on DSS models with unknown static parameters. One of them is by Gordon (Gordon et al., 1993), where the fixed parameters have an artificial evolution. Another approach, proposed by West (West, 1992, 1993a,b), which approximates the posterior PDF by the weighted kernel density based on the principle of modeling with mixtures. Storvik (Storvik, 2002) introduces a method for a special class of DSS models with static parameters, where the parameters are tractable with analytic method based on the sufficient statistics principle.

3.1.1 Artificial Evolution of Static Parameters

For the DSS model, Gordon (Gordon et al., 1993) suggested reduction of the sample degeneracy of the state by adding small random disturbances to the state particles between time steps, which is in addition to the system noise. This idea was then extended to the static model parameters. The model parameters $\boldsymbol{\theta}$, then, are indexed by t to show the artificial time dependence. For example, the model parameters $\boldsymbol{\theta}$ are modeled to evolve as according to a random walk, that is,

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \boldsymbol{\zeta}_t \tag{3.1}$$

where $\boldsymbol{\zeta}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\zeta)$ with $\boldsymbol{\Sigma}_\zeta$ being a known covariance matrix. With this model, the state vector is completely dynamic, and the PF algorithm can be applied in the usual way to get joint inference of the states and the parameters simultaneously. However, this approach has a drawback in that the approximated posterior PDF of the parameters is far more diffused than the true posterior PDF. The main reason is that, by making the static parameters dynamic with an artificial evolution, the "loss of information" occurred at each time instant. Therefore, it is clear that this approach cannot be optimal.

3.1.2 Weighted Kernel Density Method

West (West, 1992, 1993a,b) introduced an *adaptive importance sampling* algorithm to make a discrete approximation of the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$ with a *random measure*. In this algorithm, the posterior PDF is approximated with the weighted kernel density based on the principle of modeling a posterior PDF with mixtures of density. The mixture modeling technique has the flexibility to represent quite complex PDFs, where the mixture component density usually belongs to the exponential family. Here, the multivariate normal distributions are chosen as mixture component densities.

The approximation of the posterior PDF is carried out via the adaptive importance sampling algorithm with the use of the shrinkage technique. This adaptive mechanism modifies the initial crude approximation towards the true posterior distribution by dynam-

ical refinement of the mixture approximation. It avoids the information loss that occurs in the artificial evolution method. In low dimensional space, the algorithm seems to be quite effective in approximating the posterior PDF. For a chosen importance sampling density $q(\boldsymbol{\theta}_t)$, and with the sample size N and weights w_t^i, the true PDF $p(\boldsymbol{\theta}_t)$ can be approximated by the weighted kernel density as,

$$p(\boldsymbol{\theta}_t) \approx \sum_{i=1}^{N} w_t^i d_p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_t^i, h^2 V_t) \tag{3.2}$$

where $d_p(\boldsymbol{\theta}|m, M)$ is a p-variate elliptically symmetric density function centered at the mode m and is with scale matrix M. In our example, $d_p(\cdot) = \mathcal{N}_p(\boldsymbol{\theta}_t|\bar{\boldsymbol{\theta}}_t, h^2 V_t)$, is a multivariate normal PDF, where $\bar{\boldsymbol{\theta}}_t$ and V_t are the MC estimates of the weighted mean vector and the weighted covariance matrix respectively, and h is a smoothing parameter. The weight is defined as $w_t^i \propto \frac{p(\boldsymbol{\theta}_t^i)}{q(\boldsymbol{\theta}_t^i)}$.

For a DSS model, the Bayesian sequential filtering method is carried out in two stages: (1) prediction and (2) update. Suppose, at time $t-1$, we have particles $\{\boldsymbol{\theta}_{t-1}^i\}_{i=1}^N$ and their associated weights $\{w_{t-1}^i\}_{i=1}^N$, approximating the posterior PDF $p(\boldsymbol{\theta}_{t-1}|\boldsymbol{y}_{1:t-1})$, i.e.,

$$p(\boldsymbol{\theta}_{t-1}|\boldsymbol{y}_{1:t-1}) \approx \sum_{i=1}^{N} w_{t-1}^i \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}^i) \tag{3.3}$$

Note again that, the time index $t-1$ on $\boldsymbol{\theta}$ indicates only the posterior at time $t-1$. We compute the following moments from the weighted samples:

$$m_{t-1}^i = c\boldsymbol{\theta}_{t-1}^i + (1-c)\bar{\boldsymbol{\theta}}_{t-1} \tag{3.4}$$

$$V_{t-1} = \sum_{i=1}^{N} w_{t-1}^i (\boldsymbol{\theta}_{t-1}^i - \bar{\boldsymbol{\theta}}_{t-1})(\boldsymbol{\theta}_{t-1}^i - \bar{\boldsymbol{\theta}}_{t-1})^T \tag{3.5}$$

$$\bar{\boldsymbol{\theta}}_{t-1} = \sum_{i=1}^{N} w_{t-1}^i \boldsymbol{\theta}_{t-1}^i \tag{3.6}$$

At time step $t-1$, the predicted (prior) distribution is constructed as

$$p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t-1}) \approx \sum_{i=1}^{N} w_{t-1}^i N(\boldsymbol{\theta}_t|m_{t-1}^i, h^2 V_{t-1}) \tag{3.7}$$

$$where \quad h^2 = 1 - \left(\frac{3\delta - 1}{2\delta}\right)^2; c = \sqrt{(1-h^2)}; \quad \delta \in (0, 1] \tag{3.8}$$

where c is a shrinkage parameter, and h is smoothing parameter, the latter is used to control the overdispersion of the mixture kernel density. Then, at time step t, when the observation $\boldsymbol{y}_t$ becomes available, the updating of the prior is carried out via Bayes' rule.

3.2 Particle Filtering with Static Parameters: Liu-West Method

Liu and West (Liu and West, 2001) introduced an APF algorithm which performs simultaneous sequential estimation of the dynamic states and the static parameters of a DSS model, and hence, we call it LW method. The algorithm is based on the method from the previous subsection (3.1.2). The parameter vector is appended to the state vector, and a joint posterior of the enlarged state vector is estimated. The algorithm has been widely used and considered to be quite efficient. It is the Bayesian filtering algorithm and is carried out in two stages: *prediction* and *update*. The prediction is done at time step $t-1$, and the updating is performed at time step t after receiving the observation $\boldsymbol{y}_t$.

Suppose, at time $t-1$, we have combined particles of the state and parameters along with their associated weights from the predictive (prior) density. The joint predictive PDF at time $t-1$ is given by

$$\begin{aligned} Prediction : p(\boldsymbol{x}_t, \boldsymbol{\theta}_t | \boldsymbol{y}_{1:t-1}) &= \int p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \boldsymbol{\theta}_{t-1} | \boldsymbol{y}_{1:t-1}) \mathrm{d}\boldsymbol{x}_{t-1} \\ &= \int p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\theta}_{t-1}) p(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}_{t-1} | \boldsymbol{y}_{1:t-1}) \mathrm{d}\boldsymbol{x}_{t-1} \end{aligned} \quad (3.9)$$

At time t, as the observation $\boldsymbol{y}_t$ becomes available, the prior predictive distribution is updated via Bayes as

$$\begin{aligned} Update : p(\boldsymbol{x}_t, \boldsymbol{\theta}_t | \boldsymbol{y}_{1:t}) &\propto p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta}_t) p(\boldsymbol{x}_t, \boldsymbol{\theta}_t | \boldsymbol{y}_{1:t-1}) \\ &\propto p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta}_t) p(\boldsymbol{x}_t | \boldsymbol{\theta}_t, \boldsymbol{y}_{1:t-1}) p(\boldsymbol{\theta}_t | \boldsymbol{y}_{1:t-1}) \end{aligned} \quad (3.10)$$

Our objective is to obtain the weighted particles from the above posterior PDF for sequential estimation of states and parameters. The following APF algorithm provides the stepwise procedure in achieving this goal sequentially.

3.2.1 An APF Algorithm: LW Method

- Suppose, at time $t-1$, we have the particles $\{\boldsymbol{x}_{t-1}^i, \boldsymbol{\theta}_{t-1}^i\}_{i=1}^N$ and their associated weights $\{w_{t-1}^i\}_{i=1}^N$, which approximate the joint posterior PDF $p(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}_{t-1}|\boldsymbol{y}_{1:t-1})$
- At time t, we want to obtain the particles $\{\boldsymbol{x}_t^i, \boldsymbol{\theta}_t^i\}_{i=1}^N$ and their weights $\{w_t^i\}_{i=1}^N$, that approximate the joint posterior PDF $p(\boldsymbol{x}_t, \boldsymbol{\theta}_t|\boldsymbol{y}_{1:t})$

1. The prior estimates of $\{\boldsymbol{x}_{t-1}^i, \boldsymbol{\theta}_{t-1}^i\}$ are given by $\{\hat{\boldsymbol{\mu}}_t^i, \mathbf{m}_{t-1}^i\}$, $i = 1, \ldots, N$, where,
   - $\hat{\boldsymbol{\mu}}_t^i = E[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i, \boldsymbol{\theta}_{t-1}^i]$
   - $\mathbf{m}_{t-1}^i = c\boldsymbol{\theta}_{t-1}^i + (1-c)\bar{\boldsymbol{\theta}}_{t-1}$
   - $\bar{\boldsymbol{\theta}}_{t-1} = \sum_{i=1}^N w_{t-1}^i \boldsymbol{\theta}_{t-1}^i$
   - $\mathbf{V}_{t-1} = \sum_{i=1}^N w_{t-1}^i (\boldsymbol{\theta}_{t-1}^i - \bar{\boldsymbol{\theta}}_{t-1})(\boldsymbol{\theta}_{t-1}^i - \bar{\boldsymbol{\theta}}_{t-1})^T$
2. Compute the fist-stage weights: $L_t^i \propto w_{t-1}^i p(\boldsymbol{y}_t|\hat{\boldsymbol{\mu}}_t^i, \mathbf{m}_{t-1}^i), \quad i = 1, \ldots, N$
3. Normalize the weights: $\widetilde{L}_t^i = \frac{L_t^i}{\sum_{j=1}^N L_t^j}, \quad i = 1, \ldots, N$
4. Resampling: sample the indexes $J_i \in (1, \ldots, N)$ with $\widetilde{L}_t$, J_i are auxiliary variables.
5. Propagate the parameter vector: $\boldsymbol{\theta}_t^i \sim N(\boldsymbol{\theta}_t|\mathbf{m}_{t-1}^{J_i}, h^2\mathbf{V}_{t-1}), \quad i = 1, \ldots, N$
6. Propagate the state : $\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{J_i}, \boldsymbol{\theta}_t^i), \quad i = 1, \ldots, N$
7. Evaluate the 2nd-stage weights: $w_t^i \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^i, \boldsymbol{\theta}_t^i)}{p(\boldsymbol{y}_t|\hat{\boldsymbol{\mu}}_t^{J_i}, \mathbf{m}_t^{J_i})}, \quad i = 1, \ldots, N$
8. Normalize the weights: $\widetilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}, \quad i = 1, \ldots, N$
9. The posterior at time t is represented by : $\{\boldsymbol{x}_t^i, \boldsymbol{\theta}_t^i, \widetilde{w}_t^i\}_{i=1}^N \sim p(\boldsymbol{x}_t, \boldsymbol{\theta}_t|\boldsymbol{y}_{1:t})$

3.3 Application to Stochastic Volatility (SV) Models

Recall the SV model described in Chapter 1 as,

$$x_t = \alpha + \beta x_{t-1} + \sigma_u u_t \quad (state \quad equation) \tag{3.11}$$

$$y_t = e^{x_t/2} v_t \quad (observation \quad equation) \tag{3.12}$$

where, $x_t \in \mathbb{R}$ is a hidden state that represents the logarithm of the volatility, and $u_t \in \mathbb{R}$ and $v_t \in \mathbb{R}$ are uncorrelated Gaussian noise processes with zero means and unit variances. As a filtering problem, we want to estimate the state x_t sequentially based on the observations $y_{1:t}$. However, the static parameters (α, β, and σ_u^2) of the state equation are unknown.The observation equation does not have any unknown parameters. The state equation is linear and Gaussian, whereas the observation equation is nonlinear and non-stationary process. Therefore, the application of PF method is the most appropriate for sequential inferences of the hidden state x_t. Let the vector of unknown static parameters be $\boldsymbol{\theta} = (\alpha \ \beta \ \sigma_u^2)^\top$. For the distributions related to the state and observation equations, we write

$$p(x_t|x_{t-1}, \boldsymbol{\theta}) \sim \mathcal{N}(\alpha + \beta x_{t-1}, \sigma_u^2) \tag{3.13}$$

$$p(y_t|x_t) \sim \mathcal{N}(0, e^{x_t}) \tag{3.14}$$

We have applied the APF algorithm of Liu and West for the above SV model. In implementing this algorithm, we transformed σ_u^2 as, $\gamma = \log(\sigma_u^2)$ to be on the real line. Therefore, the transformed parameter space is, $\boldsymbol{\theta} = (\alpha \ \beta \ \gamma)^\top$, and the algorithm is carried out as in section (3.2.1). We have tested it on simulated datasets with $T = 1200$ sample points. The parameter values used for the simulated datasets in accordance with the literature on the SV models on the $S\&P500$ index by Jacquier et al. (1994). The simulation study was conducted with two datasets with different state noise variances, $\sigma_u^2 = 0.05$ and $\sigma_u^2 = 0.10$. The *dataset 1* was generated with ($\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05$), and the *dataset 2* was generated with ($\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10$). The APF algorithm was applied on both datasets with increased particle sizes, $N = 100, 500, 5000$ and 10000, but we have presented the graphs only with $N = 100, 5000$.

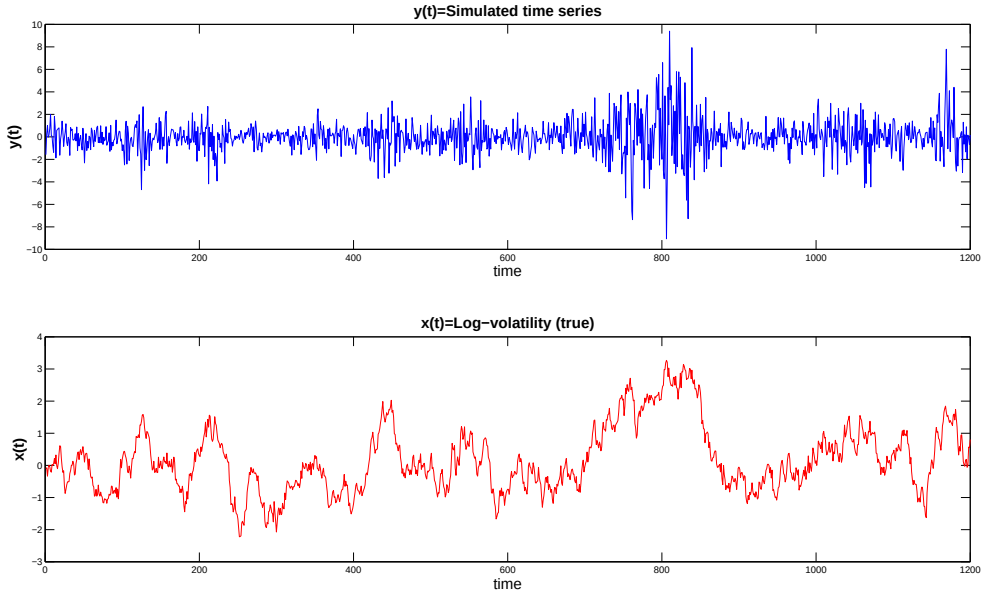

Figure 3.1: **Dataset 1** : Simulated time series y_t, and the underlying log-volatility x_t, with parameters; $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05,$ and $T = 1200$.

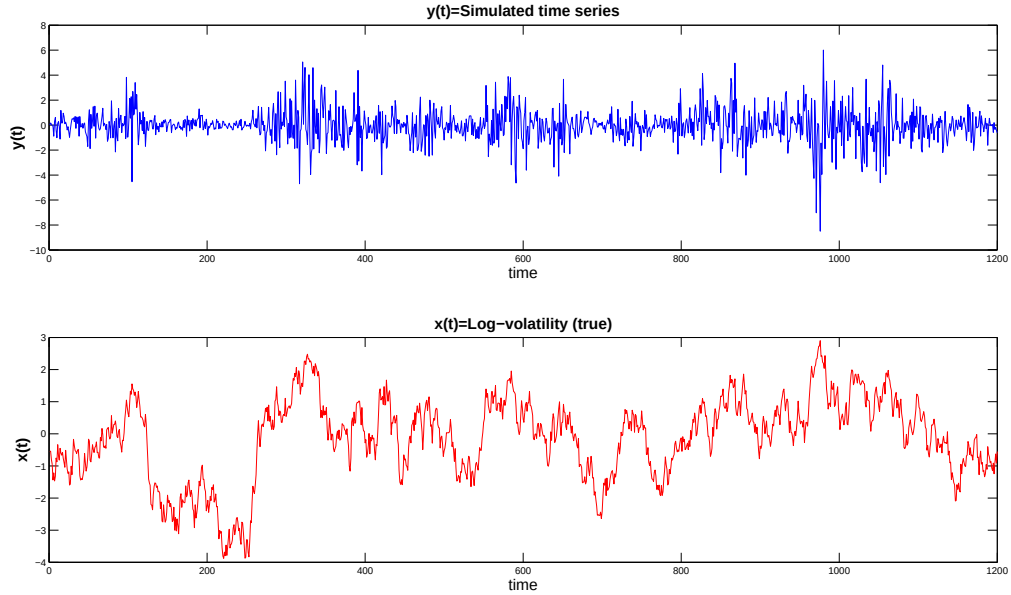

Figure 3.2: **Dataset 2** : Simulated time series y_t, and the underlying log-volatility x_t, with parameters; $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10,$ and $T = 1200$.

Note that *dataset 1* corresponds to smaller state noise variance, $\sigma_u^2 = 0.05$ than *dataset 2*, where $\sigma_u^2 = 0.10$. Both datasets used the constant level parameter $\alpha = -0.005$, and persistence parameters $\beta = 0.98$. Persistence in the variance evolves over time, is the momentum in the conditional variance, i.e., the past volatility explains the present volatility. The empirical study suggests that the volatility process follows the properties of persistence. The parameter $\beta = 0.98$ corresponds to high persistence or high clustering.

The following are the simulation results estimating hidden state x_t, and the parameters with the application of LW-APF method on both datasets (dataset 1 and dataset 2). We observe that, the accuracy of the estimate is improved with increased particle size, N. With the particle size $N = 5000$ or above, the accuracy is quiet good.

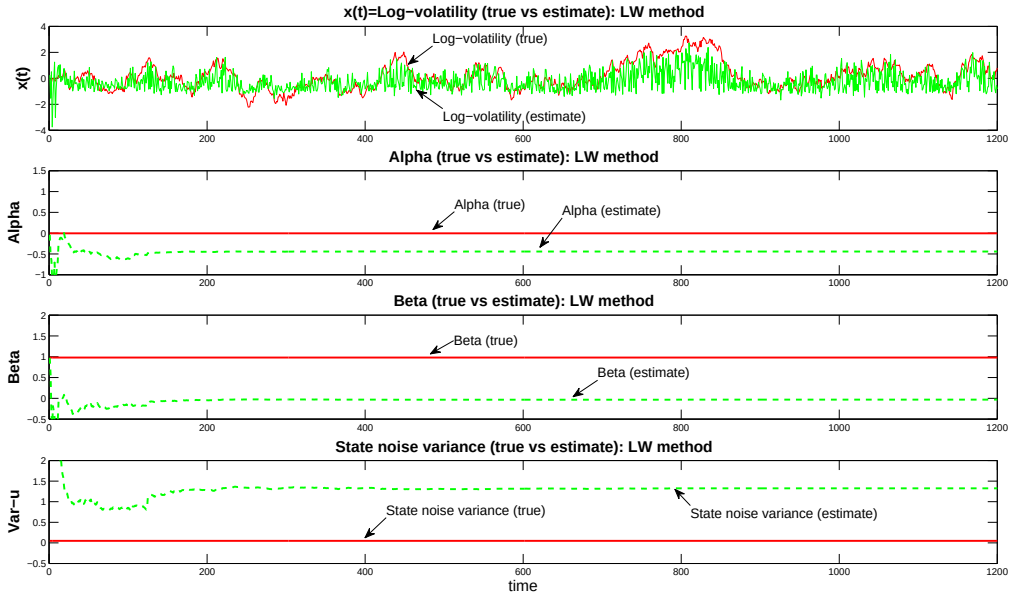

Figure 3.3: **Particle size N=100**; estimate of log-volatility & parameters: **dataset 1**

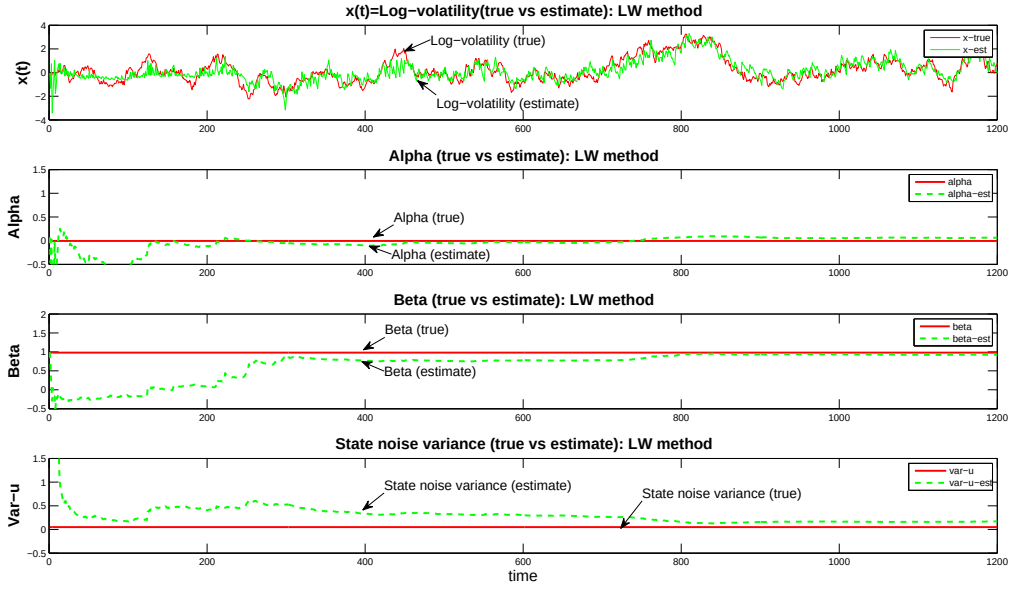

Figure 3.4: **Particle size N=5000**; estimate of log-volatility & parameters: **dataset 1**

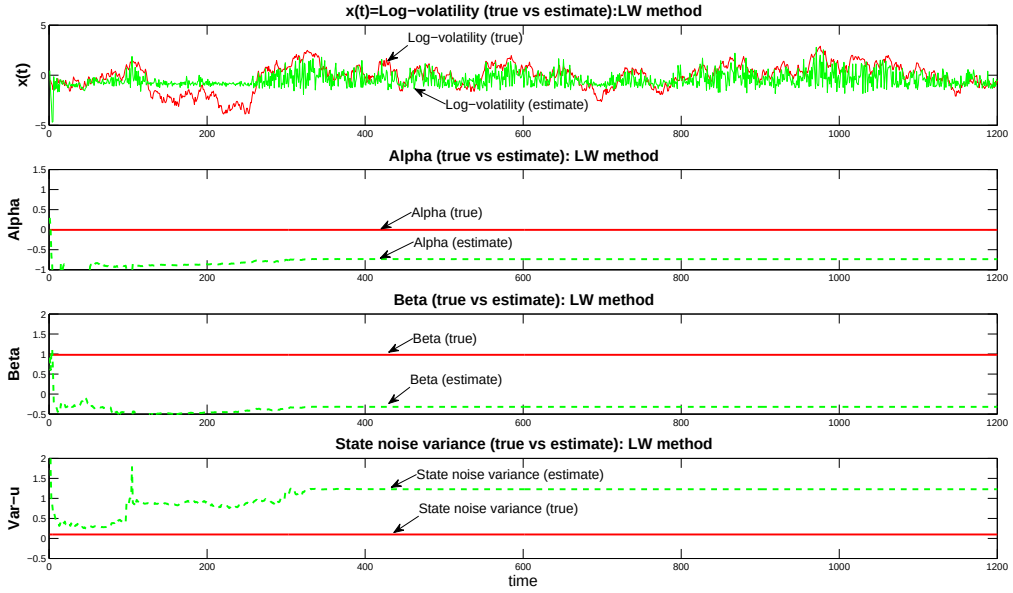

Figure 3.5: **Particle size N=100**; estimate of log-volatility & parameters: **dataset 2**

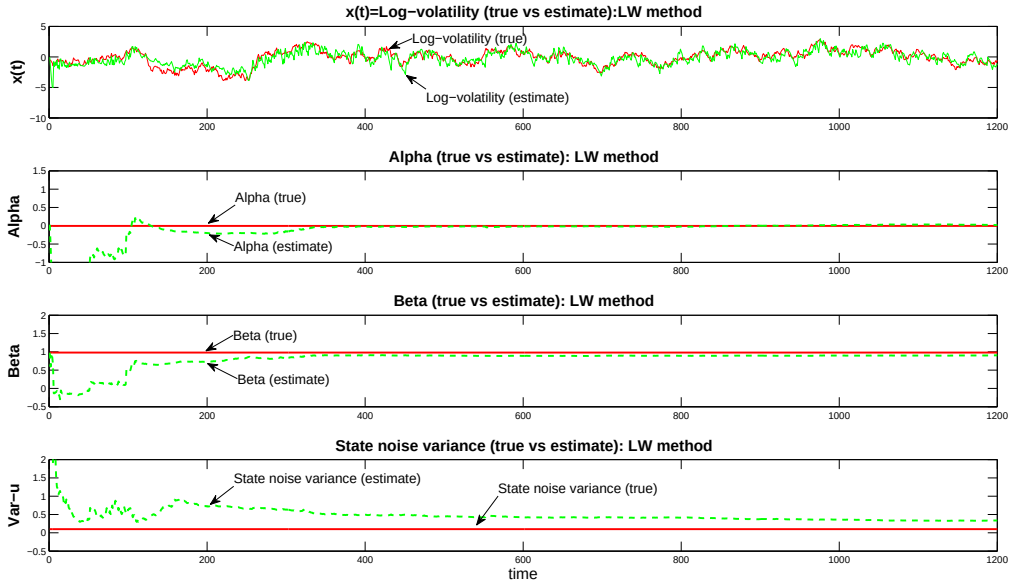

Figure 3.6: **Particle size N=5000**; estimate of log-volatility & parameters: **dataset 2**

Chapter 4

Particle Filtering on Rao-Blackwellized DSS Models

4.1 Rao-Blackwellization: Reducing Uncertainty

The application of Rao-Blackwell theorem transforms an arbitrary estimator into an improved estimator according to the *mean squared error* (MSE) criterion. We refer to the application of the Rao-Blackwell theorem as *Rao-Blackwellization* (RB). Recall that Rao-Blackwell theorem (Lehmann, 1991) states that, if $\hat{\theta}(\boldsymbol{y})$ is an estimate of θ, and $T(\boldsymbol{y})$ is a sufficient statistic for θ, then the estimate $\hat{\theta}^*(\boldsymbol{y}) = E[\hat{\theta}(\boldsymbol{y})|T(\boldsymbol{y})]$ is at least as good as $\hat{\theta}(\boldsymbol{y})$ in terms of the MSE, that is,

$$\forall\theta \qquad MSE(\hat{\theta}^*) \leq MSE(\hat{\theta}) \tag{4.1}$$

In essence, with the use of the estimator $\hat{\theta}^*(\boldsymbol{y})$, we reduce the variance or uncertainty of the unknown's estimate. The analytic integration produces no errors, but any stochastic approximation, by its nature cannot be done without errors. When there are no analytical solutions to the integration problems, the application of MC sampling techniques to approximate the integral problem contains numerical error. MC simulation method under Bayes' theorem approximates a posterior PDF by the use of discrete *random measures*. If the dimension of the unknown space is smaller, then it is obvious that one would need a smaller set of particles to approximate the posterior PDF to achieve a certain accuracy. As the dimension of the unknown space increases, the approximation of the integration becomes worse for a fixed number of particles (Evans and Swartz, 2000). So, it pays off considerably if one can reduce the dimension of the unknown space where integration is carried out.

Suppose, in applying PF, we are only interested in estimating the posterior PDF of the state $p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t})$, but the DSS model also has unknown static parameters $\boldsymbol{\theta}$. Hence, we have two options. One, is to obtain the random measure by MC sampling that approximates the full posterior PDF $p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t})$, which means that we have to work in a higher dimensional space of unknowns, $\mathbb{R}^{n_x+n_\theta}$, because of the inclusion of the unknown parameter vector $\boldsymbol{\theta}$ with the state. And, the other option is to integrate out $\boldsymbol{\theta}$ analytically, which would reduce the dimension of the unknowns, and would lead to direct approximation of the marginalized posterior PDF $p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t})$. Clearly, the second option is preferable, because it would yield better accuracy. In other words, the analytic integration of the constant parameters reduces the dimension of the sampling space by marginalizing the posterior distribution, and consequently, it reduces the numerical errors that arise due to approximation of the posterior distributions with random measures. Besides, it also reduces the computational complexity of the method. When the parameters are conditionally linear, this integration is carried out with the Kalman filter. This hybrid method of partial integration, and the use of random measure to approximate the posterior PDFs is known as RB of the MC simulation method (Casella and Robert, 1996; Liu and Chen, 1998; Doucet et al., 2000; Storvik, 2002; Maskell, 2004).

4.2 Rao-Blackwellization on DSS Models: A Novel Approach

We consider a class of DSS models with unknown static parameters, where the state equation is linear and Gaussian. The observation equation can have any form (linear or nonlinear). We apply RB method on this class of DSS models. For convenience, we rewrite the model as

$$\boldsymbol{x}_t \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) \qquad (state \quad equation) \tag{4.2}$$

$$\boldsymbol{y}_t \sim p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}) \qquad (observation \quad equation) \tag{4.3}$$

where all the symbols have already been defined. Our main interest is to sequentially estimate the hidden states $\boldsymbol{x}_t$, i.e., getting inference from the filtering PDF $p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t})$.

Due to the presence of unknown static parameters $\boldsymbol{\theta}$, we have to deal with the full posterior PDF $p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t})$, which requires MC sampling from both the state and parameter spaces. However, if we can integrate out the nuisance parameters $\boldsymbol{\theta}$, the resultant distribution is the desired filtering PDF $p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t})$, and which is the reduction in sampling space. In our work, the reduction in sampling space with the application of RB was implemented with *implied integration method* as an alternative approach. The RB requires to obtain the marginal distribution of the dynamic states by integrating out the parameters, i.e.,

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}) &= \int_{\boldsymbol{\theta}} p(\boldsymbol{x}_t, \boldsymbol{\theta}|\boldsymbol{x}_{0:t-1}) \mathrm{d}\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\theta}} p(\boldsymbol{x}_t|\boldsymbol{\theta}, \boldsymbol{x}_{0:t-1}) p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t-1}) \mathrm{d}\boldsymbol{\theta} \end{aligned} \tag{4.4}$$

and similarly, the marginalized likelihood by integrating out the parameters,

$$\begin{aligned} p(\boldsymbol{y}_t|\boldsymbol{x}_t) &= \int_{\boldsymbol{\theta}} p(\boldsymbol{y}_t, \boldsymbol{\theta}|\boldsymbol{x}_{0:t}) \mathrm{d}\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\theta}} p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{x}_{0:t}) p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t}) \mathrm{d}\boldsymbol{\theta} \end{aligned} \tag{4.5}$$

where we assume that the integral corresponding to the observation equation is analytically solvable. Thus, the implementation of the RB method requires that the integrals (4.4) and (4.5) are analytically solvable. Avoiding direct integration is preferable, since it is difficult to implement with higher dimension. With the use of the implied integration method, where we express the posterior PDF of $\boldsymbol{\theta}$ with Bayes theorem. Here, we assume that given $\boldsymbol{x}_{0:t}$, $\boldsymbol{\theta}$ is not a function of $\boldsymbol{y}_{1:t}$, and can be expressed as,

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t}) &= \frac{p(\boldsymbol{\theta}, \boldsymbol{x}_{0:t})}{p(\boldsymbol{x}_{0:t})} \\ &= \frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{\theta}) p(\boldsymbol{x}_{0:t-1}, \boldsymbol{\theta})}{p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}) p(\boldsymbol{x}_{0:t-1})} \\ &= \frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{\theta}) p(\boldsymbol{x}_{0:t-1}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}) p(\boldsymbol{x}_{0:t-1})} \\ &= \frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t-1})}{p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1})} \end{aligned}$$

Hence, we can deduce that

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}) = \frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1},\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t-1})}{p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t})} \tag{4.6}$$

The desired expression in (4.6) is free from $\boldsymbol{\theta}$, and can be obtained after evaluation of the right hand side of the equation. This procedure avoids direct integration if the complete posterior PDF of $\boldsymbol{\theta}$ are known, and this relationship is also known as the candidate's formula (Besag, 1889). The expression in (4.6) can be used as the proposal density for $\boldsymbol{x}_t$.

The Rao-Blackwellized filtering density of $\boldsymbol{x}_t$ can be written as,

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t}) &= \int_{\Theta} p(\boldsymbol{x}_t,\boldsymbol{\theta}|\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t})\mathrm{d}\boldsymbol{\theta} \\ &= \int_{\Theta} p(\boldsymbol{x}_t|\boldsymbol{\theta},\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t})p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t})\mathrm{d}\boldsymbol{\theta} \\ &\propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)\int_{\Theta} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{\theta})p(\boldsymbol{x}_{0:t-1},\boldsymbol{\theta}|\boldsymbol{y}_{1:t})\mathrm{d}\boldsymbol{\theta} \\ &= p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}) \end{aligned} \tag{4.7}$$

This results avoid sampling on the parameter space, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$, and allows to generate particles only on the desired state space $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$. In the next section, we exploit (4.6) in applying RB to DSS models related to SV.

4.3 Application to Stochastic Volatility (SV) Models

We rewrite the SV model as,

$$x_t = \alpha + \beta x_{t-1} + \sigma_u u_t \quad (state \quad equation) \tag{4.8}$$

$$y_t = e^{x_t/2} v_t \quad (observation \quad equation) \tag{4.9}$$

where all the variables have already been defined. We want to apply RB by integrating out the unknown static parameters $\boldsymbol{\theta} = (\alpha\ \beta\ \sigma_u^2)^\top$ from the state equation. The implementation of RB method marginalizes the full posterior PDF $p(x_t, \boldsymbol{\theta}|x_{0:t-1}, y_{1:t})$ to $p(x_t|x_{0:t-1}, y_{1:t})$. As pointed out earlier, we use the implied integration technique to implement the RB.

With the distributional assumptions of the state and observation equations, (4.8)-(4.9) can be expressed as,

$$p(x_t|x_{t-1},\boldsymbol{\theta}) \sim \mathcal{N}(\alpha+\beta x_{t-1},\sigma_u^2) \tag{4.10}$$

$$p(y_t|x_t) \sim \mathcal{N}(0,e^{x_t}) \tag{4.11}$$

With the use of *implied integration* method, RB is carried out as follows:

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}) &= \frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1},\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t-1})}{p(\boldsymbol{\theta}|\boldsymbol{x}_{0:t})} \\ &= \frac{\overbrace{p(x_t|x_{0:t-1},\boldsymbol{\theta})}^{(1)}\overbrace{p(\boldsymbol{\theta}|x_{0:t-1})}^{(2)}}{\underbrace{p(\boldsymbol{\theta}|x_{0:t})}_{(3)}} \qquad (4.12) \\ &\propto t_{\nu_{t-1}}(m_{t-1},r_{t-1}) \qquad (4.13) \end{aligned}$$

After evaluation of the RHS of (4.12), the $p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}) \sim t_{\nu_{t-1}}(m_{t-1},r_{t-1})$, is a non-standard *t-distribution* with ν_{t-1} degrees of freedom, m_{t-1} and r_{t-1} are the mean and variance respectively (see appendix). The m_{t-1} and r_{t-1} are estimated from $\boldsymbol{x}_{0:t-1}$ with Sequential Least Squares (SLS) method, as state equation is linear.

From (4.12), we have

$$\begin{aligned} (1)\quad p(x_t|x_{0:t-1},\boldsymbol{\theta}) &\sim N(\alpha+\beta x_{t-1},\sigma_u^2) \qquad (4.14) \\ with\quad E(x_t|x_{0:t-1},\boldsymbol{\theta}) &= \alpha+\beta x_{t-1} \\ and\quad Var(x_t|x_{0:t-1},\boldsymbol{\theta}) &= \sigma_u^2 \end{aligned}$$

$$\begin{aligned} (2)\quad p(\boldsymbol{\theta}|x_{0:t-1}) &\propto \underbrace{p(x_{0:t-1}|\boldsymbol{\theta})}_{likelihood}\underbrace{p(\boldsymbol{\theta})}_{prior} \\ &= p(x_{0:t-1}|\boldsymbol{\theta})p(\alpha\ \beta\ \sigma_u^2) \\ &= \underbrace{p(x_{0:t-1}|\boldsymbol{\theta})}_{Gaussian}\underbrace{p(\alpha\ \beta)}_{constant}\underbrace{p(\sigma_u^2)}_{\mathcal{IG}(a,\,b)} \qquad (4.15) \end{aligned}$$

and similarly

$$
\begin{aligned}
(3) \quad p(\boldsymbol{\theta}|x_{0:t}) &\propto \underbrace{p(x_{0:t}|\boldsymbol{\theta})}_{likelihood}\underbrace{p(\boldsymbol{\theta})}_{prior} \\
&= p(x_{0:t}|\boldsymbol{\theta})p(\alpha\ \beta\ \sigma_u^2) \\
&= \underbrace{p(x_{0:t}|\boldsymbol{\theta})}_{Gaussian}\underbrace{p(\alpha\ \beta)}_{constant}\ \underbrace{p(\sigma_u^2)}_{\mathcal{IG}(a,\,b)} \qquad (4.16)
\end{aligned}
$$

where $\mathcal{IG}$ stands for *Inverse-Gamma* distribution.

For the Rao-Blackwellized filtering density of x_t,

$$
\begin{aligned}
p(x_t|x_{0:t-1}, y_{1:t}) &= \int_\Theta p(x_t, \boldsymbol{\theta}|x_{0:t-1}, y_{1:t})\mathrm{d}\boldsymbol{\theta} \\
&= \int_\Theta p(x_t|\boldsymbol{\theta}, x_{0:t-1}, y_{1:t})p(\boldsymbol{\theta}|x_{0:t-1}, y_{1:t})\mathrm{d}\boldsymbol{\theta} \\
&\propto p(y_t|x_t)\int_\Theta p(x_t|x_{0:t-1}, \boldsymbol{\theta})p(x_{0:t-1}|\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \\
&\propto \underbrace{p(y_t|x_t)}_{N(y_t|0,e^{x_t})}\ \underbrace{p(x_t|x_{0:t-1})}_{t_{\nu_{t-1}}(x_t|m_{t-1},r_{t-1})} \qquad (4.17)
\end{aligned}
$$

This results allows us to generate particles for the state x_t without having to generate particles for $\boldsymbol{\theta}$. Thus, instead of exploring a four-dimensional space with particles, we explore only a one-dimensional state space.

Finally, if we choose an importance function $q(x_t|x_{0:t-1}) = p(x_t|x_{0:t-1})$, the weight update equation becomes as,

$$
\begin{aligned}
w_t &\propto w_{t-1}\frac{p(x_t|x_{0:t-1}, y_{1:t})}{q(x_t|x_{0:t-1}, y_{1:t})} \\
&\propto w_{t-1}\frac{p(y_t|x_t)p(x_t|x_{0:t-1})}{p(x_t|x_{0:t-1})} \\
&\propto w_{t-1}p(y_t|x_t) \qquad (4.18)
\end{aligned}
$$

4.4 Standard PF on Rao-Blackwellized SV Models

We implemented SPF using the RB SV models, where the importance function was $q(x_t|x_{0:t-1}) = p(x_t|x_{0:t-1})$. The algorithm samples only from the space of $x_t \in \mathbb{R}$, without sampling from the space of $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$. The SPF is actually the SIS filter, and we implemented it on simulated data with different particle sizes N. The following are the steps of the algorithm:

1. *Time step: t-1*

   - We have particles and their associated weights, $\{x^i_{0:t-1}, w^i_{t-1}\}^N_{i=1}$

   Each sequence of particles is also associated with a posterior PDF $p(\boldsymbol{\theta}|x^i_{0:t-1})$

2. *Importance sampling step*

   - Sample $\widetilde{x}^i_t \sim p(x_t|x^i_{0:t-1})$ and set $\widetilde{x}^i_{0:t} = (x^i_{0:t-1}, \widetilde{x}^i_t), i = 1, \ldots, N$
   - Evaluate importance weights, $w^i_t = w^i_{t-1} p(y_t|\widetilde{x}^i_t), i = 1, \ldots, N$
   - Normalize importance weights: $\widetilde{w}^i_t = \frac{w^i_t}{\sum^N_{j=1} w^j_t}, \quad i = 1, \ldots, N$

3. *Selection step*

   - Resample with replacement particles $\{x^i_{0:t}\}^N_{i=1}$ from $\{\widetilde{x}^i_{0:t}\}^N_{i=1}$ according to importance weights $\{\widetilde{w}^i_t\}^N_{i=1}$
   - If resampling takes place, set $\widetilde{w}^i_t = 1/N$

4. *Update step*

   - Update the parameters ν^i_t, m^i_t and r^i_t of $p(x_t|x_{0:t-1})$ as a non-standard *t-distribution* with the available state sequences $({x_{0:t}}^i)$
   - Increase $t-1$ to t and go to step 2

4.5 Auxiliary PF on Rao-Blackwellized SV Models

We also implemented the APF algorithm on the Rao-Blackwelized SV models. Again, it was applied on simulated datasets with different particle sizes N. The following are the steps of the APF algorithm:

- At time t-1, we have the particles and their weights: $\{x^i_{0:t-1}, w^i_{t-1}\}^N_{i=1}$
- Each sequence of particles is associated with a posterior PDF $p(\boldsymbol{\theta}|x^i_{0:t-1})$
- At time t, we obtain the particles $\{x^i_t\}^N_{i=1}$ and their weights $\{w^i_t\}^N_{i=1}$

1. The prior estimate of $\{x^i_{t-1}\}$ is given by $\{m^i_{t-1}\}, i = 1, \ldots, N$
   where,
   - $m^i_{t-1} = E[x_t|x^i_{t-1}, \boldsymbol{\theta}^i_{t-1}]$
2. Compute the first-stage weights: $L^i_t \propto w^i_{t-1} p(y_t|m^i_{t-1}), \quad i = 1, \ldots, N$
3. Normalize the weights: $\widetilde{L}^i_t = \frac{L^i_t}{\sum^N_{j=1} L^j_t}, \quad i = 1, \ldots, N$
4. Resampling: sample the indexes $J_i \in (1, \ldots, N)$ with $\widetilde{L}_t$, J_i are auxiliary variable indexes used to resample at time $t-1$, before the propagation.
5. Propagate the state: $x^i_t \sim p(x_t|x^{J_i}_{0:t-1}), \quad i = 1, \ldots, N$
6. Evaluate the 2nd-stage weights: $w^i_t \propto \frac{p(y_t|x^i_t)}{p(y_t|m^{J_i}_t)}, \quad i = 1, \ldots, N$
7. Normalize the weights: $\widetilde{w}^i_t = \frac{w^i_t}{\sum^N_{j=1} w^j_t}, \quad i = 1, \ldots, N$
8. Update the parameters ν^i_t, m^i_t and r^i_t of $p(x_t|x_{0:t-1})$ as a non-standard *t-distribution*, $t_{\nu_t}(x_t|m_t, r_t)$ with the available state sequences $({x_{0:t}}^i)$
9. The posterior at time t is approximated by $\{x^i_t, \widetilde{w}^i_t\}^N_{i=1}$

4.6 Simulation Results

We conducted extended simulations to compare the performance of the proposed methods, the Rao-Blackwellized SPF (RB-SPF) and the Rao-Blackwellized APF (RB-APF) algorithms on data generated according to the SV model. The performances of our algorithms were also compared to the Liu and West's (LW) method as described in Chapter 3. We used different criteria for performance measures including the effects of particle size on the estimation, and the *root mean squared error* (RMSE) of the state estimation (log-volatility).

We generated data with different lengths and with different state noise variances. For comparative analysis of RB and LW method, we have conducted simulated experiments on two types of datasets. In type-one, four datasets were simulated, where each consist of a single realization with different data lengths ($T = 1200, 2400$), and with different state noise variances, ($\sigma_u^2 = 0.05, 0.10$). In type-two, four different datasets were simulated, where each set comprised of $K = 100$ random realizations, with different data lengths ($T = 1200, 2400$) and different state noise variances, ($\sigma_u^2 = 0.05, 0.10$). The parameters used in the simulation are from the literatures by Jacquier et al. (1994); Kim et al. (1998) and Stroud et al. (2004).

More specifically, the parameters used to generate datasets are as follows:

For **type-one** datasets (single realization),

Dataset 1a: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 1200$,

Dataset 1: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 1200$,

Dataset 2 : $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 1200$,

Dataset 3: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 2400$, and

Dataset 4 : $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 2400$.

and for **type-two** datasets (multiple realizations),

Dataset 1R: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 1200, K = 100$,

Dataset 2R : $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 1200, K = 100$,

Dataset 3R: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 2400, K = 100$, and

Dataset 4R : $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 2400, K = 100$.

The simulated results with the application of RB vs LW methods on four datasets with the single realization are shown in Figures 4.1-4.14. We have shown in Figures 4.1-4.4 that the RB-APF algorithm performs better than the RB-SPF, as measured in RMSEs with the datasets 1a and 2. We compared the performances of RB-APF vs LW-APF methods on the datasets 1-4 (with single realization), and the results are shown in Figures 4.5-4.14. With extensive simulation study on datasets 1-4 with various particle size (N=100-10000), we are able to show that the RB method performs better than LW method with smaller particle size. However, as the particle size increases, the LW method shows equivalent performance as the RB method with respect to accuracy, as presented in Figures 4.5-4.6. In Figures 4.7-4.14, we observe that RB method consistently outperform the LW method, as demonstrated with the log(RMSE) with respect to estimate log-volatility (x_t). To obtain statistical inferences on the performances of the RB method, we have conducted the RMSE analysis on the datasets 1R-4R (*type-two*), where each dataset consist of 100 random realizations. We apply the RB-APF vs LW-APF methods on these datasets with increasing particle sizes (N=100-10000). The simulation results are shown in Figures 4.15-4.30, which are statistically significant. We observe that the accuracy attained by the RB-APF algorithm with the particle sizes of $100-500$ is equivalent to the accuracy attained by the LW algorithm with the particle size of 5000, a significant advantage. The log (RMSE) and its CDF plots of each dataset show that the RB-APF algorithm consistently outperforms the LW algorithm, especially with the smaller particle size.

The RMSEs for the datasets with *single realization* (type-one) were computed as,

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2} \tag{4.19}$$

And, the RMSEs for the datasets with *multiple realizations* (type-two) were computed,

$$RMSE(t) = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(x_t^k - \hat{x}_t^k)^2} \tag{4.20}$$

Simulation results with SPF vs APF algorithms on datasets 1a and 2

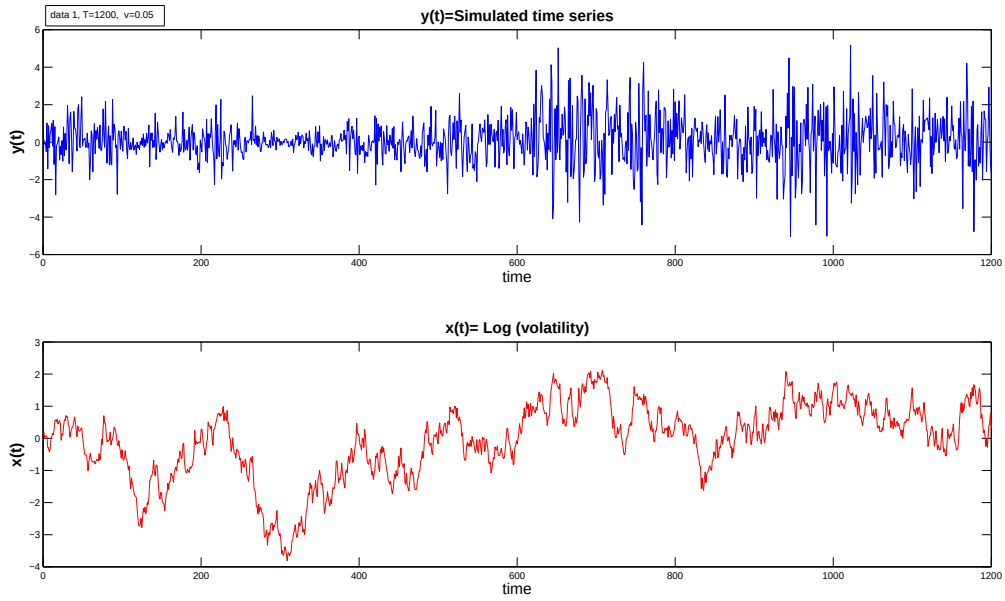

Figure 4.1: **Dataset 1a**: Simulated time series y_t, and underlying log-volatility x_t, with the parameters; $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05$ and $T = 1200$.

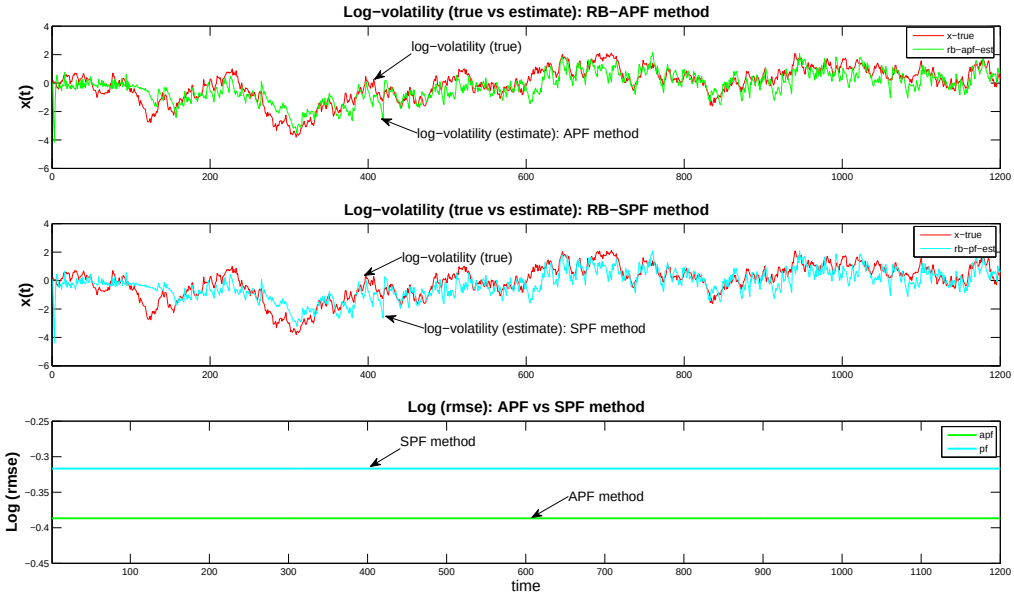

Figure 4.2: Estimate of log-volatility with log (RMSE): SPF vs APF method

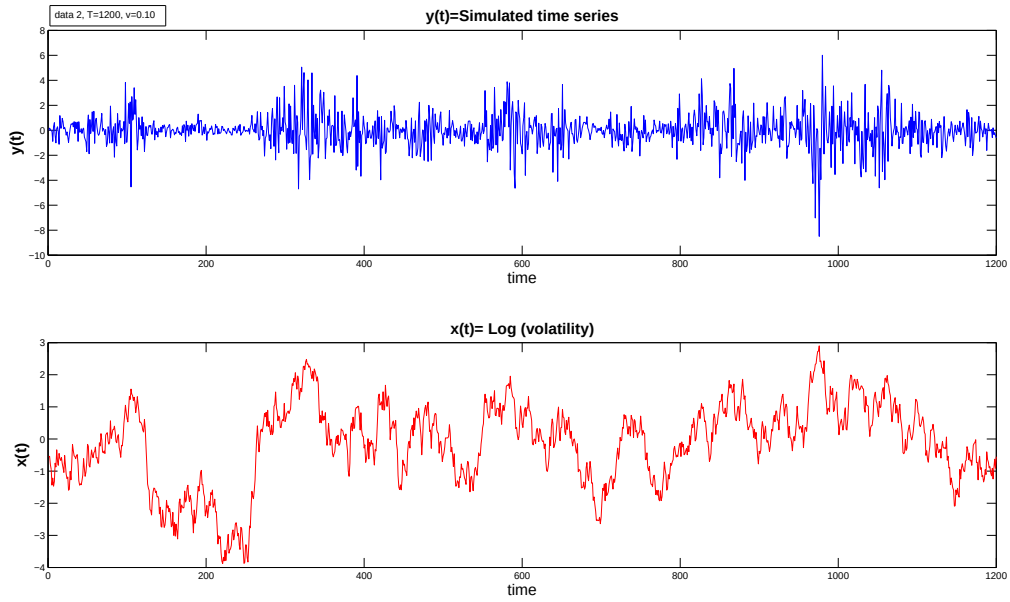

Figure 4.3: **Dataset 2**: Simulated time series y_t, and underlying log-volatility x_t, with the parameters; $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10$ and $T = 1200$.

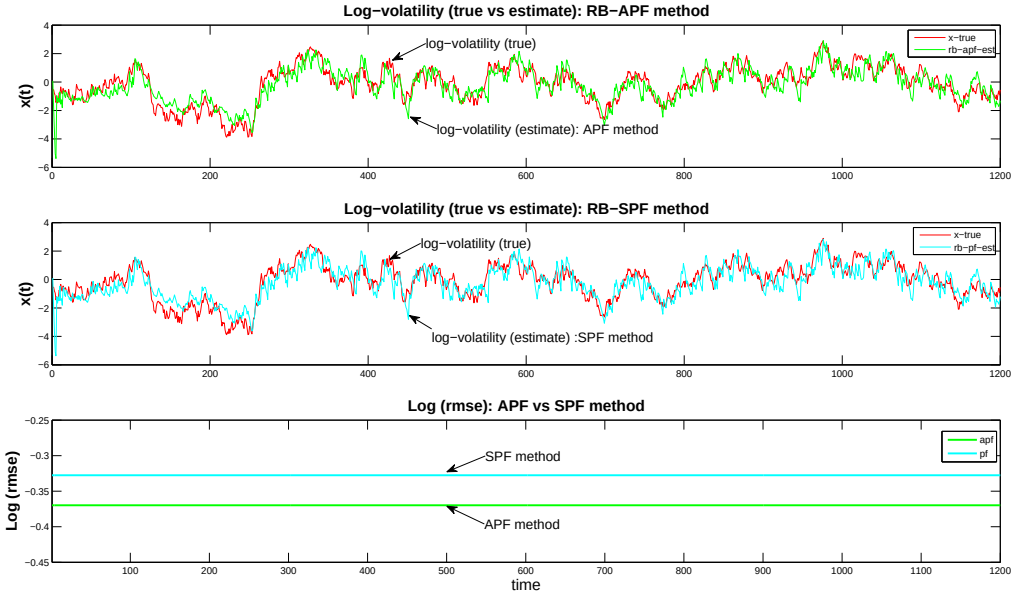

Figure 4.4: Estimate of log-volatility with log (RMSE): SPF vs APF method

Simulation results with LW-APF vs RB-APF algorithms on datasets 1-4

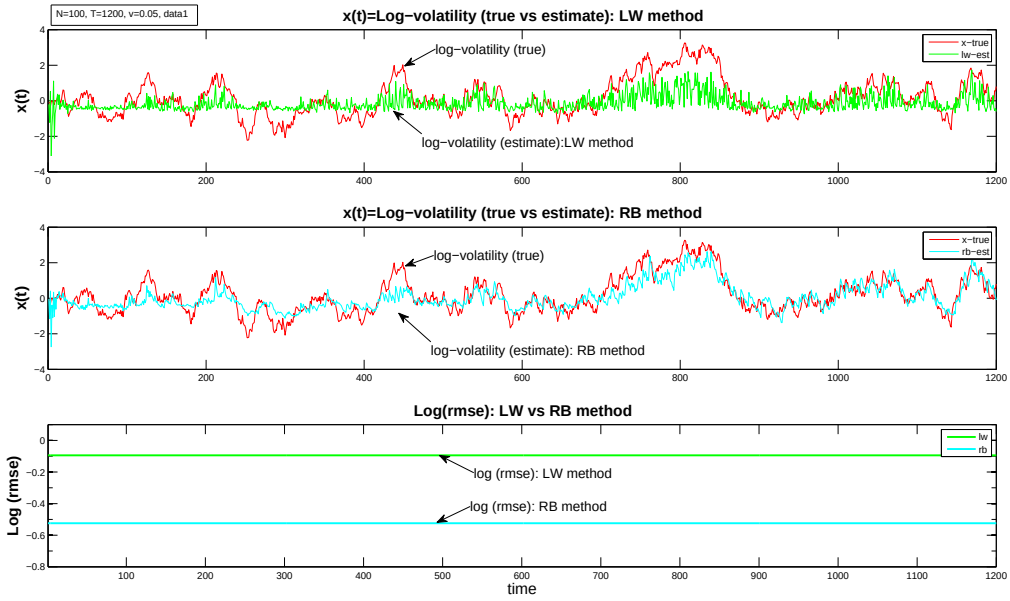

Figure 4.5: **Particle size N=100**; estimate of log-volatility & rmse : **dataset 1**

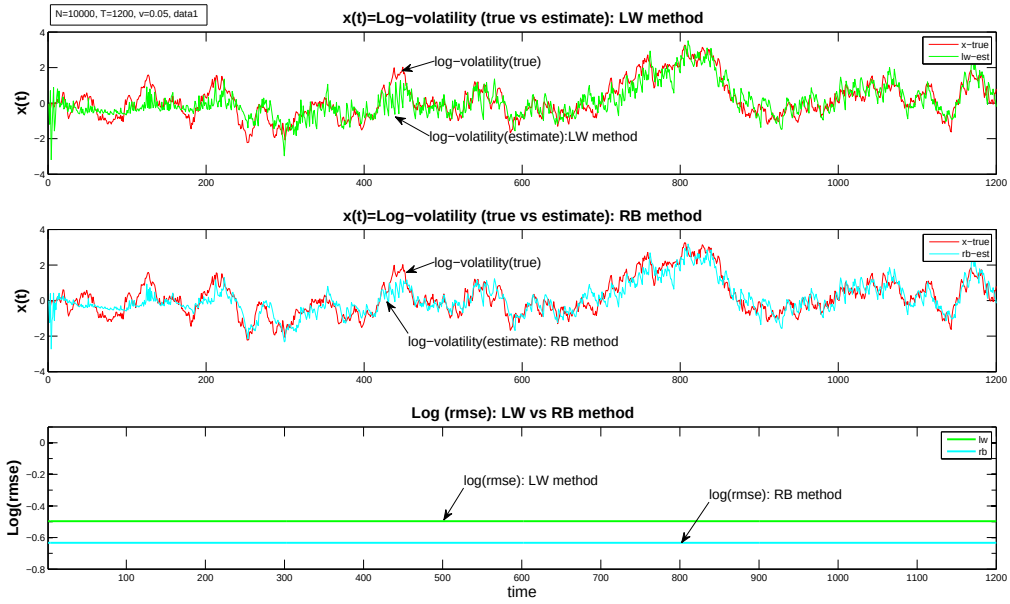

Figure 4.6: **Particle size N=10000**; estimate of log-volatility & rmse :**dataset 1**

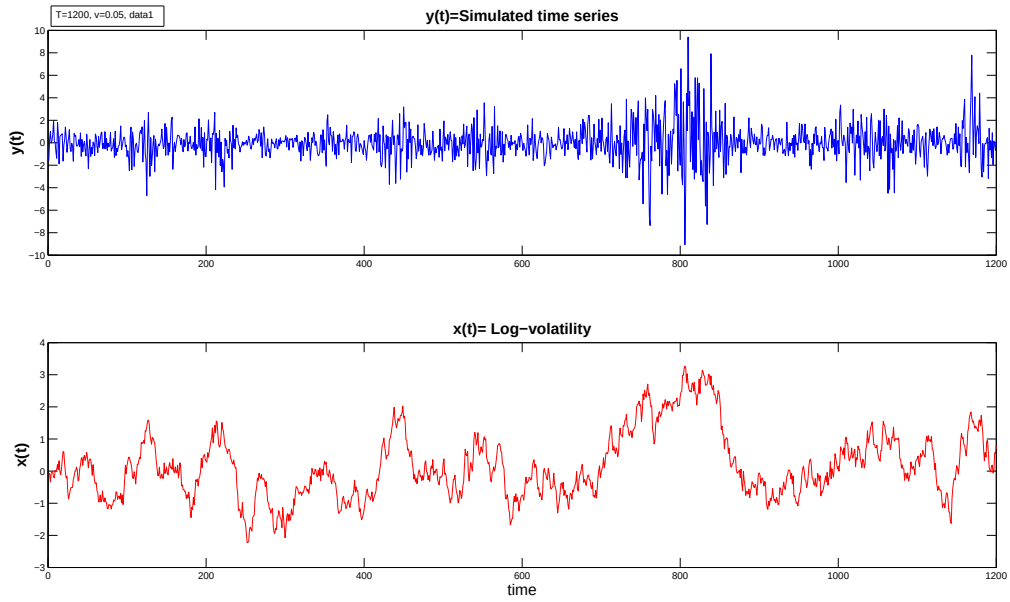

Figure 4.7: **Dataset 1**: Simulated time series y_t, and underlying log-volatility x_t, with the parameters; $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05$ and $T = 1200$.

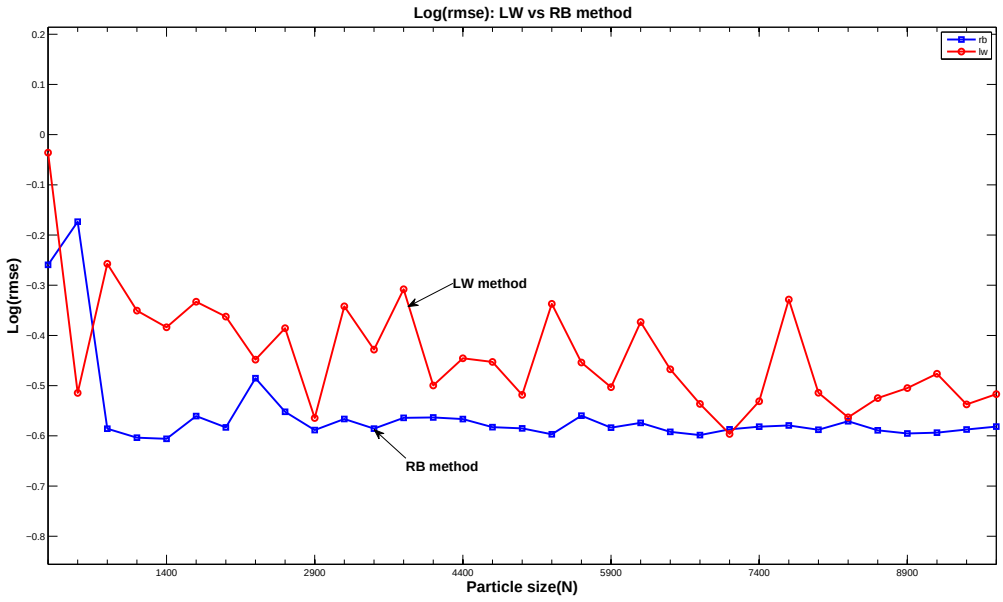

Figure 4.8: LW vs RB method: Plots comparing Log (RMSE) for estimating state (Log-volatility) with the **particle size from N=200 to 10000**: **dataset 1**

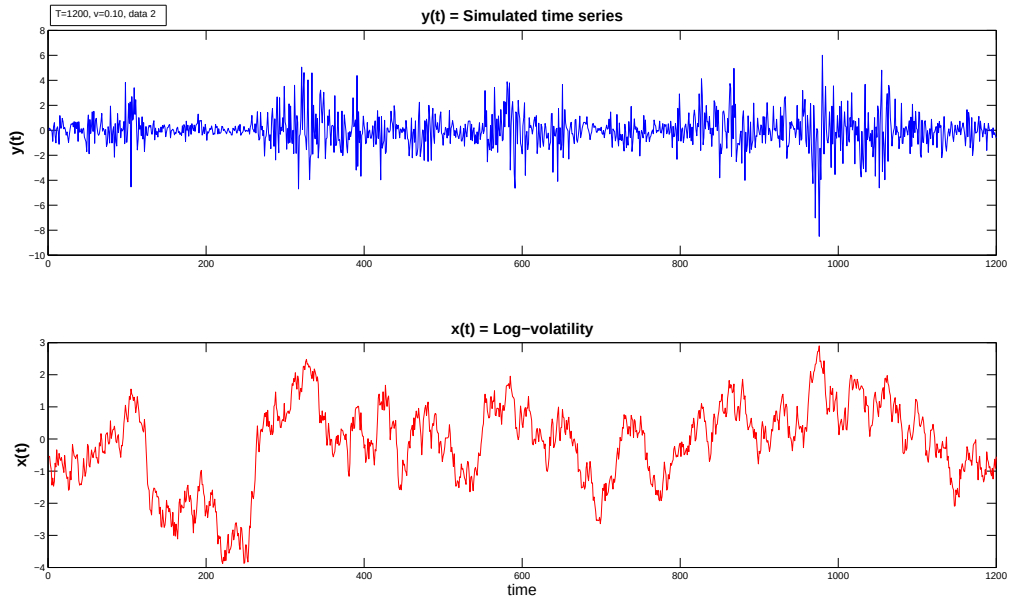

Figure 4.9: **Dataset 2**: Simulated time series y_t, and underlying log-volatility x_t, with the parameters; $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10$ and $T = 1200$

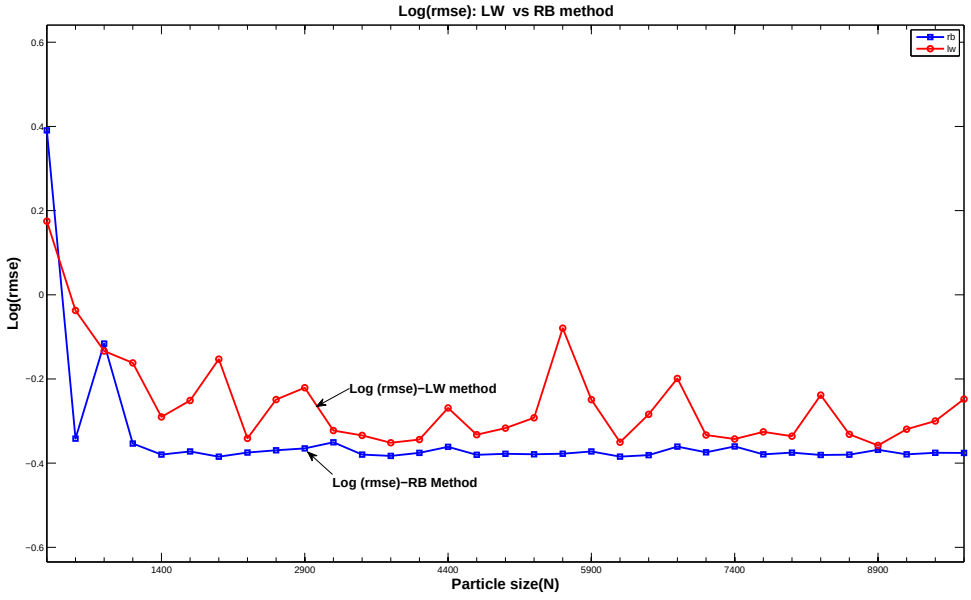

Figure 4.10: LW vs RB method : Plots comparing Log (RMSE) for estimating state (Log-volatility) with the **particle size from N=200 to 10000**: **dataset 2**

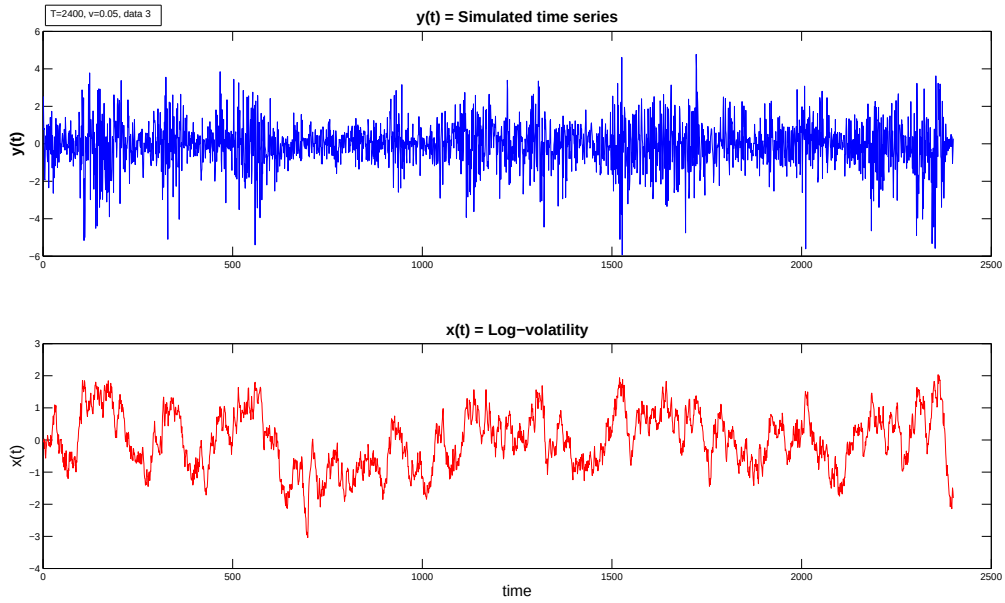

Figure 4.11: **Dataset 3** : Simulated time series y_t, and underlying log-volatility x_t, with the parameters; $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05$ and $T = 2400$

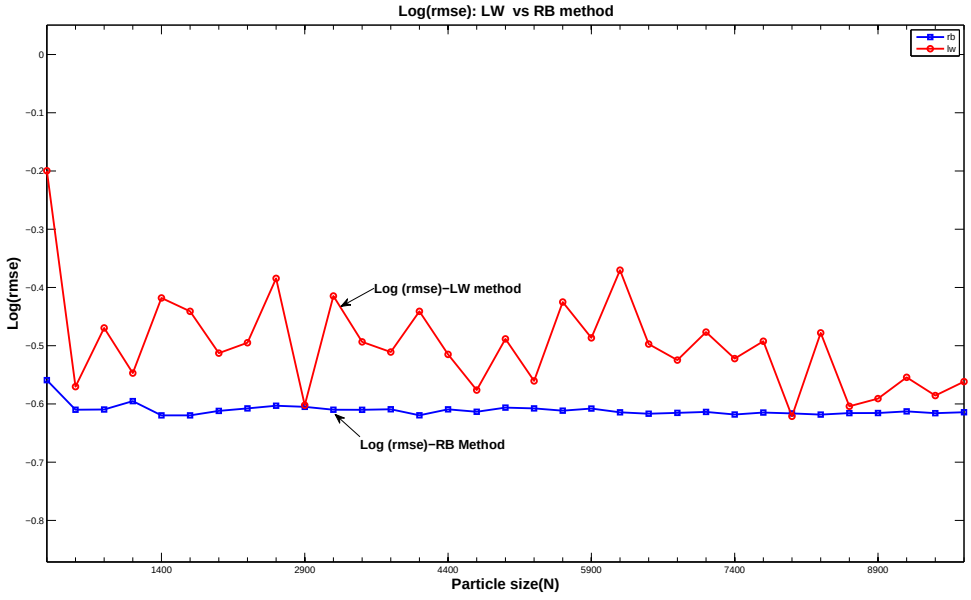

Figure 4.12: LW vs RB method : Plots comparing Log (RMSE) for estimating state (Log-volatility) with the **particle size from N=200 to 10000**: **dataset 3**

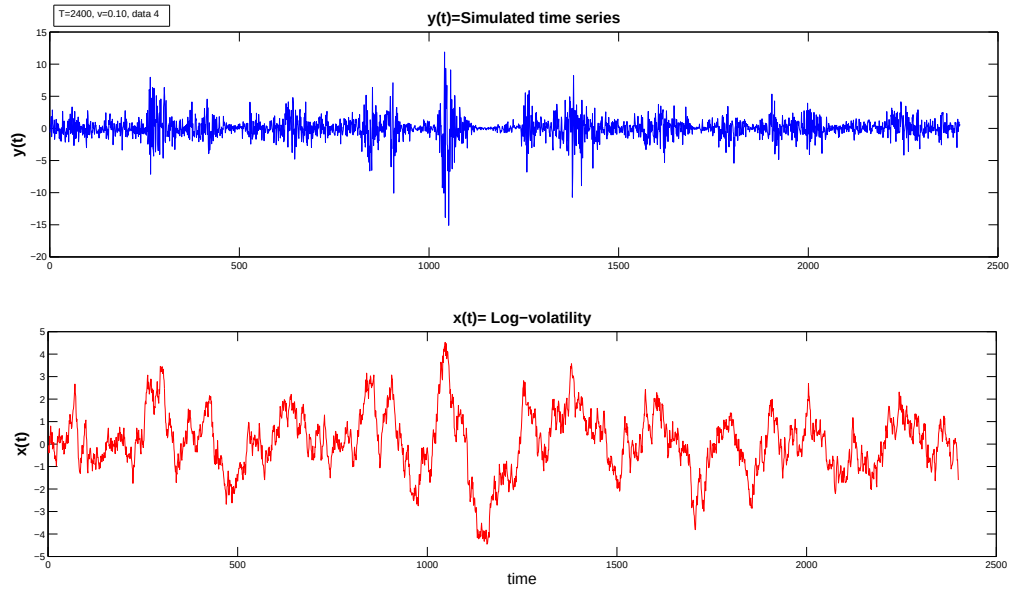

Figure 4.13: **Dataset 4** : Simulated time series y_t, and underlying log-volatility x_t, with the parameters; $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10$ and $T = 2400$

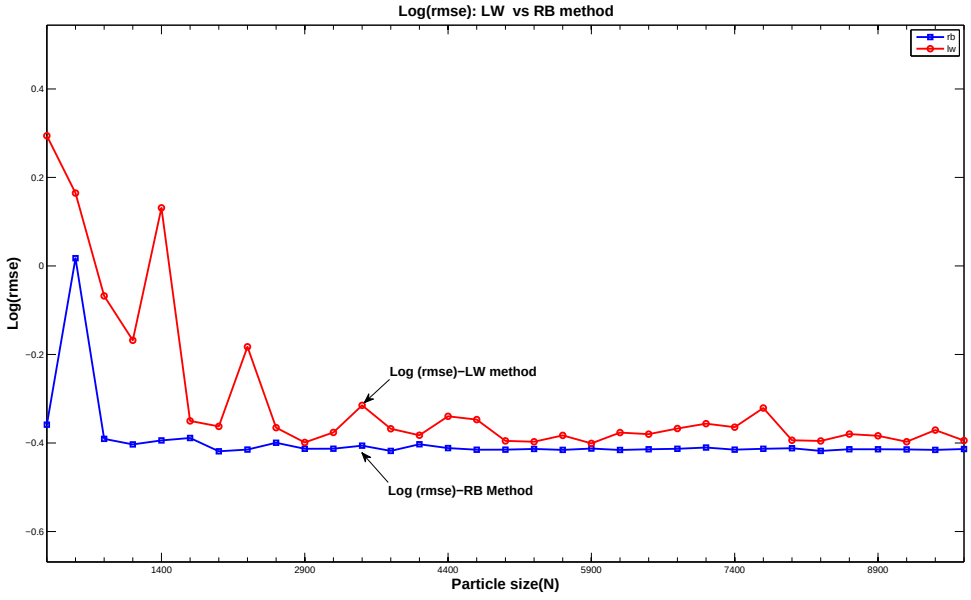

Figure 4.14: LW vs RB method : Plots comparing Log (RMSE) for estimating state (Log-volatility) with the **particle size from N=200 to 10000**: **dataset 4**

RMSE analysis on datasets with multiple realizations: LW-APF vs RB-APF

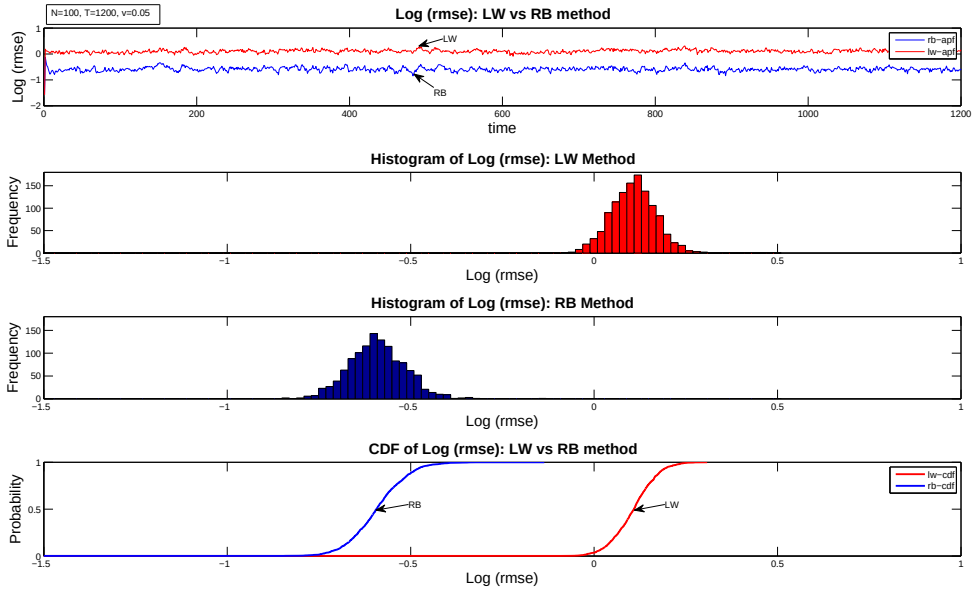

Figure 4.15: **Particle size N=100**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 1R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 1200$ and $K = 100$.

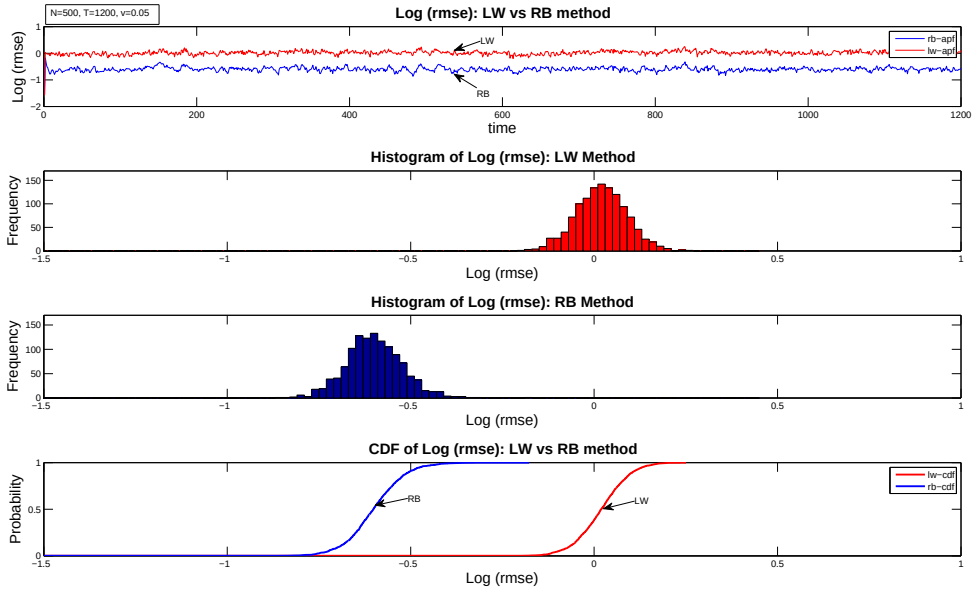

Figure 4.16: **Particle size N=500**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 1R** : $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 1200$ and $K = 100$.

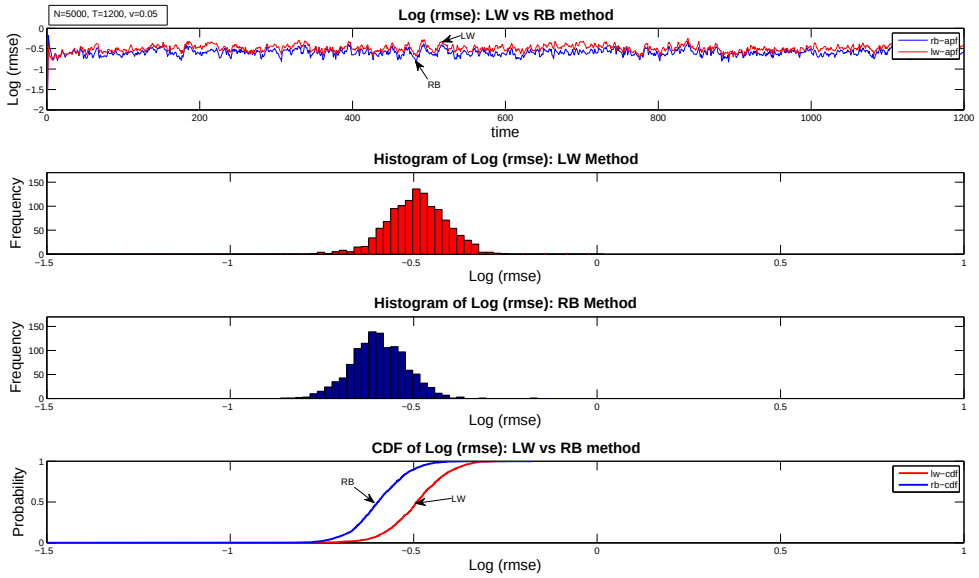

Figure 4.17: **Particle size N=5000**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 1R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 1200$ and $K = 100$.

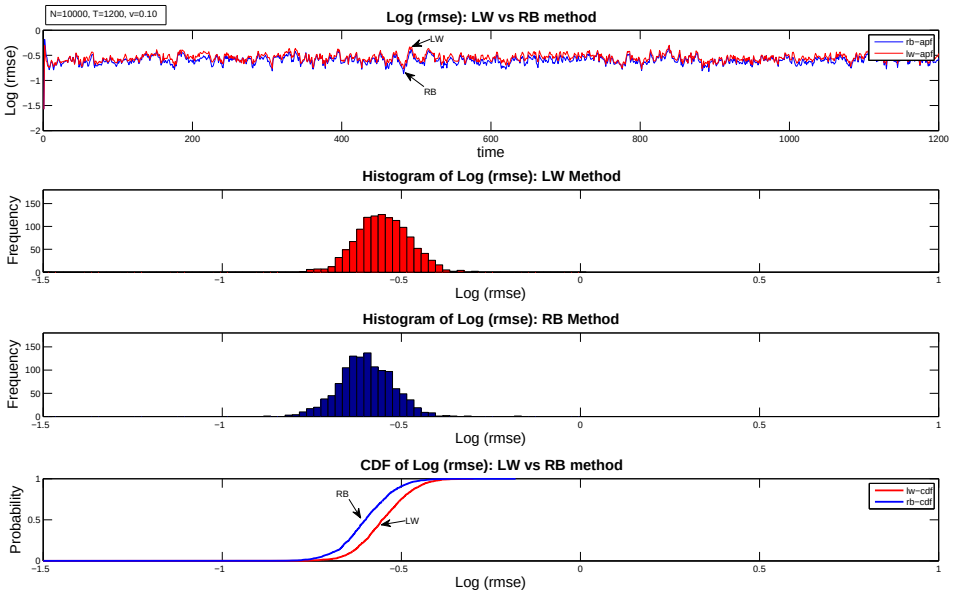

Figure 4.18: **Particle size N=10000**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 1R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 1200$ and $K = 100$.

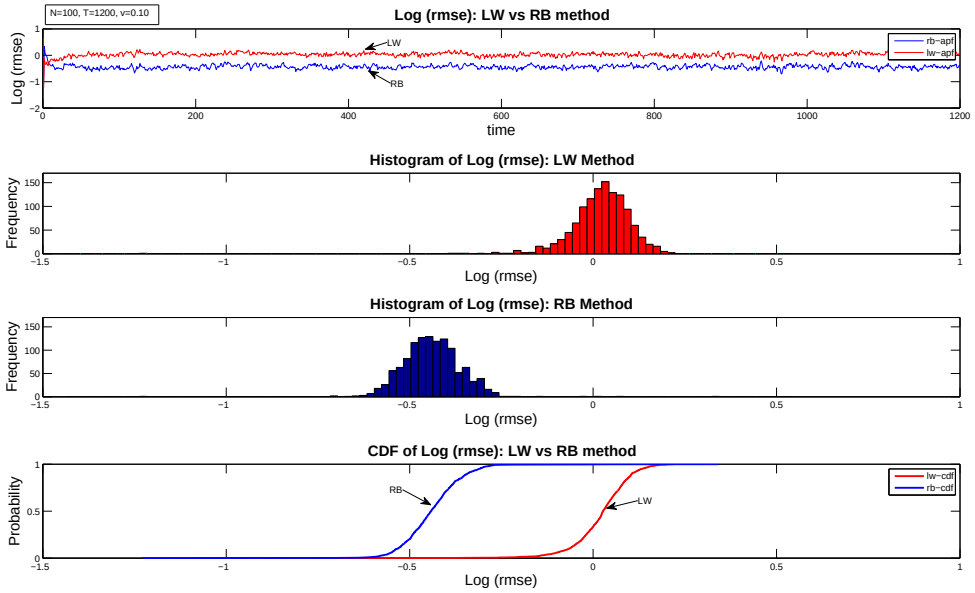

Figure 4.19: **Particle size N=100**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 2R** : $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 1200$ and $K = 100$.

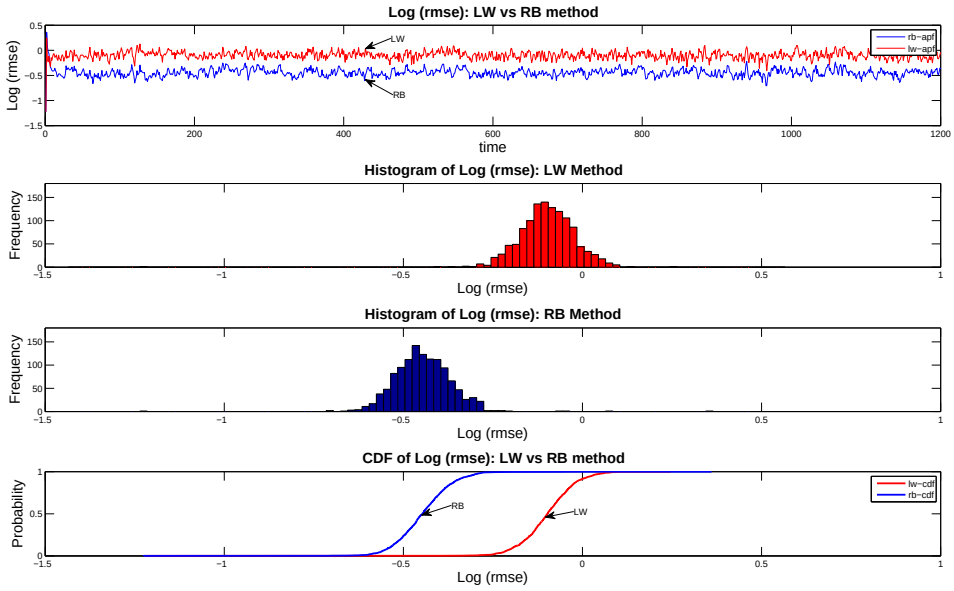

Figure 4.20: **Particle size N=500**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 2R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 1200$ and $K = 100$.

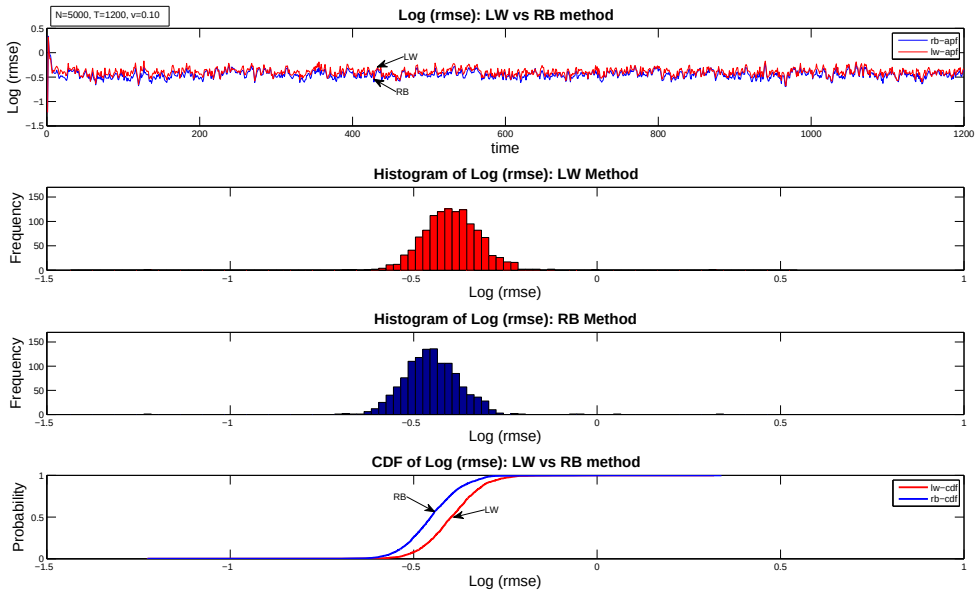

Figure 4.21: **Particle size N=5000**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 2R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 1200$ and $K = 100$.

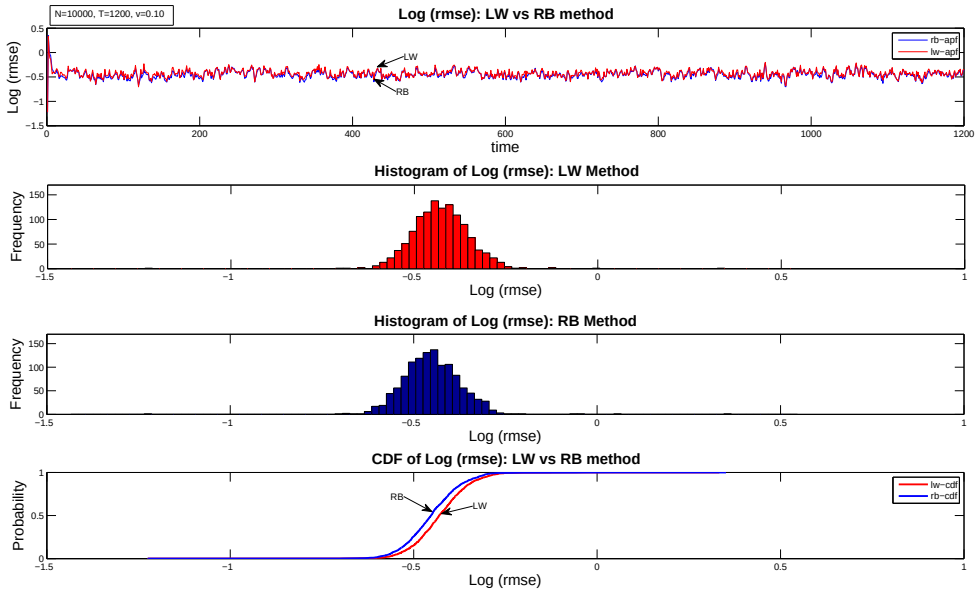

Figure 4.22: **Particle size N=10000**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 2R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 1200$ and $K = 100$.

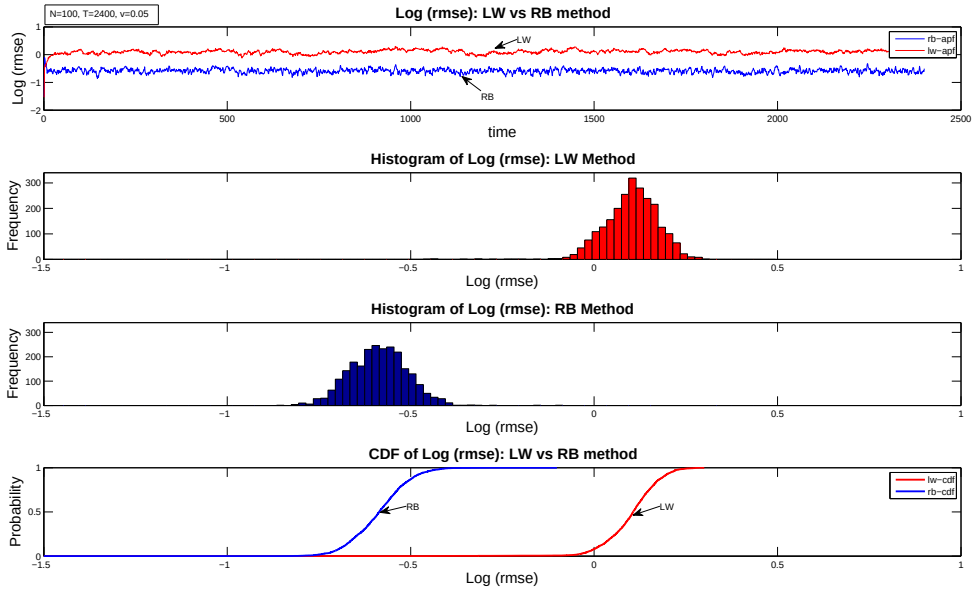

Figure 4.23: **Particle size N=100**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 3R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 2400$ and $K = 100$.

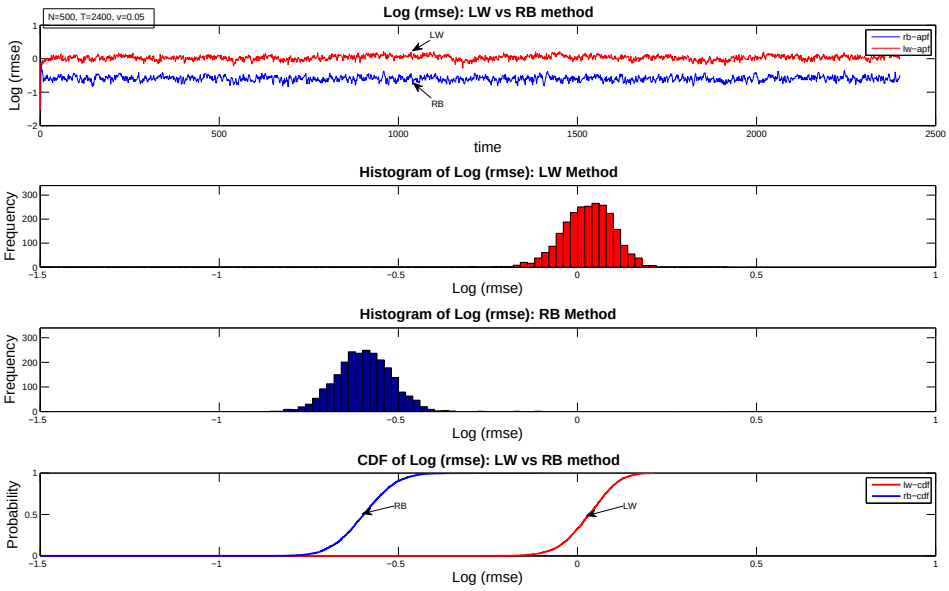

Figure 4.24: **Particle size N=500**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 3R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 2400$ and $K = 100$.

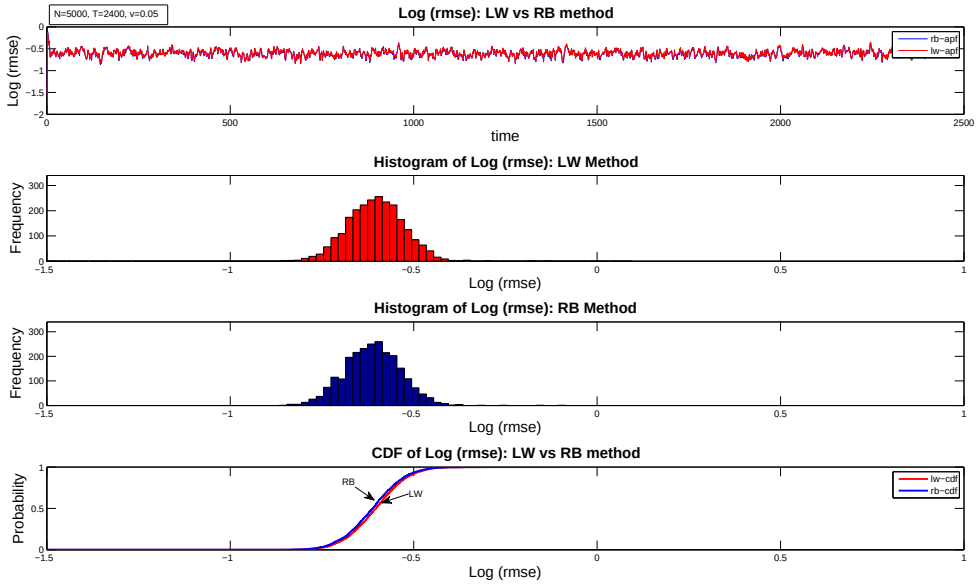

Figure 4.25: **Particle size N=5000**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 3R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 2400$ and $K = 100$.

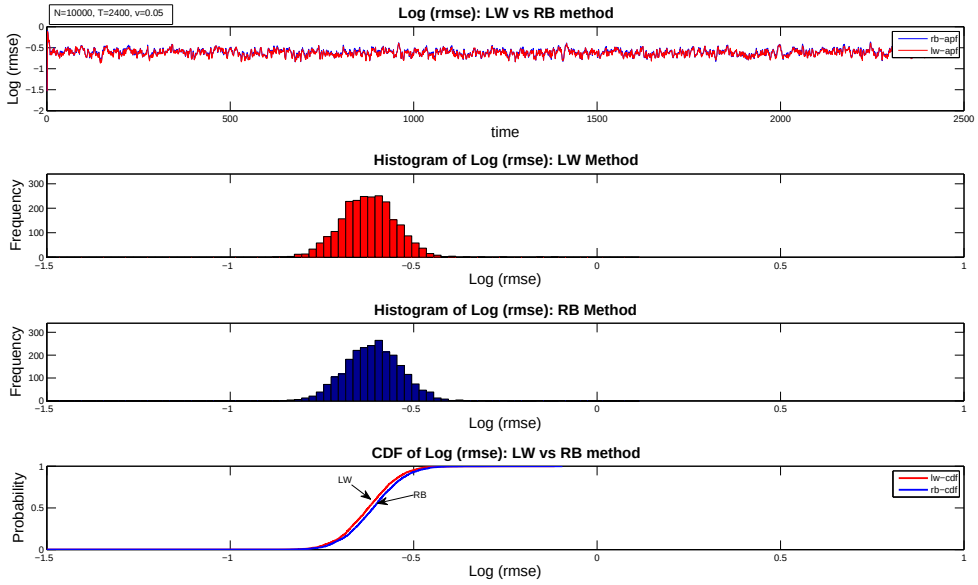

Figure 4.26: **Particle size N=10000**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 3R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.05, T = 2400$ and $K = 100$.

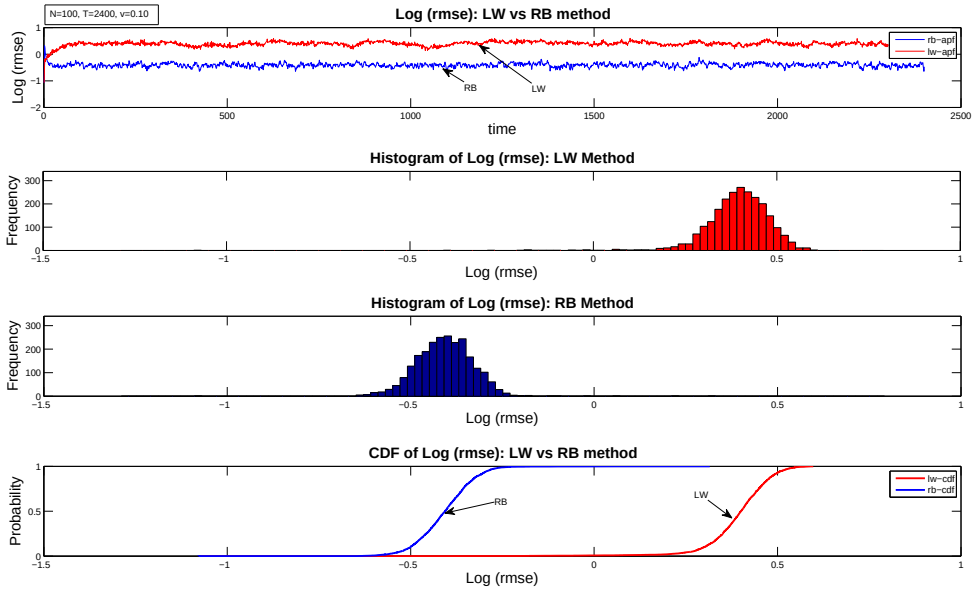

Figure 4.27: **Particle size N=100**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 4R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 2400$ and $K = 100$.

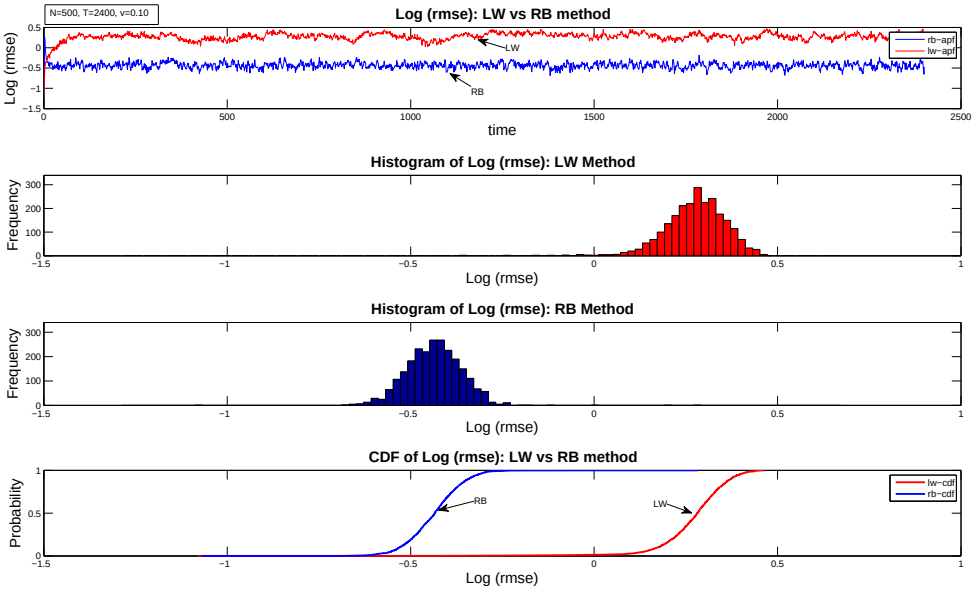

Figure 4.28: **Particle size N=500**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 4R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 2400$ and $K = 100$.

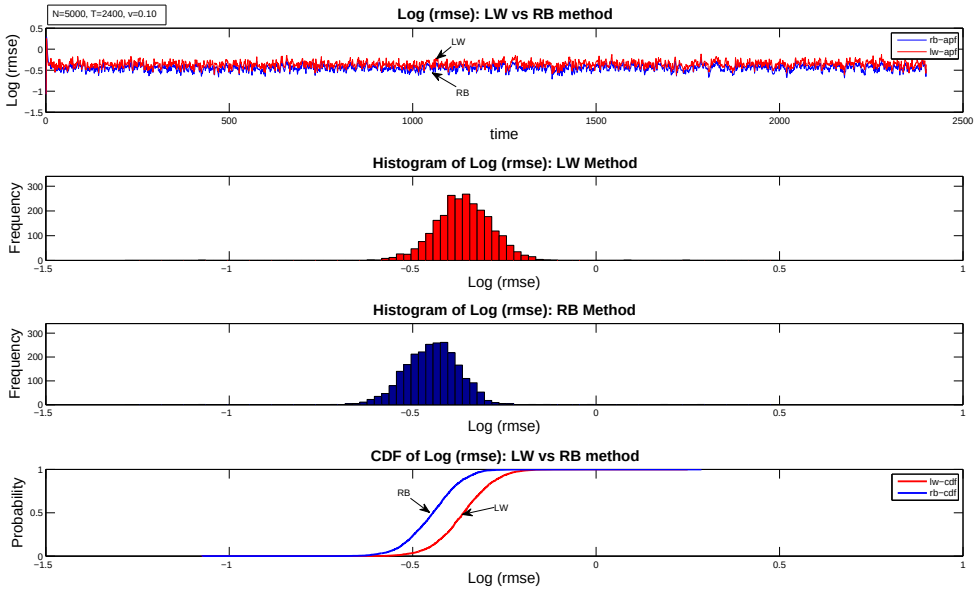

Figure 4.29: **Particle size N=5000**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 4R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 2400$ and $K = 100$.

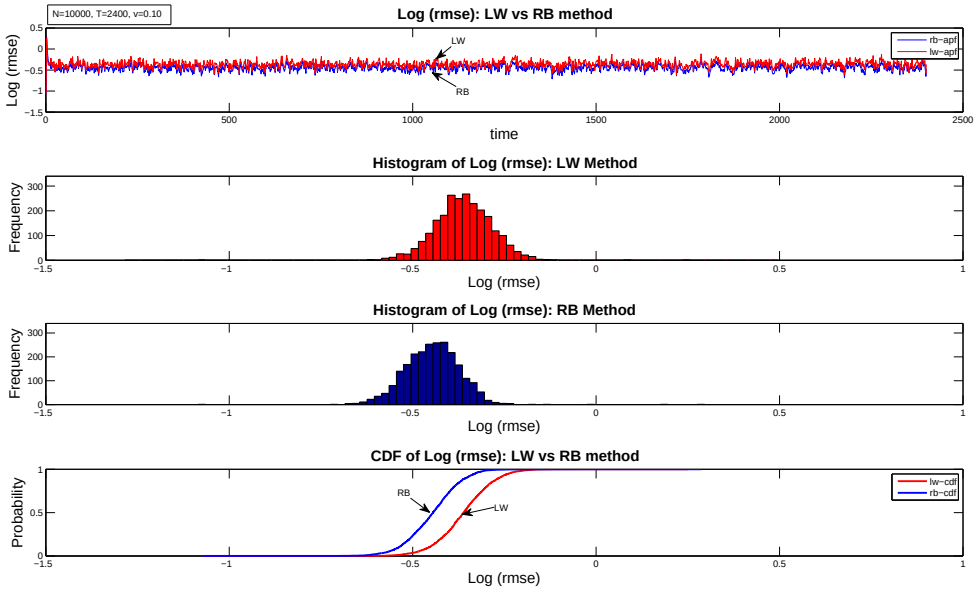

Figure 4.30: **Particle size N=10000**: RMSE analysis on log-volatility estimate, with the data set of 100 realizations. The log (rmse) and its histograms, and the CDF of log (rmse) plots are used to compare the performances of RB and LW methods. **Dataset 4R**: $\alpha = -0.005, \beta = 0.98, \sigma_u^2 = 0.10, T = 2400$ and $K = 100$.

Chapter 5

Rao-Blackwellization on Regime-Switching DSS Models

5.1 Regime-Switching DSS Models

In the previous chapters, we considered DSS models, where the states continuously evolve over time according to a predefined regime. However, empirical evidence suggests that there is a class of stochastic processes where the state process in time makes abrupt shifts, i.e, the process undergoes sudden changes. The switch in the process occurs stochastically in time, and the process moves to one of several distinct regimes. When this happens, the statistical properties of the process change. The regimes are also considered as states, except that they are discrete. When the process enters in a given regime, it stays there for some time. The transition between regimes is modeled by hidden Markov model (HMM) with a set of transition probabilities. In this chapter, we consider DSS models with changes of regimes, where the transition among the regimes follows first-order Markov process. We refer to these models as *regime-switching state space models* (Quandt, 1972; Goldfeld and Quandt, 1973; Hamilton, 1989; Chib, 1996; Kim and Nelson, 1999).

In these models, the continuous state process depends on the discrete regimes. It has been observed economics and finance, biology and engineering systems show this type of behavior. The volatility in the financial markets undergoes abrupt changes, such as, from low-volatility to high-volatility and back to low-volatility. The changes from positive to negative growth rate of the real GNP suggests a recurrent economic expansion and contraction (recession). Hence, state space models with multiple regimes provides more flexibility and better accuracy, though it increases model complexity due to an additional hidden layer.

5.2 Application to SV Models

We want to extend our work from the standard DSS model to regime-switching DSS models. As an application, we choose SV models that have more than one regime. In the context of SV models, the regimes are defined by different parameters in the state equation, which contribute to different levels of volatility. More specifically, we consider SV models with two distinct regimes, one with low-volatility and another with high-volatility of the return series. This is reflected due to changes of the levels of the log-volatility (state).

The transition between the regimes follows a first order time-homogeneous Markov process. The literature suggests that volatility undergoes occasional abrupt changes from one discrete regime to another, and hence, it is appropriate to model the volatility with distinct regimes. The literature on conditional heteroscedasticity and SV models suggests that there is a substantial persistence (Chou, 1988) in the high frequency data in finance, i.e., clustering in the volatility process. The persistence in variance which evolves over time, refers to the characteristics of momentum in the conditional variance, i.e., the past volatility explains the current volatility. Lamoureux and Lastrapes (Lamoureux and Lastrapes, 1990) have conducted empirical study on persistence of volatility on stock return data using GARCH (Generalized Autoregressive Conditional Heteroscedasticity) models. Their findings reveal that the persistence could be overstated if the structural changes/shifts in the volatility process is not considered in the model. The degree to which the persistence is present in the daily stock return data is an important economic research issue. Poterba and Summers (Poterba and Summers, 1986) argue that the movements on the stock prices reflect substantially the time-varying risk premia induced by changes in the volatility. Hamilton and Susmel (Hamilton and Susmel, 1994) proposed an ARCH model with regime switching for volatility level, and So (So et al., 1998) introduced a methodology for regime switching SV models.

As a result of increased complexity due to having another hidden layer, the model provides improved modeling of the data. Estimation of the unknowns of this model remains a challenging problem since it has two hidden (unknown) layers. Hamilton and Susmel used the ML method to estimate the parameters and the states, whereas So and Li used the MCMC sampling algorithm where they sample from the exact posterior distribution of the parameters and the states. These two methods work with batch data sets only, and therefore they are not suitable for realtime or online inference. Carvalho and Lopes used the APF method (Carvalho and Lopes, 2006; Pitt and Shepard, 1999) based on Liu and West's (Liu and West, 2001) algorithm to estimate the unknowns of the *regime-switching SV* (RSSV) model. The algorithm performs sequential realtime estimation of the log-volatility and the parameters of the regimes. We want to implement RB on the PF algorithm along the lines described previously. As before, our algorithm avoids sampling from the parameter space, which is now even larger due to the presence of two regimes.

5.2.1 The RSSV Models

The RSSV is a classical HMM with discrete state space for the regimes, and where the transition between the states takes place according to transition probabilities. For our example, we consider two discrete regimes or states, which we denote by $s_t \in (1, 2)$. The distribution of the log-volatility (x_t) for a given regime differs only in their levels (mean). The RSSV model is described as,

$$s_t|s_{t-1} \sim Markov(\mathbf{P}, \lambda) \qquad (switching \quad state \quad process) \tag{5.1}$$

$$x_t = \begin{cases} \alpha_1 + \beta x_{t-1} + \sigma_u u_t, & s_t = 1 \\ \alpha_2 + \beta x_{t-1} + \sigma_u u_t, & s_t = 2 \end{cases} \qquad (state \quad process) \tag{5.2}$$

$$y_t = e^{x_t/2} v_t \qquad (observation \quad process) \tag{5.3}$$

$$p(s_1 = j) = \lambda_j, \quad j \in (1, 2) \tag{5.4}$$

$$p(s_t = j|s_{t-1} = k) = p_{j,k}, \quad j, k \in (1, 2) \tag{5.5}$$

where y_t is the observation at time t, x_t is the log-volatility, s_t is the switching regime governed by transition probability matrix (TPM) $\mathbf{P}$, and u_t and v_t are uncorrelated Gaussian noise processes with zero mean and unit variances. The constant level parameters $\alpha = (\alpha_1,\ \alpha_2)$ correspond to the levels of the log-volatility of the regimes. The objective of the model is to discriminate clusters with high volatility from low volatility, which are captured by the parameters α_j. Hence, α_1 belongs to *regime 1*, corresponds to *low volatility*, and α_2 belongs to *regime 2* , corresponds to *high volatility*. The unknown states (x_t, s_t) are the main interest, whereas the unknown static parameters $\boldsymbol{\theta} = (\alpha_1,\ \alpha_2,\ \beta,\ \sigma_u^2,\ p_{11},\ p_{22})^\top$ are considered as nuisance. We adopt reparameterization for α_{s_t} for the two distinct regimes as,

$$\alpha_{s_t} = \gamma_1 + \gamma_2 I_{2t} \tag{5.6}$$

where $I_{2t} = 1$ if $s_t = 2$ and is zero otherwise. We also have $\gamma_1 \in R$ and $\gamma_2 > 0$. Therefore, $\alpha_1 = \gamma_1$ and $\alpha_2 = \gamma_1 + \gamma_2$.

The following are the **four datasets** generated according to RSSV models and used in our simulation study, where each dataset has data length of $T = 1200$ points. The parameters used in the simulation are from the literatures by So et al. (1998); Carvalho and Lopes (2006). We have applied LW-APF vs RB-APF methods on these four datasets with the various particle sizes (N=100-3000), where the transition probability matrix $\mathbf{P}$ is assumed known. However, we have presented the results only for dataset 1RS.

Dataset 1RS: $\alpha_1 = -2.5, \alpha_2 = -1, \beta = 0.5, \sigma_u^2 = 0.05, p_{11} = 0.990, p_{22} = 0.985,$

Dataset 2RS : $\alpha_1 = -2.5, \alpha_2 = -1, \beta = 0.5, \sigma_u^2 = 0.05, p_{11} = 0.990, p_{22} = 0.985,$

Dataset 3RS: $\alpha_1 = -2.5, \alpha_2 = -1, \beta = 0.5, \sigma_u^2 = 0.10, p_{11} = 0.990, p_{22} = 0.985,$

Dataset 4RS : $\alpha_1 = -2.5, \alpha_2 = -1, \beta = 0.5, \sigma_u^2 = 0.10, p_{11} = 0.990, p_{22} = 0.985.$

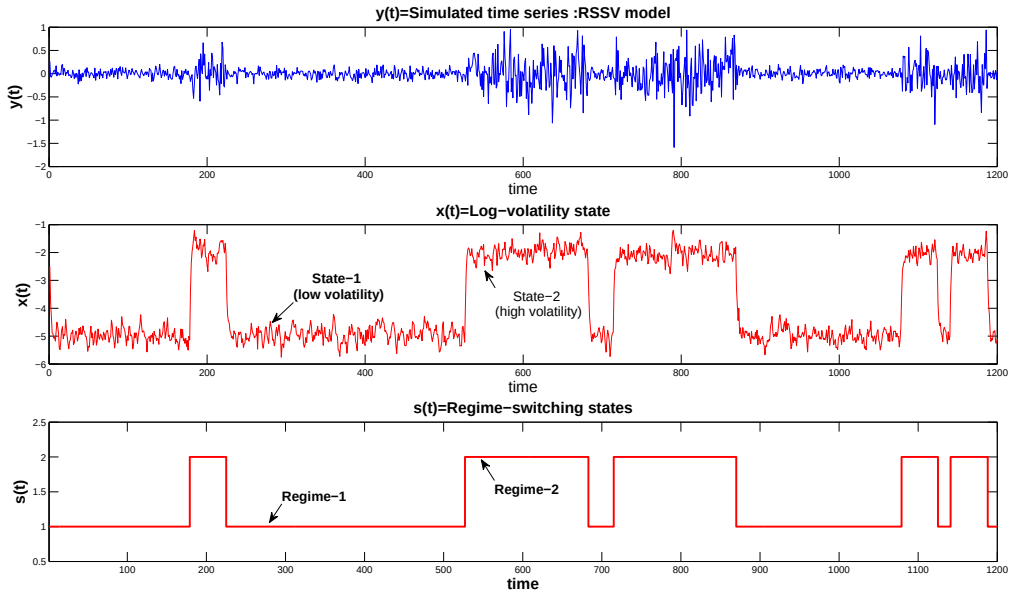

Figure 5.1: **Dataset 1RS (RSSV model)**: Simulated time series y_t, underlying volatility state x_t (log-volatility), and regime-switching states s_t, where the parameters are; $\alpha_1 = -2.5, \alpha_2 = -1, \beta = 0.5, \sigma_u^2 = 0.05, p_{11} = 0.990, p_{22} = 0.985$ and $T = 1200$.

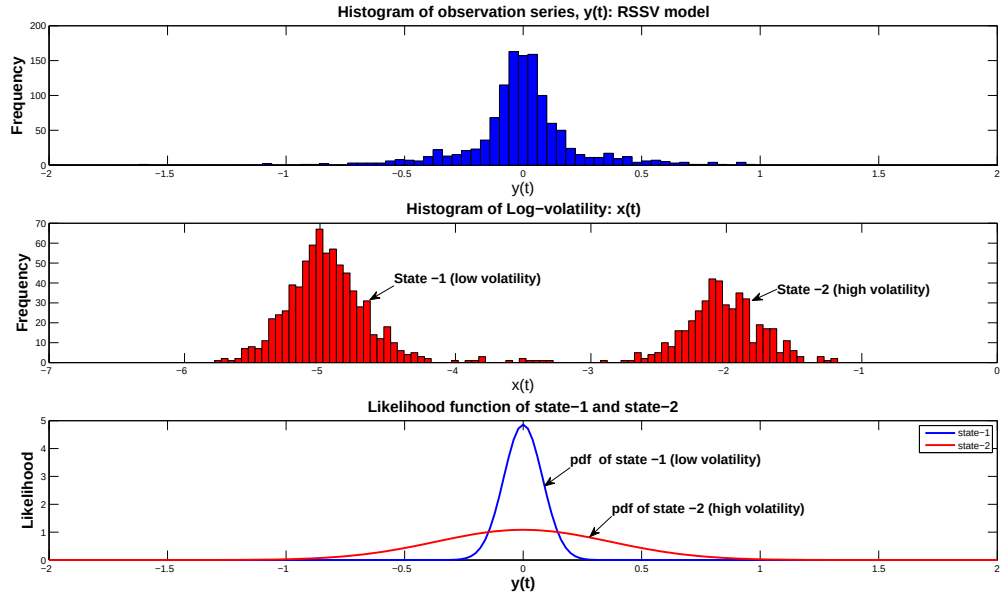

Figure 5.2: **Dataset 1RS (RSSV model)**: Histograms of time series y_t and log-volatility state x_t , and the likelihood functions of regime 1 (low volatility) and regime 2 (high volatility).

5.3 PF applied to RSSV Models: LW Method

We apply the method of Liu and West for sequential estimation of $(x_t, s_t, \boldsymbol{\theta})$ of RSSV models with two regimes. The static parameters of the model $\boldsymbol{\theta} = (\gamma_1,\ \gamma_2,\ \beta,\ \sigma_u^2,\ p_{11},\ p_{22})^\top$. The sequential inference of states and the parameters proceeds as before. The *random measure* at time instant $t-1$ is given by

$$\{\boldsymbol{z}_{t-1}^i, \boldsymbol{\theta}^i, w_{t-1}^i\}_{i=1}^N \approx p(\boldsymbol{z}_{t-1}, \boldsymbol{\theta}|y_{1:t-1}) \tag{5.7}$$

where $\boldsymbol{z}_t = (x_t, s_t)$.

At time $t-1$, the predicted (prior) distribution is defined as,

$$\begin{aligned} \textit{Prediction}: \quad p(\boldsymbol{z}_t, \boldsymbol{\theta}|\boldsymbol{y}_{1:t-1}) &= \int p(\boldsymbol{z}_t, \boldsymbol{z}_{t-1}, \boldsymbol{\theta}|y_{1:t-1})\mathrm{d}\boldsymbol{z}_{t-1} \\ &= \int p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{\theta}) p(\boldsymbol{z}_{t-1}, \boldsymbol{\theta}|y_{1:t-1})\mathrm{d}\boldsymbol{z}_{t-1} \end{aligned} \tag{5.8}$$

At time t, when the new observation y_t becomes available, the prior distribution is modified via Bayes rule to the the desired posterior distribution,

$$\begin{aligned} \textit{Update}: \quad p(\boldsymbol{z}_t, \boldsymbol{\theta}|y_{1:t}) &\propto \underbrace{p(y_t|\boldsymbol{z}_t, \boldsymbol{\theta})}_{likelihood} \underbrace{p(\boldsymbol{z}_t, \boldsymbol{\theta}|y_{1:t-1})}_{prior} \\ &\propto p(y_t|\boldsymbol{z}_t, \boldsymbol{\theta}) p(\boldsymbol{z}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1}) p(\boldsymbol{\theta}|y_{1:t-1}) \end{aligned} \tag{5.9}$$

Our objective is obtaining *random measure* $\{\boldsymbol{z}_t^i, \boldsymbol{\theta}^i, w_t^i\}_{i=1}^N$ to approximate $p(\boldsymbol{z}_t, \boldsymbol{\theta}|y_{1:t})$, i.e.,

$$\{\boldsymbol{z}_t^i, \boldsymbol{\theta}^i, w_t^i\}_{i=1}^N \approx p(\boldsymbol{z}_t, \boldsymbol{\theta}|y_{1:t}) \tag{5.10}$$

We implement the APF algorithm on the RSSV model in accordance with Carvalho and Lopes (Carvalho and Lopes, 2006). The sequential inference of $\boldsymbol{z}_t$ and the parameter vector $\boldsymbol{\theta}$ is performed based on the particles of the states, and, the parameter vector and their weights as described previously. We assume that the parameter vector $\boldsymbol{\theta} = (\gamma_1,\ \gamma_2,\ \beta,\ \sigma_u^2,\ p_{11},\ p_{22})^\top$ follows mixture of multivariate normals (3.1.2), i.e.,

$$p(\boldsymbol{\theta}|y_{1:t}) \approx \sum_{i=1}^{N} w_t^i N(\boldsymbol{\theta}|\mathbf{m}_t^i, h^2\mathbf{V}_t) \tag{5.11}$$

We transform some of the components of $\boldsymbol{\theta}$ so that they are defined on the real line, $\mathbb{R}$. Therefore, with the transformation, $\widetilde{\boldsymbol{\theta}} = \{\gamma_1,\ log(\gamma_2),\ \beta,\ log(\sigma_u^2),\ log(\frac{p_{11}}{1-p_{11}}),\ log(\frac{p_{22}}{1-p_{22}})\}^\top$.

The following is a step-by-step procedures of the algorithm. Again, the notation $\widetilde{\boldsymbol{\theta}}_t$ indicates the posterior at time instant t, $\widetilde{\boldsymbol{\theta}}_t$ is not time-varying.

- Suppose, at time $t-1$, we have weighted samples from the posterior PDF, i.e., $\{x_{t-1}^i, s_{t-1}^i, \widetilde{\boldsymbol{\theta}}_{t-1}^i, w_{t-1}^i\}_{i=1}^N \approx p(x_{t-1}, s_{t-1}, \widetilde{\boldsymbol{\theta}}_{t-1}|y_{t-1})$

1. The prior estimates of $\{x_{t-1}^i, s_{t-1}^i, \widetilde{\boldsymbol{\theta}}_{t-1}^i\}_{i=1}^N$ are given by $\{\hat{\mu}_t^i, \hat{s}_t, \mathbf{m}_{t-1}^i\}$, where
   - $\hat{s_t}^i = \text{argmax}\ p(s_t = j|s_{t-1} = s_{t-1}^i, \widetilde{\boldsymbol{\theta}}^i); j \in (1,2)$
   - $\hat{\mu}_t^i = \alpha_{\hat{s_t^i}}^i + \beta^i x_{t-1}^i$
   - $\mathbf{m}_{t-1}^i = c\widetilde{\boldsymbol{\theta}}_{t-1}^i + (1-c)\bar{\boldsymbol{\theta}}_{t-1}$
   - $\bar{\boldsymbol{\theta}}_{t-1} = \sum_{i=1}^N w_{t-1}^i \widetilde{\boldsymbol{\theta}}_{t-1}^i$
   - $\mathbf{V}_{t-1} = \sum_{i=1}^N w_{t-1}^i (\widetilde{\boldsymbol{\theta}}_{t-1}^i - \bar{\boldsymbol{\theta}}_{t-1})(\widetilde{\boldsymbol{\theta}}_{t-1}^i - \bar{\boldsymbol{\theta}}_{t-1})^T$
2. Compute the first-stage weights: $L_t^i \propto w_{t-1}^i p(y_t|\hat{\mu}_t^i, \mathbf{m}_{t-1}^i), \quad i = 1, \ldots, N$
3. Normalize the weights: $\widetilde{L}_t^i = \frac{L_t^i}{\sum_{j=1}^N L_t^j}, \quad i = 1, \ldots, N$
4. Resampling: sample the indexes $J_i \in (1, \ldots, N)$ with $\widetilde{L}_t$, J_i are auxiliary variables.
5. Propagate the parameter vector: $\widetilde{\boldsymbol{\theta}}_t^i \sim N(\widetilde{\boldsymbol{\theta}}_t|\mathbf{m}_{t-1}^{J_i}, h^2\mathbf{V}_{t-1}), \quad i = 1, \ldots, N$
6. Propagate the regime-switching states: $s_t^i \sim p(s_t|s_{t-1}^{J_i}, \widetilde{\boldsymbol{\theta}}^i)$
7. Propagate the volatility state: $x_t^i \sim p(x_t|x_{t-1}^{J_i}, s_t^i, \widetilde{\boldsymbol{\theta}}_t^i), \quad i = 1, \ldots, N$
8. Evaluate the 2nd-stage weights: $w_t^i \propto \frac{p(y_t|x_t^i, \boldsymbol{\theta}_t^i)}{p(y_t|\hat{\mu}_t^{J_i}, \mathbf{m}_t^{J_i})}, \quad i = 1, \ldots, N$
9. Normalize the weights: $\widetilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}, \quad i = 1, \ldots, N$
10. The posterior at time t is represented by $\{x_t^i, s_t^i, \widetilde{\boldsymbol{\theta}}_t^i, \widetilde{w}_t^i\}_{i=1}^N \approx p(x_t, s_t, \widetilde{\boldsymbol{\theta}}_t|y_{1:t})$.

5.4 PF applied to RSSV Models: RB-based Method

In this section, we show how to implement the RB on the RSSV model. For convenience, we reintroduce the model here again,

$$
\begin{aligned}
s_t|s_{t-1} &\sim Markov(\mathbf{P}, \lambda) \quad (switching \quad state \quad process) && (5.12)\\
x_t &= \begin{cases} \alpha_1 + \beta x_{t-1} + \sigma_u u_t, & s_t = 1 \\ \alpha_2 + \beta x_{t-1} + \sigma_u u_t, & s_t = 2 \end{cases} \quad (state \quad process) && (5.13)\\
y_t &= e^{x_t/2} v_t \qquad (observation \quad process). && (5.14)
\end{aligned}
$$

For the distributions of states and observations with the parameter vector $\boldsymbol{\psi} = (\alpha_1, \alpha_2, \beta, \sigma_u^2)$,

$$
\begin{aligned}
p(s_t = j|s_{t-1} = i) &= p_{i,j}, \quad i, j \in (1, 2) \quad (switching \quad state \quad process) && (5.15)\\
p(x_t|x_{0:t-1}, s_t, \boldsymbol{\psi}) &\sim \begin{cases} \mathcal{N}(\alpha_1 + \beta x_{t-1}, \sigma_u^2), & s_t = 1 \\ \mathcal{N}(\alpha_2 + \beta x_{t-1}, \sigma_u^2), & s_t = 2 \end{cases} \quad (state \quad process) && (5.16)\\
p(y_t|x_t) &\sim \mathcal{N}(0, e^{x_t}) \qquad (observation \quad process). && (5.17)
\end{aligned}
$$

We can apply RB on the parameter space $\boldsymbol{\psi}$ with the *implied integration* method. RB on the state equation amounts to integrating out the nuisance parameter vector $\boldsymbol{\psi}$, i.e.,

$$
\begin{aligned}
p(x_t|x_{0:t-1}, s_{1:t}) &= \int_{\boldsymbol{\psi}} p(x_t, \boldsymbol{\psi}|x_{0:t-1}, s_{1:t}) \mathrm{d}\boldsymbol{\psi} \\
&= \int_{\boldsymbol{\psi}} p(x_t|x_{0:t-1}, \boldsymbol{\psi}, s_{1:t}) p(\boldsymbol{\psi}|x_{0:t-1}, s_{1:t}) \mathrm{d}\boldsymbol{\psi} && (5.18)
\end{aligned}
$$

We perform the above integration using the implied method instead of direct integration as discussed previously. We write as,

$$
\begin{aligned}
p(\boldsymbol{\psi}|x_{0:t}, s_{1:t}) &= \frac{p(\boldsymbol{\psi}, x_{0:t}, s_{1:t})}{p(x_{0:t}, s_{1:t})} \\
&= \frac{p(x_t|x_{0:t-1}, s_{1:t}, \boldsymbol{\psi},) p(x_{0:t-1}|s_{1:t}, \boldsymbol{\psi}) p(s_{1:t}|\boldsymbol{\psi}) p(\boldsymbol{\psi})}{p(x_t|x_{0:t-1}, s_{1:t}) p(x_{0:t-1}|s_{1:t}) p(s_{1:t})} \\
&= \frac{p(x_t|x_{0:t-1}, s_t, \boldsymbol{\psi},) p(x_{0:t-1}|s_{1:t-1}, \boldsymbol{\psi}) p(s_{1:t}|\boldsymbol{\psi}) p(\boldsymbol{\psi})}{p(x_t|x_{0:t-1}, s_t) p(x_{0:t-1}|s_{1:t-1}) p(s_{1:t})} \\
&= \frac{p(x_t|x_{0:t-1}, s_t, \boldsymbol{\psi}) p(\boldsymbol{\psi}|x_{0:t-1}, s_{1:t-1})}{p(x_t|x_{0:t-1}, s_t)}.
\end{aligned}
$$

Hence,

$$p(x_t|x_{0:t-1}, s_t) = \frac{\overbrace{p(x_t|x_{0:t-1}, s_t, \boldsymbol{\psi},)}\overbrace{p(\boldsymbol{\psi}|x_{0:t-1}, s_{1:t-1})}}{\underbrace{p(\boldsymbol{\psi}|x_{0:t}, s_{1:t})}} \tag{5.19}$$

A more detailed description of the expression in (5.19) as follows:

$$\begin{aligned} p(x_t|x_{0:t-1}, s_t) &\sim \begin{cases} t_{\nu_1}(x_t|m_{1,t-1}, r_{t-1}), & \text{when} \quad s_t = 1 \\ t_{\nu_2}(x_t|m_{2,t-1}, r_{t-1}), & \text{when} \quad s_t = 2 \end{cases} \\ Where, \quad m_{1,t-1} &= \widehat{\alpha}_1 + \widehat{\beta} x_{t-1}, \quad (mean) \\ m_{2,t-1} &= \widehat{\alpha}_2 + \widehat{\beta} x_{t-1}, \quad (mean) \\ r_{t-1} &= \frac{\frac{2}{b} + RSS_{t-1}}{2a + t - 1}, \quad (variance) \\ \nu_1 &= M - 1, \\ \nu_2 &= t - M - 1, \end{aligned} \tag{5.20}$$

where $\widehat{\alpha}_1$, $\widehat{\alpha}_2$ and $\widehat{\beta}$ are estimated with the sequential least squares (SLS) method from $x_{0:t-1}$. The variance r_{t-1} is also estimated from $x_{0:t-1}$, where RSS_{t-1} is the Residual Sum Squares of the linear state model. The symbol M is the number of data points that belong to regime-1, and ν_1 and ν_2 are the degrees of freedom of the $t-pdf$ of the respective regimes.

After RB on the state, the filtering posterior distribution is given by,

$$\begin{aligned} p(x_t, s_t, \mathbf{P}|x_{0:t-1}, s_{1:t-1}, y_{1:t}) &= \int_{\boldsymbol{\psi}} p(x_t, s_t, \mathbf{P}, \boldsymbol{\psi}|x_{0:t-1}, s_{1:t-1}, y_{1:t}) \mathrm{d}\boldsymbol{\psi} \\ &= \int_{\boldsymbol{\psi}} p(x_t, s_t, \mathbf{P}|x_{0:t-1}, s_{1:t-1}, y_{1:t}, \boldsymbol{\psi}) p(\boldsymbol{\psi}|x_{0:t-1}, s_{1:t-1}, y_{1:t}) \mathrm{d}\boldsymbol{\psi} \\ &\propto p(y_t|x_t) p(x_t|x_{0:t-1}, s_t) p(s_t|s_{t-1}, \mathbf{P}) p(\mathbf{P}|s_{1:t-1}). \end{aligned} \tag{5.21}$$

where $\mathbf{P}$ is the unknown transition probability matrix (TPM) for the switching states s_t, and it is given by,

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}. \tag{5.22}$$

It is assumed that p_{11} and p_{22} are independent. Therefore, since there are only two states, we generate samples only from $p(p_{11}|s_{1:t-1})$ and $p(p_{22}|s_{1:t-1})$.

We choose the Beta distributions for the priors $p(p_{11})$ and $p(p_{22})$:

$$p(p_{11}) = \frac{\Gamma(\eta_{11}+\eta_{12})}{\Gamma(\eta_{11})\Gamma(\eta_{12})} p_{11}^{\eta_{11}-1}(1-p_{11})^{\eta_{12}-1}, \quad \eta_{11},\eta_{12}>0, \quad 0\le p_{11}\le 1 \tag{5.23}$$

$$p(p_{22}) = \frac{\Gamma(\eta_{22}+\eta_{21})}{\Gamma(\eta_{22})\Gamma(\eta_{21})} p_{22}^{\eta_{22}-1}(1-p_{22})^{\eta_{21}-1}, \quad \eta_{22},\eta_{21}>0, \quad 0\le p_{22}\le 1. \tag{5.24}$$

The posterior distributions $p(p_{11}|s_{1:t-1})$ and $p(p_{22}|s_{1:t-1})$ are also Beta distributions. Since $p_{12} = 1 - p_{11}$ and $p_{21} = 1 - p_{22}$, we have

$$\begin{aligned} p(p_{11}|s_{1:t-1}) &\propto p(s_{1:t-1}|p_{11})p(p_{11}) \\ &\propto \left[p(s_1)\prod_{k=2}^{t-1} p(s_k=j|s_{k-1}=i)\right]\left[p_{11}^{\eta_{11}-1}p_{12}^{\eta_{12}-1}\right] \\ &\propto \left[p(s_1)\prod_{k=2}^{t-1} p_{11}^{I_k(c_{11})}p_{12}^{I_k(c_{12})}\right]\left[p_{11}^{\eta_{11}-1}p_{12}^{\eta_{12}-1}\right] \\ &\propto p_{11}^{\eta_{11}+\sum_{k=1}^{t-1} I_k(c_{11})-1} p_{12}^{\eta_{12}+\sum_{k=1}^{t-1} I_k(c_{12})-1} \\ &\propto Beta\left(\eta_{11}+\sum_{k=1}^{t-1} I_k(c_{11}), \eta_{12}+\sum_{k=1}^{t-1} I_k(c_{12})\right). \end{aligned} \tag{5.25}$$

Similarly, the posterior distribution of $p(p_{22}|s_{1:t-1})$ is also Beta,

$$\begin{aligned} p(p_{22}|s_{1:t-1}) &\propto p(s_{1:t-1}|p_{22})p(p_{22}) \\ &\propto Beta\left(\eta_{22}+\sum_{k=1}^{t-1} I_k(c_{22}), \eta_{21}+\sum_{k=1}^{t-1} I_k(c_{21})\right) \end{aligned} \tag{5.26}$$

where $I_t(c_{jk})$ is the indicator variable which takes values 0 or 1, based on the transition from $(s_{t-1}=j)\rightarrow(s_t=k)$, i.e.,

$$I_t(c_{jk}) = \begin{cases} 1, & \text{when} \quad (s_{t-1}=j)\rightarrow(s_t=k) \\ 0, & \text{otherwise} \end{cases}. \tag{5.27}$$

The posterior distribution $p(s_t|s_{1:t-1},\mathbf{P},x_{0:t-1},y_{1:t})$ is given by

$$\begin{aligned} p(s_t|s_{1:t-1},\mathbf{P},x_{0:t-1},y_{1:t}) &\propto p(s_{1:t-1},\mathbf{P},x_{0:t-1},y_{1:t}|s_t)p(s_t) \\ &\propto p(s_t|s_{t-1},\mathbf{P}). \end{aligned} \tag{5.28}$$

The posterior state distribution $p(x_t|x_{0:t-1},s_t)$ is derived as,

$$p(x_t|x_{0:t-1},s_t) \sim \begin{cases} t_{\nu_1}(x_t|m_{1,t-1},r_{t-1}), & \text{when} \quad s_t=1 \\ t_{\nu_2}(x_t|m_{2,t-1},r_{t-1}), & \text{when} \quad s_t=2 \end{cases}. \tag{5.29}$$

We generate particles for p_{11}, p_{22}, s_t and x_t from (5.25,5.26,5.28,5.29), respectively, and the weights are computed according to

$$\begin{aligned} w_t^i &\propto w_{t-1}^i \frac{p(x_t^i, s_t^i, \mathbf{P}^i | x_{0:t-1}^i, s_{1:t-1}^i, y_{1:t})}{q(x_t^i, s_t^i, \mathbf{P}^i | x_{0:t-1}^i, s_{1:t-1}^i, y_{1:t})} \\ &\propto w_{t-1}^i p(y_t | x_t^i). \end{aligned} \tag{5.30}$$

We develop RB-APF algorithms for sequential estimation for two cases, (1) when $\mathbf{P}$ is assumed known, and (2) when $\mathbf{P}$ is unknown.

5.4.1 RB-APF Method for Known TPM

We develop an RB-APF algorithm where the transition probabilities, $\mathbf{P}$, are assumed known. We sample only s_t^i and x_t^i from the the posterior PDF $p(x_t, s_t | x_{0:t-1}, s_{1:t-1}, y_{1:t}, \mathbf{P})$,

$$p(x_t, s_t | x_{0:t-1}, s_{1:t-1}, y_{1:t}, P) \propto p(y_t | x_t) p(x_t | x_{0:t-1}, s_t) p(s_t | s_{t-1}, \mathbf{P}) \tag{5.31}$$

The following are the steps for the implementation of the algorithm:

- Suppose, at time $t-1$, we have weighted particles from the posterior PDF, i.e., $\{x_{t-1}^i, s_{t-1}^i, w_{t-1}^i\}_{i=1}^N \approx p(x_{t-1}, s_{t-1} | x_{0:t-2}^i, s_{1:t-2}^i, y_{1:t-1}, \mathbf{P})$

1. We get the prior estimates of $\widehat{s_t}^i$ and $\widehat{\mu}_t^i = E(x_t | x_{t-1}^i, \widehat{s_t}^i, \widehat{\boldsymbol{\psi}}_{t-1}^i)$
   - $\widehat{s_t}^i = \text{argmax}\, p(s_t = j | s_{t-1} = s_{t-1}^i, \mathbf{P}); j \in (1, 2)$
   - $\widehat{\mu}_t^i = \widehat{\alpha}_{\widehat{s_t^i}}^i + \widehat{\beta}^i x_{t-1}^i$
2. Compute the first stage weights: $L_t^i \propto w_{t-1}^i p(y_t | \widehat{\mu}_t^i), \quad i = 1, \ldots, N$
3. Normalize the weights: $\widehat{L}_t^i = \frac{L_t^i}{\sum_{j=1}^N L_t^j}, \quad i = 1, \ldots, N$
4. Resampling: sample the indexes $J_i \in (1, \ldots, N)$ with $\widehat{L}_t$, J_i are auxiliary variables.
5. Propagate the regime-switching states: $s_t^i \sim p(s_t | s_{t-1}^{J_i}, \mathbf{P})$
6. Propagate the volatility state: $x_t^i \sim p(x_t | x_{0:t-1}^{J_i}, s_t^i,), \quad i = 1, \ldots, N$
7. Estimate $\widehat{\boldsymbol{\psi}}^i = (\widehat{\alpha}_1^i, \widehat{\alpha}_2^i, \widehat{\beta}^i)$ from $x_{0:t}^i$ with the sequential least squares (SLS) method.

8. Evaluate the second stage weights: $w_t^i \propto \frac{p(y_t|x_t^i, \widehat{\boldsymbol{\psi}}^i)}{p(y_t|\widehat{\mu}_t^{J_i})}, \quad i = 1, \ldots, N$

9. Normalize the weights: $\widetilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}, \quad i = 1, \ldots, N$

10. The posterior at time t is represented by

$$\{x_t^i, s_t^i, \widetilde{w}_t^i\}_{i=1}^N \approx p(x_t, s_t | x_{0:t-1}^i, s_{1:t-1}^i, y_{1:t}, \mathbf{P})$$

5.4.2 RB-APF Method for Unknown TPM

We, now present the RB-APF algorithm for the case of unknown TPM, where we sample from the posterior PDF $p(x_t, s_t, \mathbf{P}|x_{0:t-1}, s_{1:t-1}, y_{1:t})$. The algorithm are as follows:

- Suppose, at time $t-1$ we have weighted particles from the posterior PDF, i.e., $\{x_{t-1}^i, s_{t-1}^i, P^i, w_{t-1}^i\}_{i=1}^N \approx p(x_{t-1}, s_{t-1}, \mathbf{P}|x_{0:t-2}^i, s_{1:t-2}^i, y_{1:t-1})$

1. We get the prior estimates of $\widehat{s_t}^i$ and $\widehat{\mu}_t^i = E(x_t|x_{t-1}^i, \widehat{s_t}^i, \widehat{\boldsymbol{\psi}}_{t-1}^i)$

   - $\widehat{s_t}^i = \text{argmax}\, p(s_t = j|s_{t-1} = s_{t-1}^i, \mathbf{P}^i); j \in (1, 2)$
   - $\widehat{\mu}_t^i = \widehat{\alpha}_{\widehat{s_t^i}}^i + \widehat{\beta}^i x_{t-1}^i$

2. Compute the first stage weights: $L_t^i \propto w_{t-1}^i p(y_t|\widehat{\mu}_t^i), \quad i = 1, \ldots, N$

3. Normalize the weights: $\widehat{L}_t^i = \frac{L_t^i}{\sum_{j=1}^N L_t^j}, \quad i = 1, \ldots, N$

4. Resampling: sample the indexes $J_i \in (1, \ldots, N)$ with $\widehat{L}_t$, J_i are auxiliary variables.

5. Propagate the transition probability: $p_{11}^i \sim p(p_{11}|s_{1:t-1}^{J_i}), \quad i = 1, \ldots, N$

6. Propagate the transition probability: $p_{22}^i \sim p(p_{22}|s_{1:t-1}^{J_i}), \quad i = 1, \ldots, N$

7. Propagate the regime-switching states: $s_t^i \sim p(s_t|s_{t-1}^{J_i}, P^i)$

8. Propagate the volatility state: $x_t^i \sim p(x_t|x_{0:t-1}^{J_i}, s_t^i,), \quad i = 1, \ldots, N$

9. Estimate $\widehat{\boldsymbol{\psi}}^i = (\widehat{\alpha}_1^i, \widehat{\alpha}_2^i, \widehat{\beta}^i)$ from $x_{0:t}^i$ with the SLS method

10. Evaluate the second stage weights: $w_t^i \propto \frac{p(y_t|x_t^i, \widehat{\boldsymbol{\psi}}^i)}{p(y_t|\widehat{\mu}_t^{J_i})}, \quad i = 1, \ldots, N$

11. Normalize the weights: $\widetilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}, \quad i = 1, \ldots, N$

12. The posterior at time t is represented by

$$\{x_t^i, s_t^i, P^i, \widetilde{w}_t^i\}_{i=1}^N \approx p(x_t, s_t, P | x_{0:t-1}^i, s_{1:t-1}^i, y_{1:t}).$$

5.5 Simulation Results

We implement the APF algorithm developed by Liu and West (LW-APF) and compared its performances with the RB-APF algorithm, where $\mathbf{P}$ is assumed known. With known $\mathbf{P}$, the dimension of $\boldsymbol{\theta} = (\gamma_1,\ \gamma_2,\ \beta,\ \sigma_u^2)$ is reduced to four. For performance comparison, simulated results on **dataset 1RS** is presented. The **dataset 1RS** has less fluctuation in the observation series, and it has a state noise variance $\sigma_u^2 = 0.05$ (Figures 5.1). We run the simulation experiments with increasing particle sizes $N = 100 - 3000$ on these datasets, and find that performance of the algorithm is consistent. We also observe that dataset with higher fluctuations is difficult to track and estimate with good accuracy. For this type of sequence requires large particle sizes. We are able to show that RB method has advantages over LW method with respect to accuracy on the use of small particle size, as demonstrated in Figures 5.3-5.10.

For the LW-APF method, where all the unknowns $(s_t, x_t, \boldsymbol{\theta})$ were sampled and the diffused priors are used, where the means are chosen as true values. In the RB-APF method, sampling are performed only on s_t and x_t, and $\boldsymbol{\theta}$ are estimated from $x_{0:t-1}$ with the SLS method. We observe that the RB-APF method consistently perform better than the LW-APF method with the smaller particle sizes ($N = 100, 500$). However, with the increased particle size, $N = 3000$ or above, the LW method showed good performance as demonstrated in Figures 5.11-5.18. For the parameter estimate, the RB method is consistently shown to be better than LW method. We also observe that the RB method is less sensitive to the prior distributions than the LW method. Finally, the LW method is shown to be highly sensitive to the prior distribution of the noise variance (σ_u^2).

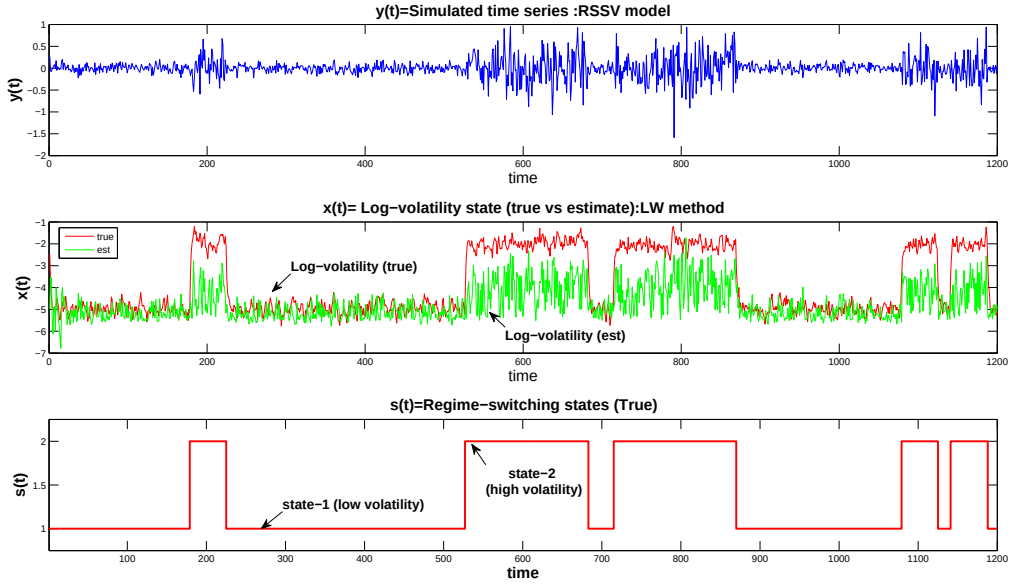

Figure 5.3: **Dataset 1RS, N=100, LW method**: Plots of simulated time series y_t, log-volatility x_t (true vs estimate), regime-switching states s_t (true), with parameters values: $\alpha_1 = -2.5, \alpha_2 = -1, \beta = 0.5, \sigma_u^2 = 0.05, p_{11} = 0.990, p_{22} = 0.985$ and $T = 1200$.

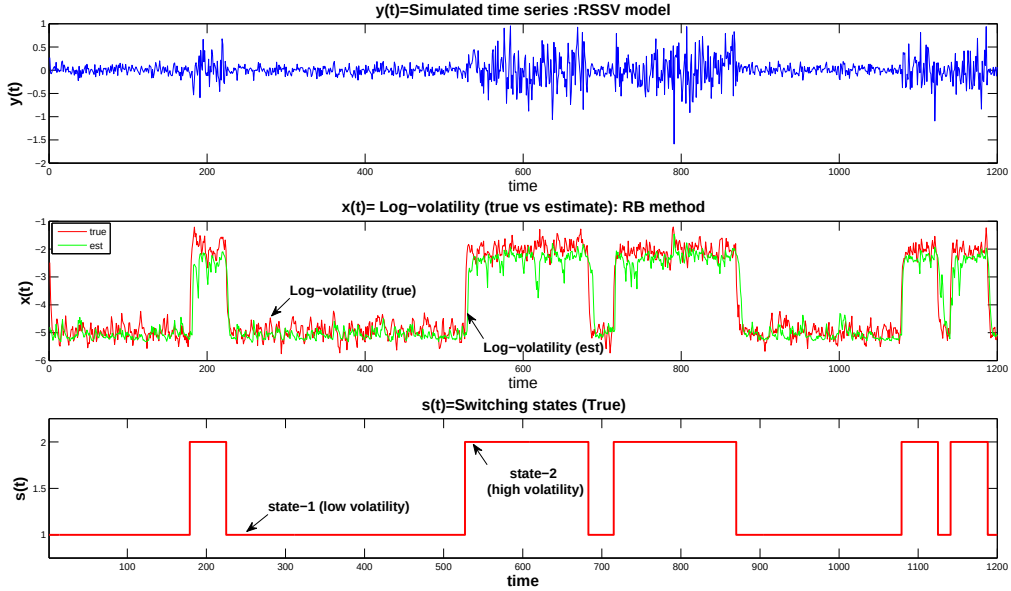

Figure 5.4: **Dataset 1RS, N=100, RB method**: Plots of simulated time series y_t, log-volatility x_t (true vs estimate), regime-switching states s_t (true), with parameters values: $\alpha_1 = -2.5, \alpha_2 = -1, \beta = 0.5, \sigma_u^2 = 0.05, p_{11} = 0.990, p_{22} = 0.985$ and $T = 1200$.

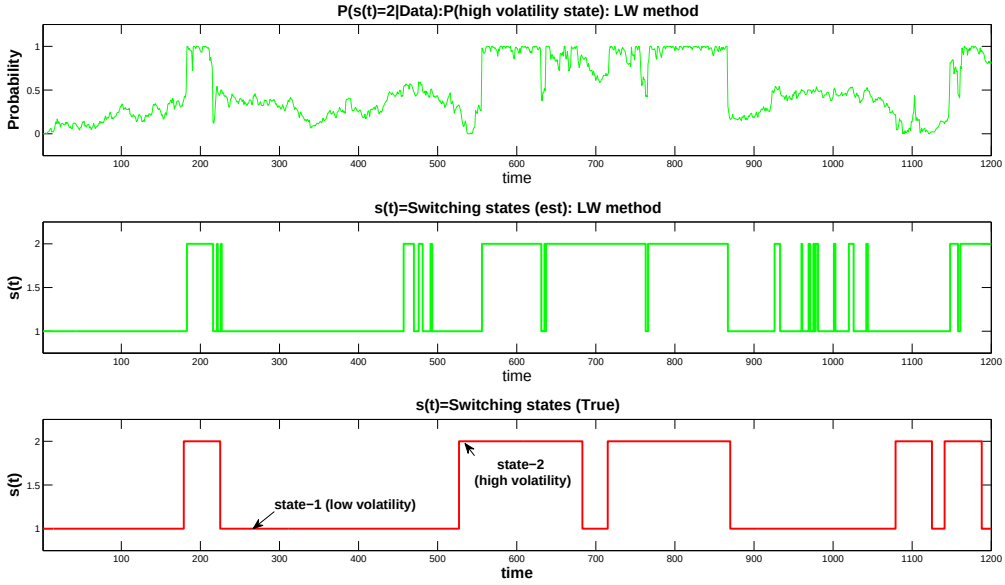

Figure 5.5: **Dataset 1RS, N=100, LW method** : Plots of estimate of $p(s_t = 2|data)$, switching states s_t (true vs estimate).

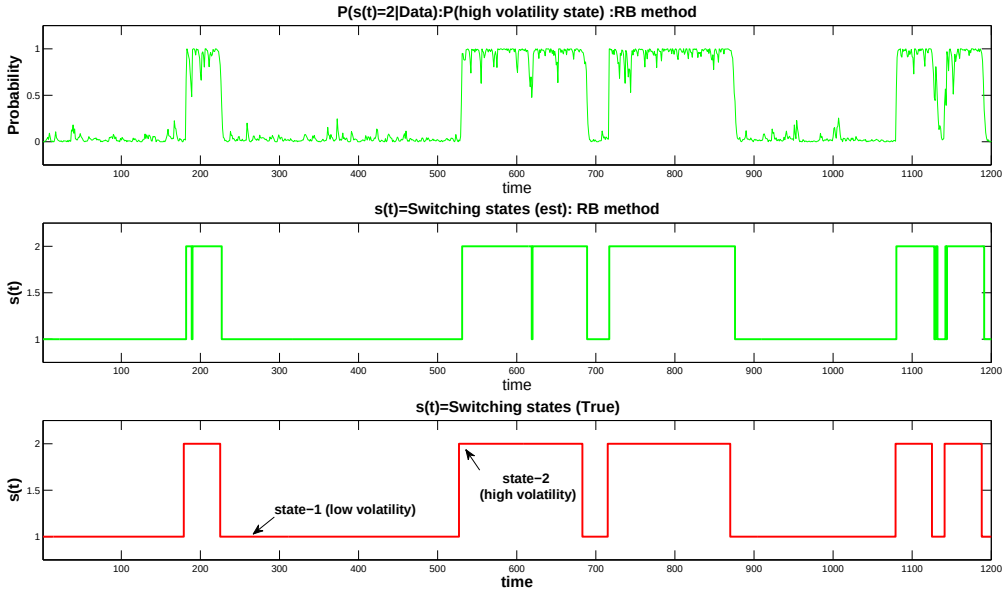

Figure 5.6: **Dataset 1RS, N=100, RB method** : Plots of estimate of $p(s_t = 2|data)$, switching states s_t (true vs estimate).

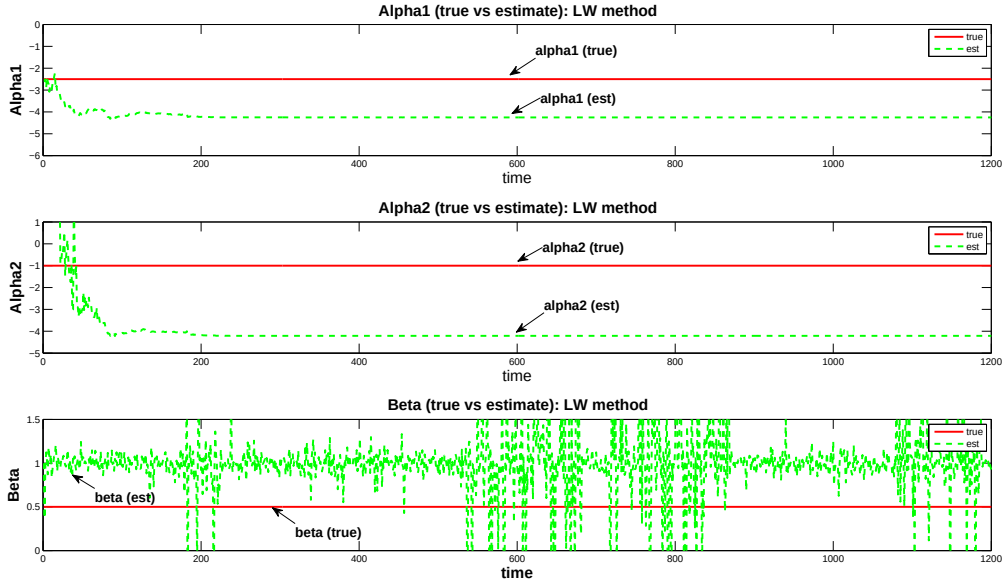

Figure 5.7: **Dataset 1RS, N=100, LW method**: Plots of parameters (true vs estimate); level parameters $\alpha 1$, $\alpha 2$, and persistence parameter β.

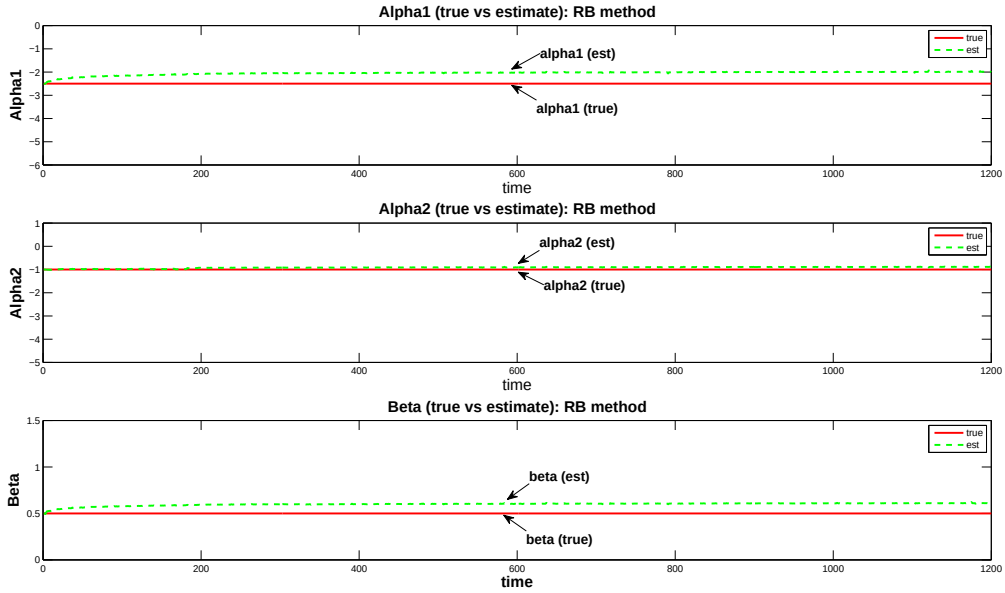

Figure 5.8: **Dataset 1RS, N=100, RB method**: Plots of parameters (true vs estimate); level parameters $\alpha 1$, $\alpha 2$, and persistence parameter β.

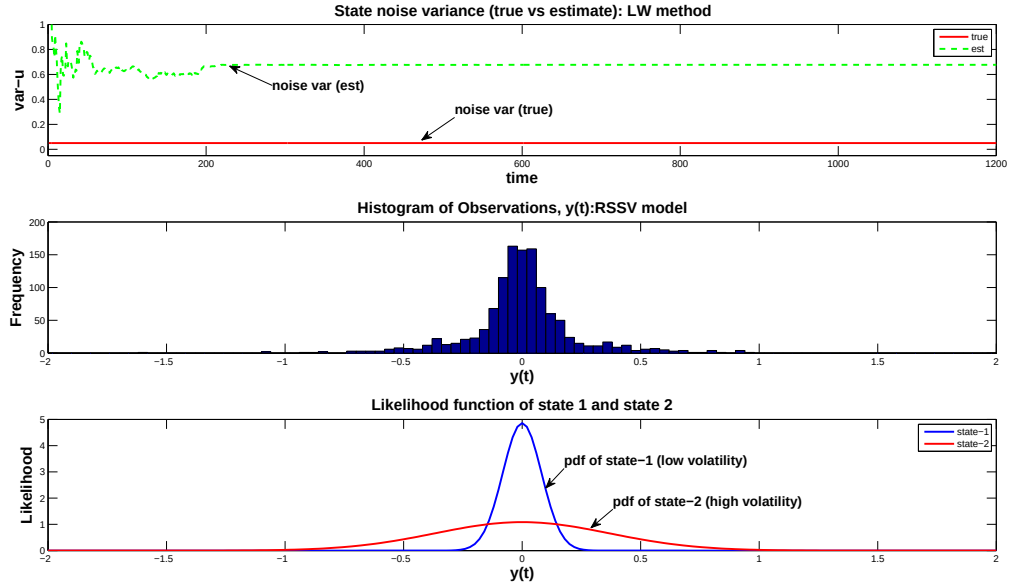

Figure 5.9: **Dataset 1RS, N=100, LW method**: Plots of state noise variance, σ_u^2(true vs estimate), histogram of time series y_t, and $p(y_t|x_t, s_t = 1)$ and $p(y_t|x_t, s_t = 2)$ are the likelihood functions of state-1 and state-2 respectively.

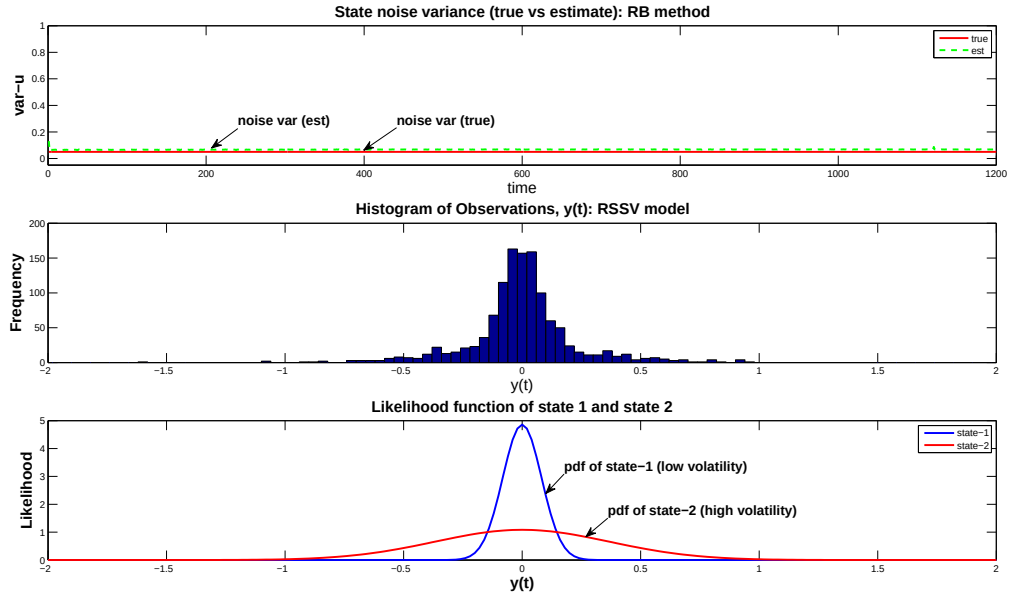

Figure 5.10: **Dataset 1RS, N=100, RB method**: Plots of state noise variance, σ_u^2(true vs estimate), histogram of time series y_t, and $p(y_t|x_t, s_t = 1)$ and $p(y_t|x_t, s_t = 2)$ are the likelihood functions of state-1 and state-2 respectively.

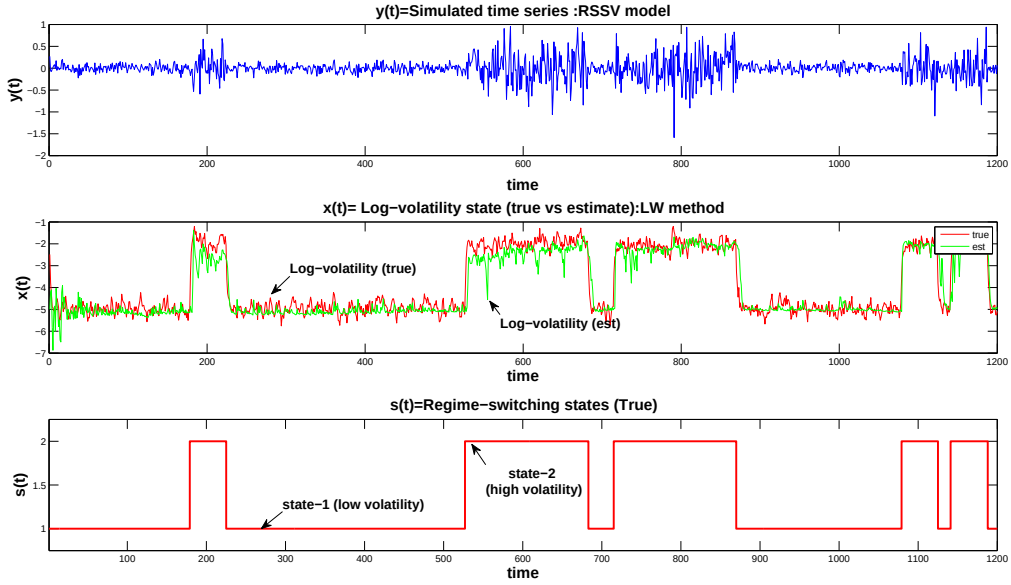

Figure 5.11: **Dataset 1RS, N=3000, LW method**: Plots of simulated time series y_t, log-volatility x_t (true vs estimate), regime-switching states s_t (true), with the parameters values; $\alpha_1 = -2.5, \alpha_2 = -1, \beta = 0.5, \sigma_u^2 = 0.05, p_{11} = 0.990, p_{22} = 0.985$ and $T = 1200$.

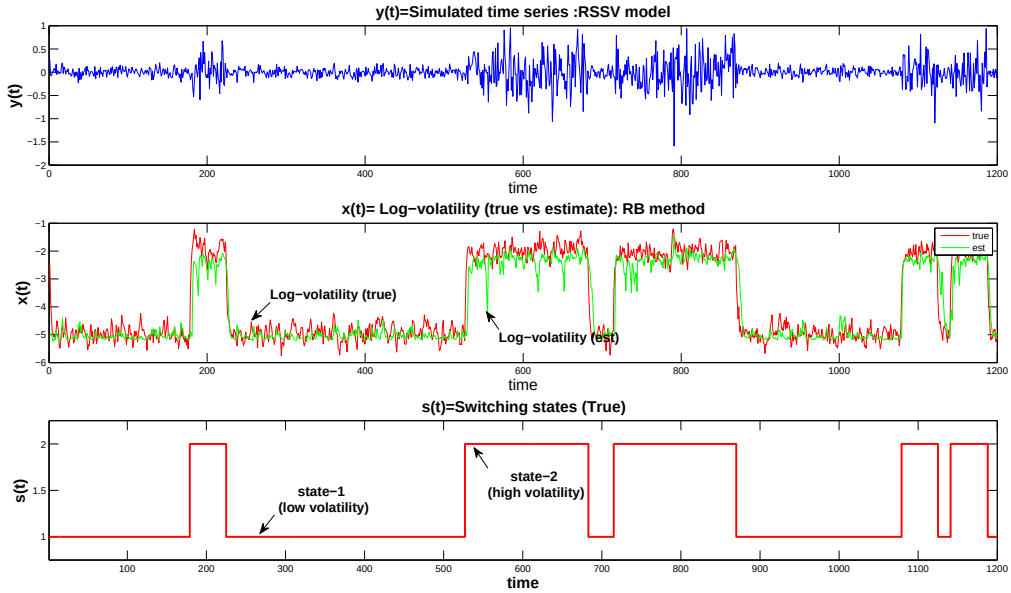

Figure 5.12: **Dataset 1RS, N=3000, RB method**: Plots of simulated time series y_t, log-volatility x_t (true vs estimate), regime-switching states s_t (true), with the parameters values; $\alpha_1 = -2.5, \alpha_2 = -1, \beta = 0.5, \sigma_u^2 = 0.05, p_{11} = 0.990, p_{22} = 0.985$ and $T = 1200$.

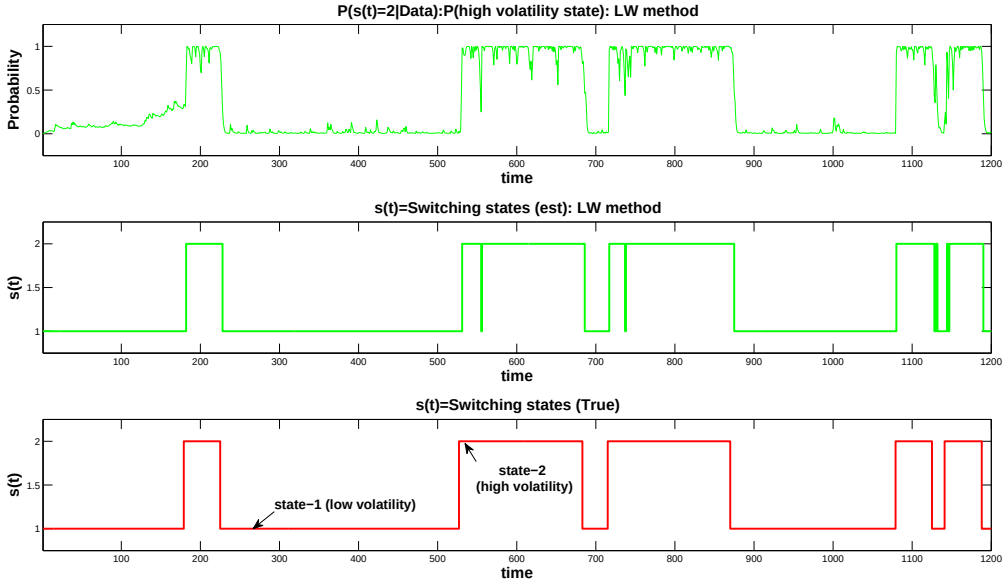

Figure 5.13: **Dataset 1RS, N=3000, LW method**: Plots of estimate of $p(s_t = 2|data)$, switching states s_t (true vs estimate).

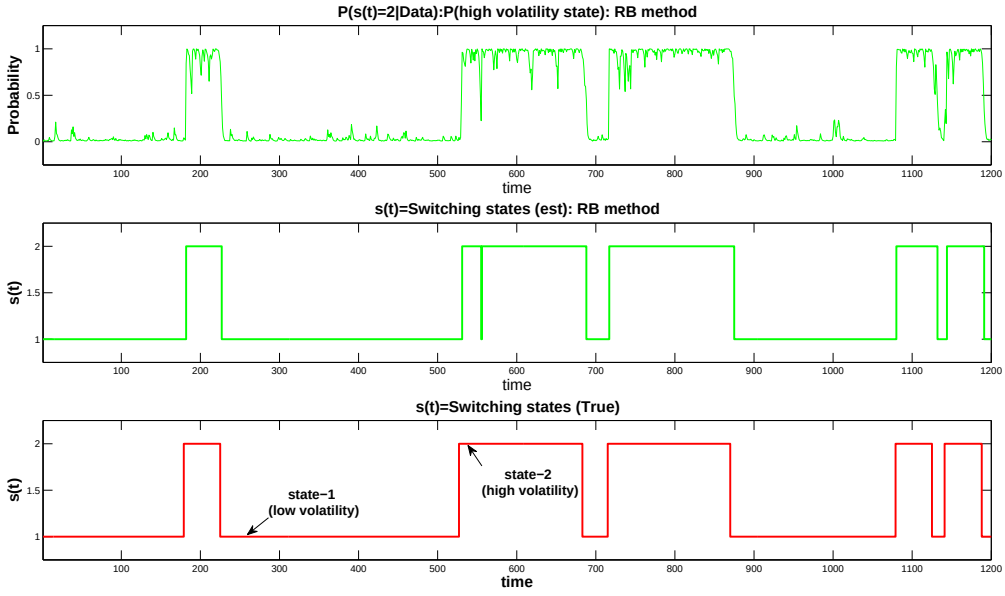

Figure 5.14: **Dataset 1RS, N=3000, RB method**: Plots of estimate of $p(s_t = 2|data)$, switching states s_t (true vs estimate).

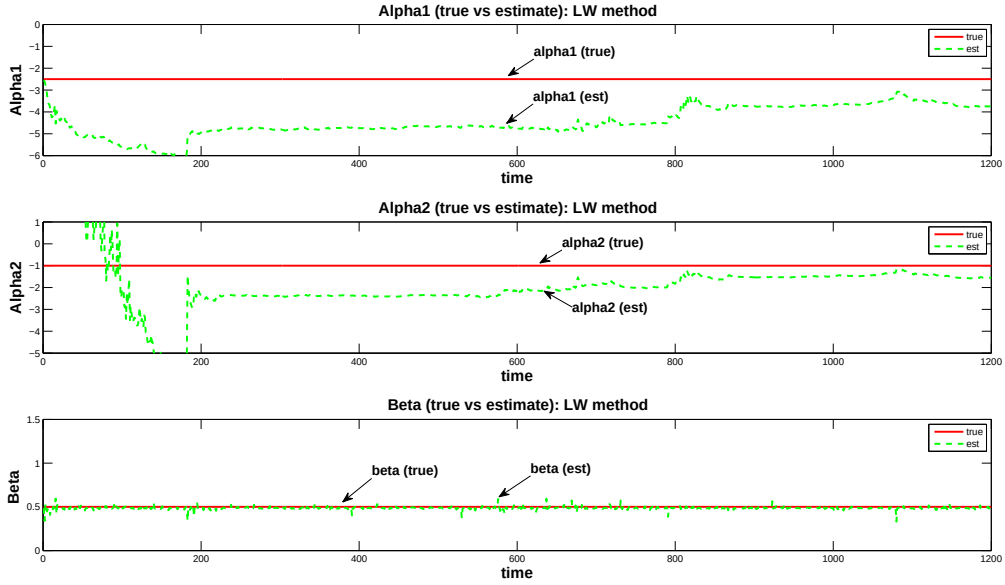

Figure 5.15: **Dataset 1RS, N=3000, LW method**: Plots of parameters (true vs estimate); level parameters $\alpha 1$, $\alpha 2$, and persistence parameter β.

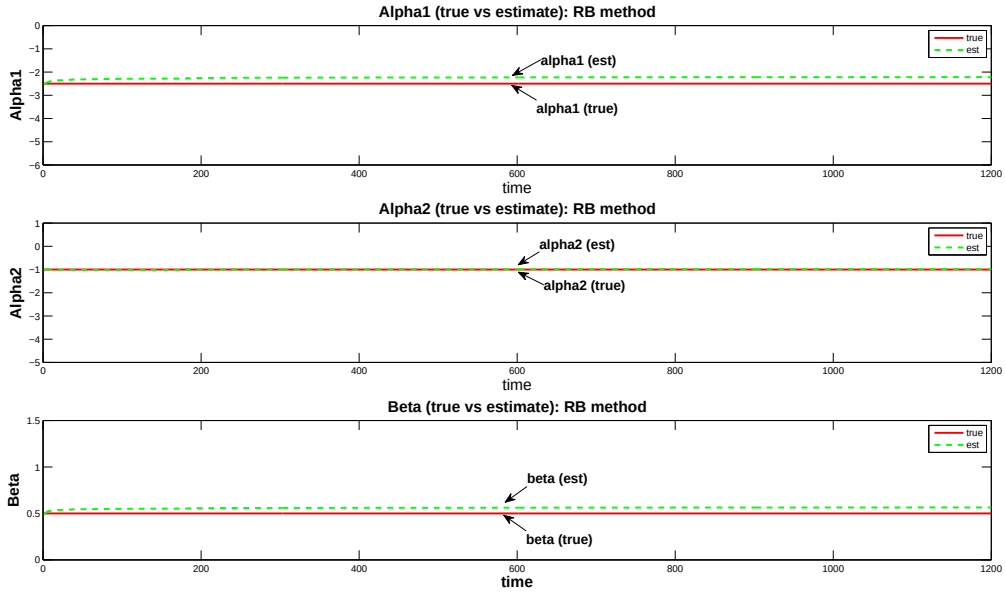

Figure 5.16: **Dataset 1RS, N=3000, RB method**: Plots of parameters (true vs estimate); level parameters $\alpha 1$, $\alpha 2$, and persistence parameter β.

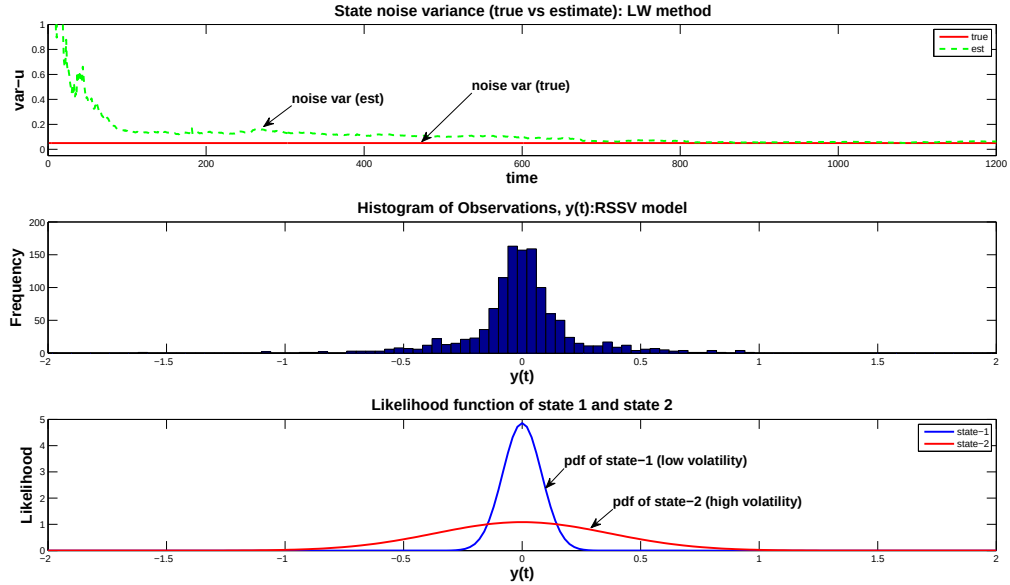

Figure 5.17: **Dataset 1RS, N=3000, LW method**: Plots of state noise variance, σ_u^2(true vs estimate), histogram of time series y_t, and $p(y_t|x_t, s_t = 1)$ and $p(y_t|x_t, s_t = 2)$ are the likelihood functions of state-1 and state-2 respectively.

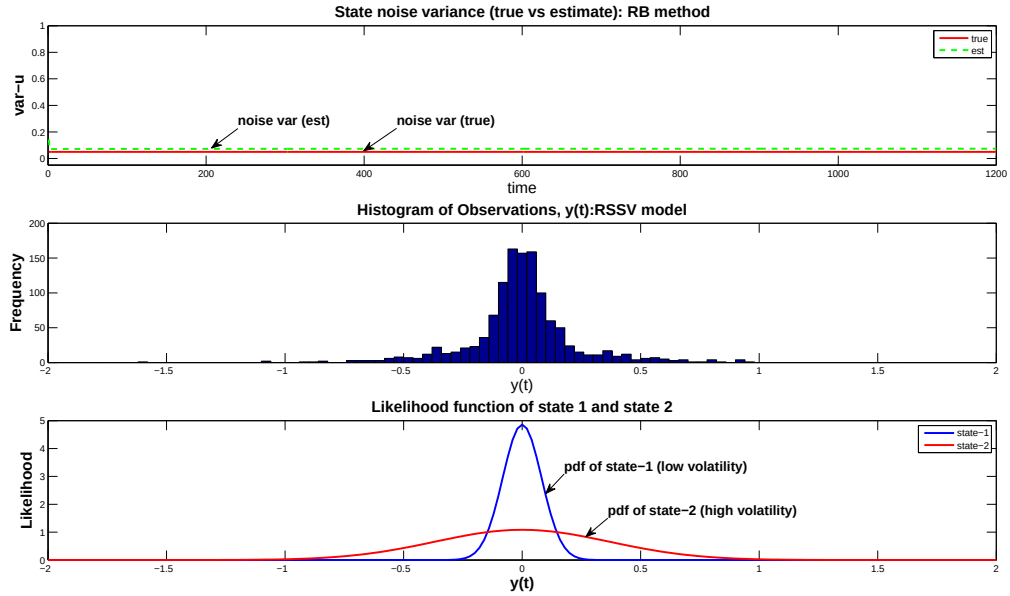

Figure 5.18: **Dataset 1RS, N=3000, RB method**: Plots of state noise variance, σ_u^2(true vs estimate), histogram of time series y_t, and $p(y_t|x_t, s_t = 1)$ and $p(y_t|x_t, s_t = 2)$ are the likelihood functions of state-1 and state-2 respectively.

Chapter 6

Conclusions and Future Work

Dynamic state space (DSS) models play an important role in many scientific and engineering disciplines including finance. Of special research interest are DSS models used for dynamic systems which are highly nonlinear and non-Gaussian. The *Kalman filter* is the optimal solution for linear and Gaussian models, but is not applicable for nonlinear and/or non-Gaussian DSS models. Therefore, for studying of the latter, many alternative approaches have been proposed. DSS models usually require sequential and/or real-time inference. The popular MCMC algorithms are not suitable for this type of data processing. By contrast, PF, which is also an MC-based methodology, is, and it produces estimates of distributions as MCMC methods. PF, however, has difficulty for sequential joint inference of states and constant parameters, and therefore a lot of effort has been dedicated to finding good solutions for such scenarios. One of them is the use of RB, which is a variance reduction technique that exploits Rao-Blackwell theorem. In this dissertation, we consider a class of discrete-time nonlinear DSS models, where the state equation is linear and contains unknown static parameters. For these models, we develop a PF algorithm where we integrate out the unknown static and nuisance parameters. This reduces the dimension of the unknown sampling space which leads to improvement in accuracy of the PF algorithm. One contribution of our work is the implementation of the RB via the *implied integration method* instead of direct integration.

As application examples, we chose the standard SV and RSSV models. The SV models are DSS models with a linear state equation in the parameters, and a nonlinear observation

equation in the state. Our main objective was to obtain sequential estimates of the posterior of the log-volatility. We applied RB with the *implied integration* method to integrate out the constant parameters. Thereby, we reduced the dimension of the unknowns, and we only had to sample the log-volatility. The reduction in the sampling space usually leads to significant improvements in accuracy and computational complexity. We also considered RSSV models, where we allowed that they have two regimes, one representing *low* and another, *high volatility* states.

We performed extensive simulations to study the proposed methods. In Chapter 4, we compared the performances of our Rao-Blackwellized PF algorithm with the PF algorithm developed by Liu and West. The latter method employs sampling both of the state and the constant and nuisance parameter vector. We performed RMSE analysis of the estimates from data records of different lengths and were able to show that our RB method consistently outperformed the algorithm of Liu and West. We also studied the performance of RB on the RSSV model. Again, we showed that our RB method outperformed the method of Liu and West.

A natural extension of our work on univariate SV models is its application to the multivariate case, where one models the changing variance of multivariate asset returns. Since financial time series exhibit heteroscedasticity, this multivariate models attempt to capture the cross-variance patterns of the time series. A class of models for this purpose is known as dynamic factor models. They try to capture the co-movements of the return series by a set of dynamic hidden (unobserved) factor process. The dynamic factor models have been one of the most powerful tools for modeling dependence and co-dependence of multivariate time series. The factor model implies that the dependence and co-dependence structure in the observable time series is explained by some common hidden factors. One interpretation is that the models attempt to explain the high dimensional covariance structure in a low dimensional space without losing much explanation power. The objective is to choose the smallest number of factors, which explain the most dependent structure of the observations.

Due to the computational advances, Bayesian inference on the dynamic factor models has received keen interest by researchers and practitioners. The factor models however have challenges in choosing the number of factors, model uncertainty, model specifications and model fitting. ML and MCMC-based methods have been applied on static and dynamic factor models on retrospective data. The dynamic structure of the factors with time varying variances has been considered, and the variances of the factors have been modeled as a log-volatility hidden process. The modeling assumption is that movements in the return series are driven by SV process as a factors. Our approach would be the application of RB on these models followed by PF. The application of RB will reduce the dimension of the unknown sampling space, and consequently, will improve the accuracy of the estimate and also will reduce the computational complexity.

Bibliography

Alspach, D. L. and Sorenson, H. W. (1972). Nonlinear bayesian estimation using gaussian sum approximation. *IEEE Transactions on Automatic Control*, 17(4):439–448.

Anderson, B. and Moore, J. (1979). *Optimal Filtering*. Dover Publishing Inc.

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

Besag, J. (1889). A candidate's formula: A curious result in Bayesian prediction. *Biometrika*, 76:183.

Bollerslev, T. (1986). Generalized auto regressive conditional heteroscedasticity. *Journal of Econometrics*, 31:307–327.

Box, G. and Tiao, G. (1992). *Bayesian Inference in Statistical Analysis*. John Wiley and Sons, Inc.

Cappe, O. and Robert, C. (2000). Markov chain monte carlo: 10 years and still running! *Journal of the American Statistical Association*, 95:1282–1286.

Carlin, B., Polson, N., and Stoffer, D. (1992). A monte carlo approach to nonnormal and nonlinear state space models. *Journal of Americal Statistical Asssociation*, 87(418):493–500.

Carvalho, C. and Lopes, H. (2006). Simulation based sequential analysis of markov switching stochastic volatility models. *Computational Statistics & Data Analysis*.

Casella, G. and Robert, C. P. (1996). Rao-Blackwellization of sampling schemes. *Biometrika*, 84:81–94.

Chib, S. (1996). Calculating posterior distributions and modal estimates in markov mixture models. *Journal of Econometrics*, 75:79–97.

Chou, R. (1988). Volatility persistence and stock valuations:some empirical evidence using garch. *Journal of Applied Econometrics*, 3:279–294.

Clark, P. (1973). A subordinated stochastic process model with finite variance for speculative prices. *Econometrica*, 41:135–155.

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

Djuric, P. and Bugallo, M. (2009). *"Particle Filtering " in Advances in Adaptive Filtering*, chapter 1. Wiley & Sons.

Doucet, A., de Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo Methods in Practice*. Springer, New York.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing.*, 10(3):197–208.

Engle, R. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of the united kingdom inflation. *Econometrica*, 50(4):987–1007.

Evans, M. and Swartz, T., editors (2000). *Approximating Integrals via Monte Carlo and Deterministic Methods*. Oxford University Press, Oxford, UK.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Geweke, J. (1989). Bayesian inference in econometrics models using monte carlo integration. *Econometrica*, 57:1317–1339.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target – monte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society B*, 63:126–146.

Goldfeld, S. and Quandt, R. (1973). A markov model for switching regressions. *Journal of Econometrics*, 1(1).

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE proceedings-F*, 140(2):107–113.

Hamilton, J. and Susmel, R. (1994). Autoregressive conditional heteroskedasticity and changes in regime. *Journal of Econometrics*, 64:307–333.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2):357–384.

Harvey, A. C. (1981). *Time Series Models*. MIT Press.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their application. *Biometrika*, 57:97–109.

Jacquier, E., Polson, N., and Rossi, P. (1994). Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics*, 12(4):371–389.

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

Kim, C.-J. and Nelson, C. R. (1999). *State Space Models With Regime Switching*. The MIT Press.

Kim, S., Shephard, N., and Chib, S. (1998). Stochastic volatility: Likelihood inference and comparison with arch models. *Review of Economic Studies*, 65:361–393.

Kitagawa, G. (1987). Non-gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82(400).

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(11):1–25.

Kong, A., Liu, J., and Wong, W. (1994). Sequential imputation and bayesian missing data problems. *Journal of the American Statistical Association*, 9(278-288).

Lamoureux, C. and Lastrapes, W. (1990). Persistence in variance, structureal change and the garch model. *Journal of Business and Economic Statistics*, 8(2):225–234.

Lehmann, E. (1991). *Theory of Point Estimation*. Wadsworth & Books.

Liu, J. and West, M. (2001). *"Combined Parameter and State Estimation in Simulation Based Filtering" in Sequential Monte Carlo Methods in Practice*, chapter 10. Springer.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computation*. Springer, ist edition.

Liu, J. S. and Chen, R. (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American Stastistical Association*, 93(443):1032–1044.

Maskell, S. (2004). *"An Introduction to Particle Filters" in State Space and Unobserved Component Models*, chapter 3. Cambridge University Press.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341.

Muller, P. (1991). Monte carlo integration in general dynamic models. *Contemporary Mathematics*, 115:145–163.

Musso, C., Oudjane, N., and LeGland, F. (2001). *"Improving Regularized particle filters" in Sequential Monte Carlo Methods*, chapter 12. Springer.

Pitt, M. and Shepard, N. (1999). Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.

Poterba, L. and Summers, L. (1986). The persistence of volatility and stock market fluctuations. *American Economic Review*, 76:1142–1151.

Quandt, R. (1972). A new approach to estimating switching regressions. *Journal of the Americal Statistical Association*, 67(338).

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceeding of the IEEE*, 77(2):257–286.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. prectice hall.

Rubin, D. B. (1988). *Bayesian Statistics 3*, chapter Using the SIR algorithm to simulate posterior distributions, pages 395–402 (with discussion). Oxford: University Press.

Shephard, N. (1996). *"Ststistical aspects of ARCH and stochastic volatility" in Time Series Modles in Econometrics, Finance and other fields*, chapter 1, pages 1–67. Chapman & Hall.

Shephard, N. (2005). *Stochastic Volatility: Selected Readings*. Oxford University Press.

So, M., Lam, K., and Li, W. (1998). A stochastic volatility model with markov switching. *Journal of Business and Economic Statistics*, 16(2).

Sorenson, H. W. and Alspach, D. L. (1971). Recursive bayesian estimation using gaussian sums. *Automatica*, 7:465–479.

Storvik, G. (2002). Particle filters for state-space models with presence of unknown static parameters. *IEEE Transactions on Signal Processing*, 50(2):281–289.

Stroud, J., Paulson, N., and Muller, P. (2004). *"Practical Filtering for Stochastic Volatility Models" in State Space and Unobserved Component Models*, chapter 11. Cambridge University Press.

Taylor, S. (1982). *"Financial returns modelled by the product of two stochastic processes-a study of daily sugar prices" in Time Series Analysis: Theory and Parctice*, chapter 1, pages 203–226. Amsterdam:North-Holland.

Taylor, S. (2005). *Asset Price Dynamics, Volatility, and Prediction*. Princeton University Press.

West, M. (1992). Modelling with mixtures. *Bayesian Statistics*, 4:503–524.

West, M. (1993a). Approximating posterior distributions by mixtures. *Journal of the Royal Statistical Society B*, 55(2):409–422.

West, M. (1993b). Mixture models, monte carlo, bayesian updating and dynamic models. *Computing Science and Statistics*, 24:325–333.

West, M. (1997). *Bayesian Forecasting and Dynamic Models*. Springer Verlag, New York.

Zellner, A. (1996). *An Introduction to Bayesian Inference in Econometrics*. John Wiley and Sons, Inc.

Appendix A

Appendix

A.1 Rao-Blackwellization on the SV Model

Our SV model is defined as

$$x_t = \alpha + \beta x_{t-1} + \sigma_u u_t \quad (state \quad equation) \tag{A.1}$$

$$y_t = e^{x_t/2} v_t \quad (observation \quad equation) \tag{A.2}$$

where $x_t \in \mathbb{R}$ is a hidden state and represents the *log (volatility)*, and $u_t \in \mathbb{R}$ and $v_t \in \mathbb{R}$ are uncorrelated Gaussian noises with zero means and unit variances. We define the unknown parameter vector as $\boldsymbol{\theta} = (\alpha \ \beta \ \sigma_u^2)^\top$. Based on the assumptions, the probability distributions obtained from the state and observation equations are,

$$p(x_t|x_{0:t-1}, \boldsymbol{\theta}) \sim \mathcal{N}(x_t|\alpha + \beta x_{t-1}, \sigma_u^2) \tag{A.3}$$

$$p(y_t|x_t) \sim \mathcal{N}(y_t|0, e^{x_t}) \tag{A.4}$$

where from the state equation (A.1) we can write

$$E(x_t|x_{0:t-1}, \boldsymbol{\theta}) = \alpha + \beta x_{t-1}$$

$$Var(x_t|x_{0:t-1}, \boldsymbol{\theta}) = \sigma_u^2$$

and, from the observation equation (A.2),

$$\begin{aligned} E(y_t|x_t) &= E\left(e^{\frac{x_t}{2}}v_t|x_t\right) \\ &= e^{\frac{x_t}{2}}E(v_t); \quad E(v_t)=0 \\ &= 0 \\ V(y_t|x_t) &= E\left[\left(e^{\frac{x_t}{2}}v_t\right)^2|x_t\right] \\ &= E\left[e^{x_t}v_t^2|x_t\right] \\ &= e^{x_t}E\left(v_t^2\right); \quad E\left(v_t^2\right)=1 \\ &= e^{x_t}. \end{aligned}$$

Rao-Blackwellization (RB) is a technique which integrates out the *nuisance parameters* from the model in order to obtain a marginal distribution free from these parameters. The necessary condition for applying RB is that, one is able to carry out the integration analytically. Here, we integrate out the unknown static parameters $\boldsymbol{\theta} = (\alpha, \beta, \sigma_u^2)$ which belongs to the state equation. We perform this integration using the implied integration method (Besag, 1889).

With the application of RB in the dynamic state equation, our goal is to construct a marginal posterior density of the state, i.e.,

$$\begin{aligned} p(x_t|x_{0:t-1}) &= \int_{\boldsymbol{\theta}} p(x_t, \boldsymbol{\theta}|x_{0:t-1})\mathrm{d}\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\theta}} p(x_t|\boldsymbol{\theta}, x_{0:t-1})p(\boldsymbol{\theta}|x_{0:t-1})\mathrm{d}\boldsymbol{\theta} \\ &\propto \int_{\boldsymbol{\theta}} p(x_t|x_{0:t-1}, \boldsymbol{\theta})p(x_{0:t-1}|\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \end{aligned} \tag{A.5}$$

where the posterior of $\boldsymbol{\theta}$ is expressed as,

$$\underbrace{p(\boldsymbol{\theta}|x_{0:t-1})}_{posterior} \propto \underbrace{p(x_{0:t-1}|\boldsymbol{\theta})}_{likelihood}\underbrace{p(\boldsymbol{\theta})}_{prior} \tag{A.6}$$

According to Bayes' theorem,

$$\begin{aligned} p(\boldsymbol{\theta}|x_{0:t}) &= \frac{p(\boldsymbol{\theta}, x_{0:t})}{p(x_{0:t})} \\ &= \frac{p(x_t|x_{0:t-1}, \boldsymbol{\theta})p(x_{0:t-1}, \boldsymbol{\theta})}{p(x_t|x_{0:t-1})p(x_{0:t-1})} \\ &= \frac{p(x_t|x_{0:t-1}, \boldsymbol{\theta})p(x_{0:t-1}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(x_t|x_{0:t-1})p(x_{0:t-1})} \\ &= \frac{p(x_t|x_{0:t-1}, \boldsymbol{\theta})p(\boldsymbol{\theta}|x_{0:t-1})}{p(x_t|x_{0:t-1})} \end{aligned} \tag{A.7}$$

Hence, we deduce that

$$p(x_t|x_{0:t-1}) = \frac{\overbrace{p(x_t|x_{0:t-1}, \boldsymbol{\theta})}^{(1)}\overbrace{p(\boldsymbol{\theta}|x_{0:t-1})}^{(2)}}{\underbrace{p(\boldsymbol{\theta}|x_{0:t})}_{(3)}}. \tag{A.8}$$

To obtain the marginal posterior distribution of the state, $p(x_t|x_{0:t-1})$, we need to obtain the respective PDF of (1), (2) and (3) of the RHS in (A.8). We derive those PDFs and substitute them in (A.8) for evaluation. The PDFs of (1), (2), and (3) are derived as,

$$(1) \quad p(x_t|x_{0:t-1}, \boldsymbol{\theta}) \sim N(x_t|\alpha + \beta x_{t-1}, \sigma_u^2) \tag{A.9}$$

With Bayes' theorem,

$$\begin{aligned} (2) \quad \underbrace{p(\boldsymbol{\theta}|x_{0:t-1})}_{posterior} &\propto \underbrace{p(x_{0:t-1}|\boldsymbol{\theta})}_{likelihood}\underbrace{p(\boldsymbol{\theta})}_{prior} \\ &\propto \underbrace{p(x_{0:t-1}|\boldsymbol{\theta})}_{Gaussian}\underbrace{p(\alpha, \beta)}_{constant}\underbrace{p(\sigma_u^2)}_{\mathcal{IG}} \\ &\propto p(x_0|\boldsymbol{\theta})\prod_{k=1}^{t-1} p(x_k|x_{0:k-1}, \boldsymbol{\theta})p(\sigma_u^2) \\ &\propto \frac{1}{\sqrt{2\pi\sigma_u^2}} e^{-\frac{x_0^2}{2\sigma_u^2}} \left(\frac{1}{\sqrt{2\pi\sigma_u^2}}\right)^{t-1} e^{-\frac{1}{2\sigma_u^2}\sum_{k=1}^{t-1}(x_k-\alpha-\beta x_{k-1})^2} \left(\sigma_u^2\right)^{-(a+1)} e^{-\frac{1}{\sigma_u^2 b}} \\ &= \left(\frac{1}{\sqrt{2\pi\sigma_u^2}}\right)^{t} e^{-\frac{1}{2\sigma_u^2}\left[x_0^2+\sum_{k=1}^{t-1}(x_k-\alpha-\beta x_{k-1})^2\right]} \left(\sigma_u^2\right)^{-(a+1)} e^{-\frac{1}{\sigma_u^2 b}} \\ &\propto \left(\sigma_u^2\right)^{-(a+1+\frac{t}{2})} e^{-\frac{1}{\sigma_u^2}\left(\frac{D_{t-1}}{2}+\frac{1}{b}\right)} \\ &= \left(\sigma_u^2\right)^{-(\widetilde{a}_{t-1}+1)} e^{-\frac{1}{\sigma_u^2 \widetilde{b}_{t-1}}} \end{aligned}$$

where

$$\begin{aligned}\widetilde{a}_{t-1} &= a+\frac{t}{2}\\ \widetilde{b}_{t-1} &= \frac{2b}{2+bD_{t-1}}\\ D_{t-1} &= (x_0)^2+\sum_{k=1}^{t-1}(x_k-\alpha-\beta x_{k-1})^2\end{aligned}$$

and $\mathcal{IG}$ in the second line stands for *Inverse-Gamma* distribution.

Hence,

$$p(\boldsymbol{\theta}|x_{0:t-1}) \propto \left(\sigma_u^2\right)^{-(\widetilde{a}_{t-1}+1)} e^{-\frac{1}{\sigma_u^2 \widetilde{b}_{t-1}}} \tag{A.10}$$

(3) Similarly, $p(\boldsymbol{\theta}|x_{0:t-1})$ has analogous expression, i.e.,

$$p(\boldsymbol{\theta}|x_{0:t}) \propto \left(\sigma_u^2\right)^{-(\widetilde{a}_t+1)} e^{-\frac{1}{\sigma_u^2 \widetilde{b}_t}} \tag{A.11}$$

where

$$\begin{aligned}\widetilde{a}_t &= a+\frac{t+1}{2}\\ \widetilde{b}_t &= \frac{2b}{2+bD_t}\\ D_t &= (x_0)^2+\sum_{k=1}^{t}(x_k-\alpha-\beta x_{k-1})^2\end{aligned}$$

Finding the distribution of $p(x_t|x_{0:t-1})$ in (A.8), we need to find the *normalizing constants* in (A.10) and (A.11). To obtain this normalizing constants we write the posterior of $\boldsymbol{\theta}$ as,

$$p(\boldsymbol{\theta}|x_{0:t-1}) = \underbrace{p(\boldsymbol{\mu}|\sigma_u^2, x_{0:t-1})}_{Gaussian}\underbrace{p(\sigma_u^2|x_{0:t-1})}_{\mathcal{IG}} \tag{A.12}$$

of which the first density on the right is a two-dimensional Gaussian with mean $\widehat{\boldsymbol{\mu}}_{t-1}$ and covariance matrix $\widehat{\mathbf{C}}_{t-1}$, where $\boldsymbol{\mu} = [\alpha\ \beta]^\top$, and the second density is $\mathcal{IG}$,

$$p(\boldsymbol{\mu}|\sigma_u^2, x_{0:t-1}) = \frac{1}{2\pi|\widehat{\boldsymbol{C}}_{t-1}|^{\frac{1}{2}}} e^{-\frac{1}{2}[(\boldsymbol{\mu}-\widehat{\boldsymbol{\mu}})^\top \widehat{\boldsymbol{C}}_{t-1}^{-1}(\boldsymbol{\mu}-\widehat{\boldsymbol{\mu}})]} \tag{A.13}$$

where

$$\widehat{\boldsymbol{\mu}}_{t-1} = \left(\boldsymbol{H}_{t-1}^\top \boldsymbol{H}_{t-1}\right)^{-1} \boldsymbol{H}_{t-1}^\top \boldsymbol{x}_{t-1} \tag{A.14}$$

and

$$\widehat{\boldsymbol{C}}_{t-1} \;=\; \sigma_u^2 \left(\boldsymbol{H}_{t-1}^\top \boldsymbol{H}_{t-1}\right)^{-1} \tag{A.15}$$

and where the vector $\boldsymbol{x}_{t-1} = [x_0\, x_1\, \cdots\, x_{t-1}]^\top$ and the matrix $\boldsymbol{H}_{t-1}$ is defined as,

$$\boldsymbol{H}_{t-1} = \begin{bmatrix} 1 & 0 \\ 1 & x_0 \\ 1 & x_1 \\ . & . \\ . & . \\ 1 & x_{t-2} \end{bmatrix} \tag{A.16}$$

The second density is an Inverse-Gamma given by

$$p(\sigma_u^2|x_{0:t-1}) \;=\; \frac{1}{\Gamma(a_{t-1}) b_{t-1}^{a_{t-1}}} (\sigma_u^2)^{-(a_{t-1}+1)} e^{-\frac{1}{b_{t-1}\sigma_u^2}} \tag{A.17}$$

where

$$\begin{aligned} a_{t-1} &= a + \frac{t-1}{2} \\ b_{t-1} &= \frac{2b}{2 + b\, \boldsymbol{x}_{t-1}^\top \boldsymbol{P}_{t-1}^\perp \boldsymbol{x}_{t-1}} \end{aligned}$$

$$\boldsymbol{P}_{t-1}^\perp \;=\; \mathbf{I} - \boldsymbol{H}_{t-1} \left(\boldsymbol{H}_{t-1}^\top \boldsymbol{H}_{t-1}\right)^{-1} \boldsymbol{H}_{t-1}^\top \tag{A.18}$$

and $\mathbf{I}$ is the identity matrix.

Using the principle in (A.12), the RHS of (A.8) can be rewritten as

$$p(x_t|x_{0:t-1}) = \frac{\overbrace{p(x_t|x_{0:t-1}, \boldsymbol{\theta})}^{Gaussian}\, \overbrace{p(\boldsymbol{\mu}|\sigma_u^2, x_{0:t-1})}^{Gaussian}\, \overbrace{p(\sigma_u^2|x_{0:t-1})}^{\mathcal{IG}}}{\underbrace{p(\boldsymbol{\mu}|\sigma_u^2, x_{0:t})}_{Gaussian}\, \underbrace{p(\sigma_u^2|x_{0:t})}_{\mathcal{IG}}} \tag{A.19}$$

and the evaluation of the RHS in (A.19) with the respective distributions results in the desired PDF, $p(x_t|x_{0:t-1})$, which is free from the parameters $\boldsymbol{\theta}$.

Consider $p(x|m,\sigma^2) \sim \mathcal{N}(m,\sigma^2)$ and $p(\sigma^2|a,b) \sim \mathcal{IG}(a,b)$, where the Inverse-Gamma distribution is defined as

$$p(\sigma_u^2|a,b) = \frac{1}{\Gamma(a)b^a}(\sigma^2)^{-(a+1)}e^{-(\frac{1}{\sigma^2 b})} \tag{A.20}$$

Using the Bayes' principle, the marginal distribution $p(x|m)$ can be written as,

$$p(x) = \frac{\overbrace{p(x|m,\sigma^2)}^{Gaussian}\overbrace{p(\sigma^2)}^{\mathcal{IG}}}{\underbrace{p(\sigma^2|x,m)}_{\mathcal{IG}}} \tag{A.21}$$

where the parameter m in $p(x)$ is suppressed for notational convenience. The posterior distribution $p(\sigma^2|x,m)$ can be shown as,

$$\begin{aligned} p(\sigma^2|x,m) &\propto \underbrace{p(x|m,\sigma^2)}_{Gaussian}\underbrace{p(\sigma^2)}_{\mathcal{IG}(a,b)} \\ &\propto \mathcal{IG}(\widetilde{a},\widetilde{b}) \end{aligned} \tag{A.22}$$

where

$$\begin{aligned} \widetilde{a} &= a + \frac{1}{2} \\ \widetilde{b} &= \frac{2b}{2 + b(x-m)^2} \end{aligned}$$

It can be shown that (West, 1997, p. 641-42), the marginal distribution $p(x)$ is derived as,

$$p(x) \propto \left(\widetilde{b}\right)^{-\widetilde{a}} \tag{A.23}$$

$$\begin{aligned} &= (2a + (x-m)^2 ab)^{-\frac{2a+1}{2}} \\ &\propto (1 + \frac{(x-m)^2}{\nu r})^{-\frac{\nu+1}{2}} \\ &\propto t_\nu(m,r) \end{aligned} \tag{A.24}$$

which is a non-standard *t-distribution*, with $\nu = 2a$ is degrees of freedom, m is mode and $r = \frac{1}{ab}$ is a scale parameter.

Similarly, from (A.19), we have

$$p(x_t|x_{0:t-1}, \boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{h}_t^\top \boldsymbol{\mu}, \sigma_u^2) \tag{A.25}$$

where $\boldsymbol{h}_t^\top = [1\ x_{t-1}]$, and the other distributions in (A.19) are,

$$p(\boldsymbol{\mu}|x_{0:t-1}, \sigma_u^2) \sim \mathcal{N}(\widehat{\boldsymbol{\mu}}_{t-1}, \widehat{\boldsymbol{C}}_{t-1}) \tag{A.26}$$

$$p(\sigma_u^2|x_{0:t-1}) \sim \mathcal{IG}(a_{t-1}, b_{t-1}) \tag{A.27}$$

$$p(\boldsymbol{\mu}|x_{0:t}, \sigma_u^2) \sim \mathcal{N}(\widehat{\boldsymbol{\mu}}_t, \widehat{\boldsymbol{C}}_t) \tag{A.28}$$

and

$$p(\sigma_u^2|x_{0:t}) \sim \mathcal{IG}(a_t, b_t) \tag{A.29}$$

where $\widehat{\boldsymbol{\mu}}_{t-1} = [\widehat{\alpha}_{t-1}\ \widehat{\beta}_{t-1}]$, $a_t = a + \frac{t}{2}$, and the other parameters are already defined.

Then, as in (West, 1997, p. 641-42), after evaluation of the RHS in (A.19) with these respective PDFs, $p(x_t|x_{0:t-1})$ is a non-standard *t-distribution* with ν_{t-1} degrees of freedom, mode m_{t-1}, and scale r_{t-1}, i.e.,

$$p(x_t|x_{0:t-1}) \sim t_{\nu_{t-1}}(m_{t-1}, r_{t-1}) \tag{A.30}$$

where

$$\begin{aligned}
\nu_{t-1} &= 2a + t - 1 \\
m_{t-1} &= \boldsymbol{h}_t^\top \widehat{\boldsymbol{\mu}}_{t-1} \\
r_{t-1} &= \frac{1}{a_{t-1} b_{t-1}} \\
&= \frac{\frac{2}{b} + \boldsymbol{x}_{t-1}^\top \boldsymbol{P}_{t-1}^\perp \boldsymbol{x}_{t-1}}{\nu_{t-1}} \\
&= \frac{\frac{2}{b} + RSS_{t-1}}{\nu_{t-1}}
\end{aligned}$$

where RSS_{t-1} is the Residual Sum Squares on linear state equation. The non-standard *t-pdf* is defined as (Zellner, 1996, p. 366-67),

$$p(x|\nu, m, R) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{1}{2})\Gamma(\frac{\nu}{2})} \left(\frac{1}{\nu r}\right)^{\frac{1}{2}} \left(1 + \frac{(x-m)^2}{\nu r}\right)^{-(\frac{\nu+1}{2})} ; \quad t_\nu(m, r) \tag{A.31}$$

A.2 Derivation of Rao-Blackwelized Sequential Bayesian Filtering Stages for SV Models

The Rao-Blackwellized sequential Bayesian filtering is performed recursively in two stages: (1) *prediction* and (2) *update*. Suppose that at time $t-1$, the required PDF $p(x_{t-1}|x_{0:t-2}, y_{1:t-1})$ is available. Then the propagation or prediction on the state is done according to the following stages,

$$
\begin{aligned}
(1)\quad p(x_t|x_{0:t-1}, y_{1:t-1}) &= \int_{\boldsymbol{\theta}} \underbrace{p(x_t, \boldsymbol{\theta}|x_{0:t-1}, y_{1:t-1})}_{joint \quad posterior} \mathrm{d}\boldsymbol{\theta}; \qquad \boldsymbol{\theta} = (\alpha, \beta, \sigma_u^2) \\
&= \int_{\boldsymbol{\theta}} \underbrace{p(x_t|\boldsymbol{\theta}, x_{0:t-1}, y_{1:t-1})}_{posterior} \underbrace{p(\boldsymbol{\theta}|x_{0:t-1}, y_{1:t-1})}_{posterior} \mathrm{d}\boldsymbol{\theta} \\
&\propto \int_{\boldsymbol{\theta}} p(x_t|x_{0:t-1}, y_{1:t-1}, \boldsymbol{\theta}) \underbrace{p(x_{0:t-1}, y_{1:t-1}|\boldsymbol{\theta})}_{likelihood} \underbrace{p(\boldsymbol{\theta})}_{prior} \mathrm{d}\boldsymbol{\theta} \\
&\propto \int_{\boldsymbol{\theta}} p(x_t|x_{0:t-1}, y_{1:t-1}, \boldsymbol{\theta}) p(y_{1:t-1}|x_{0:t-1}, \boldsymbol{\theta}) p(x_{0:t-1}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \\
&\propto \int_{\boldsymbol{\theta}} p(x_t|x_{0:t-1}, y_{1:t-1}, \boldsymbol{\theta}) p(y_{1:t-1}|x_{0:t-1}) p(x_{0:t-1}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \\
&\propto \int_{\boldsymbol{\theta}} p(x_{0:t-1}, y_{1:t-1}, \boldsymbol{\theta}|x_t) p(x_t) p(x_{0:t-1}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \\
&\propto \int_{\boldsymbol{\theta}} p(y_{1:t-1}|x_{0:t}, \boldsymbol{\theta}) p(x_{0:t-1, \boldsymbol{\theta}|x_t}) p(x_t) p(x_{0:t-1}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \\
&\propto \int_{\boldsymbol{\theta}} p(y_{1:t-1}|x_{0:t-1}) p(x_{0:t-1, \boldsymbol{\theta}|x_t}) p(x_t) p(x_{0:t-1}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \\
&\propto \int_{\boldsymbol{\theta}} p(x_t|x_{0:t-1}, \boldsymbol{\theta}) p(x_{0:t-1}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \\
&\propto p(x_t|x_{0:t-1}); \quad by \quad \text{(A.5)}
\end{aligned}
\tag{A.32}
$$

At time instant t, when the new observation y_t becomes available, we update the prior (prediction) via Bayes rule,

$$
\begin{aligned}
(2)\quad p(x_t|x_{0:t-1}, y_{1:t}) &= \int_{\boldsymbol{\theta}} \underbrace{p(x_t, \boldsymbol{\theta}|x_{0:t-1}, y_{1:t})}_{joint \quad posterior} \mathrm{d}\boldsymbol{\theta}; \qquad \boldsymbol{\theta} = (\alpha, \beta, \sigma_u^2) \\
&= \int_{\boldsymbol{\theta}} \underbrace{p(x_t|\boldsymbol{\theta}, x_{0:t-1}, y_{1:t})}_{posterior} \underbrace{p(\boldsymbol{\theta}|x_{0:t-1}, y_{1:t})}_{posterior} \mathrm{d}\boldsymbol{\theta} \\
&\propto \int_{\boldsymbol{\theta}} p(x_t|x_{0:t-1}, y_{1:t}, \boldsymbol{\theta}) \underbrace{p(x_{0:t-1}, y_{1:t}|\boldsymbol{\theta})}_{likelihood} \underbrace{p(\boldsymbol{\theta})}_{prior} \mathrm{d}\boldsymbol{\theta} \\
&\propto \int_{\boldsymbol{\theta}} p(x_t|x_{0:t-1}, y_{1:t}, \boldsymbol{\theta}) p(x_{0:t-1}|y_{1:t}, \boldsymbol{\theta}) p(y_{1:t}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \\
&\propto \int_{\boldsymbol{\theta}} p(x_t|x_{0:t-1}, y_{1:t}, \boldsymbol{\theta}) p(x_{0:t-1}|\boldsymbol{\theta}) p(y_{1:t}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}; (cond. \quad indep) \\
&\propto \int_{\boldsymbol{\theta}} \underbrace{p(y_t|x_t)}_{likelihood} \underbrace{p(x_t|x_{0:t-1}, \boldsymbol{\theta})}_{prior} p(x_{0:t-1}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \\
&\propto p(y_t|x_t) \underbrace{\int_{\boldsymbol{\theta}} p(x_t|x_{0:t-1}, \boldsymbol{\theta}) p(x_{0:t-1}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}}_{=p(x_t|x_{0:t-1}) \quad by \quad \text{(A.5)}} \\
&\propto p(y_t|x_t) p(x_t|x_{0:t-1}). \qquad \text{(A.33)}
\end{aligned}
$$

Hence, the recursive weight update equation is derived as,

$$
\begin{aligned}
w_t^i &\propto w_{t-1}^i \frac{p(y_t|x_t^i) p(x_t^i|x_{0:t-1}^i)}{q(x_t^i|x_{0:t-1}^i, y_{1:t})} \\
&\propto w_{t-1}^i p(y_t|x_t^i) \qquad \text{(A.34)}
\end{aligned}
$$

where $q(x_t^i|x_{0:t-1}^i, y_{1:t}) = p(x_t|x_{0:t-1})$ is the importance sampling function.