Bayesian Monte Carlo Filtering for Stochastic Volatility Models

Roberto Casarin [†] [‡]

CEREMADE

University Paris IX (Dauphine)

and

Dept. of Economics

University Ca' Foscari, Venice

Abstract

Modelling of the financial variable evolution represents an important issue in financial econometrics. Stochastic dynamic models allow to describe more accurately many features of the financial variables, but often there exists a trade-off between the modelling accuracy and the complexity. Moreover the degree of complexity is increased by the use of latent factors, which are usually introduced in time series analysis, in order to capture the heterogeneous time evolution of the observed process. The presence of unobserved components makes the maximum likelihood inference more difficult to apply. Thus the Bayesian approach is preferable since it allows to treat general state space models and makes easier the simulation based approach to parameters estimation and latent factors filtering. The main aim of this work is to produce an updated review of Bayesian inference approaches for latent factor models. Moreover, we provide a review of simulation based filtering methods in a Bayesian perspective focusing, through some examples, on stochastic volatility models.

KEYWORDS: Monte Carlo Filtering, Particle Filter, Gibbs Sampling, Stochastic Volatility Models.

1 Introduction

The analysis of dynamic phenomena, which evolve over time is a common problem to many fields like engineering, physics, biology, statistics, economics and finance. A time varying system can be represented through a *dynamic model*, which is constituted by an observable component and an unobservable internal state. The hidden state vector represents the desired information that we want to extrapolate from the observations.

[†] I'm thankful M. Billio, C.P. Robert and D. Sartore for all the useful suggestions.

[‡] Address: CEREMADE, University Paris IX (Dauphine), 16 place du Maréchal de Lattre de Tassigny, 75775 Paris Cédex 16. E-mail: `casarin@ceremade.dauphine.fr`

Several kinds of dynamic models have been proposed in the literature for time series analysis and many approaches have been used for the estimation of these models. The seminal work of Kalman [22] and Kalman and Bucy [23] introduces filtering techniques (Kalman-Bucy filter) for continuous valued, linear and Gaussian dynamic systems. Another relevant work on dynamic model analysis is due to Maybeck [30], [31], [32]. He motivates the use of stochastic dynamic systems in engineering and examines filtering, smoothing and estimation problems for continuous state space models, in both a continuous and a discrete time framework. Moreover Harvey [20] extensively studies state space representation of dynamic models for time series analysis and treats the use of Kalman filter for states and parameters estimation, in continuous state space setting. Hamilton [19] analyzes several kinds of time series models and in particular introduces a filter (Hamilton-Kitagawa filter) for discrete time and discrete valued dynamic system. This filter can be used for dynamic models with a finite number of state values.

Bauwens, Lubrano and Richard [2] compare maximum likelihood inference with Bayesian inference on static and dynamic econometric models. Harrison and West [21] treat the problem of the dynamic model estimation in a Bayesian perspective. They give standard filtering and smoothing equations for Gaussian linear models and investigate the estimation problem for conditionally Gaussian linear models and for general nonlinear and non-Gaussian models. They review some Markov Chain Monte Carlo (MCMC) simulation techniques for filtering and smoothing the hidden states and for estimating parameters. Moreover, also the problem of processing data sequentially has been examined through the use of the adaptive importance sampling algorithm. Kim and Nelson [24] analyze Monte Carlo simulation methods for non-linear discrete valued model (switching regimes models). Recently, Durbin and Koopman [15] propose an updated review on Markov Chain Monte Carlo methods for estimation of general dynamic models, with both a Bayesian and a maximum likelihood approach.

Sequential simulation methods for filtering and smoothing in general dynamic models have been recently developed to overcome some problems of the traditional MCMC methods. As pointed out by Liu and Chen [26], Gibbs sampler is less attractive when we consider on-line data processing. Furthermore Gibbs sampler may be inefficient when simulated states are very sticky and the sampler has difficulties to move in the state space. In these situations, the use of sequential Monte Carlo techniques and in particular of particle filter algorithms may result more efficient. Doucet, Freitas and Gordon [13] provide the state of the art on sequential Monte Carlo methods. They discuss both applications and theoretical convergence results for these algorithms, with special attention to particle filters.

The main aim of this work is to introduce to Bayesian dynamic modelling and to the related inference problems of parameters estimation and hidden states filtering, with application to some basic discrete time stochastic volatility models. The second aim of the work is to review and

discuss drawbacks of most important simulation based inference approaches and to show how traditional simulation methods, like the single-move Gibbs sampler and the multi-move Gibbs sampler apply to a dynamic model. Finally we introduces recently developed sequential Monte Carlo simulation methods, focusing on Particle Filters. Throughout the work, many examples illustrate how Bayesian simulation based inference apply to basic Stochastic Volatility models.

The work is structured as follow. Section 2 introduces the general representation of a dynamic model in a Bayesian framework and deals with continuous stochastic volatility model (SV) and with Markov switching SV model which do not admit analytical filtering and smoothing densities. Section 3 reviews simulation based methods for inference. In particular the section reviews MCMC methods, presents an adaptive importance sampling algorithm and discusses particle filter algorithms. Section 4 concludes.

2 Bayesian Dynamic Models

In the following we give a quite general formulation of a probabilistic *dynamic model* and we obtain some fundamental relations for Bayesian inference on it. This definition of dynamic model would be general enough to include time series models analyzed in Kalman [22], Hamilton [19], Harrison and West [21] and in Doucet, Freitas and Gordon [13]. Throughout this work, we use a notation similar to that one commonly used in particle filter literature (see Doucet, Freitas and Gordon [13]).

We denote by $\{\mathbf{x}_t; t \in \mathbb{N}\}$, $\mathbf{x}_t \in \mathcal{X}$, the hidden state vectors of the system, by $\{\mathbf{y}_t; t \in \mathbb{N}_0\}$, $\mathbf{y}_t \in \mathcal{Y}$, the observable variables and by $\theta \in \Theta$ the parameter vector of the model. We assume that state space, observation space and parameter space respectively are $\mathcal{X} \subset \mathbb{R}^{n_x}$, $\mathcal{Y} \subset \mathbb{R}^{n_y}$ and $\Theta \subset \mathbb{R}^{n_\theta}$. n_x, n_y and n_θ represent the dimensions of the state vector, of the observable variable and of the parameter vector respectively.

The main advantage in using the general Bayesian state space representation of a dynamic model, is that it accounts also for nonlinear and non-Gaussian models. The Bayesian state space representation is given by an *initial distribution* $p(\mathbf{x}_0|\theta)$, a *measurement density* $p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta)$ and a *transition density* $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1}, \theta)$. The dynamic model is

$$\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta) \tag{1}$$

$$p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1}, \theta) \tag{2}$$

$$x_0 \sim p(\mathbf{x}_0|\theta), \quad \text{with } t = 1, \dots, T \tag{3}$$

where $p(\mathbf{x}_0|\theta)$ can be interpreted as the prior distribution on the initial state of the system.

By $\mathbf{x}_{0:t} \triangleq (\mathbf{x}_0, \ldots, \mathbf{x}_t)$ and by $\mathbf{y}_{1:t} \triangleq (\mathbf{y}_1, \ldots, \mathbf{y}_t)$ we denote respectively the collection of state vectors and of observable vectors, up to time t. We denote by $\mathbf{x}_{-t} \triangleq (\mathbf{x}_0, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_T)$ the collection of all the state vectors without the t-th element. The same notation is used also for the observable variable and for the parameter vectors.

If the transition density depends on the past, only through the last value of the hidden state vector, the dynamic model is defined first-order *Markovian*. In this case the system becomes

$$(\mathbf{y}_t | \mathbf{x}_t) \sim p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta) \tag{4}$$

$$(\mathbf{x}_t | \mathbf{x}_{t-1}) \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}, \theta) \tag{5}$$

$$x_0 \sim p(\mathbf{x}_0 | \theta), \quad \text{with } t = 1, \ldots, T. \tag{6}$$

Assuming that the first-order Markov property holds is not restrictive because a Markov model of order p can always be rewritten as a first-order Markovian model.

The stochastic volatility model given in Example 1, is a continuous states and discrete time dynamic model, which are now widely used in finance.

Example 1 - *(Stochastic Volatility Models)*
Two of the main features of the financial time series are time varying volatility and clustering phenomena in volatility. Thus stochastic volatility models have been introduced, in order to account for these features. An example of stochastic log-volatility model is

$$y_t \sim \mathcal{N}\left(0, e^{h_t}\right) \tag{7}$$

$$h_t \sim \mathcal{N}\left(\nu + \phi h_{t-1}, \sigma^2\right) \tag{8}$$

$$h_0 \sim \mathcal{N}\left(0, \sigma^2\right), \quad \text{with } t = 1, \ldots, T \tag{9}$$

where y_t is the observable process, with time varying volatility and h_t represents the stochastic log-volatility process. In Fig. 1 we exhibit a simulated path of the observable process y_t and of the stochastic volatility process h_t.

$\square$

In the following we discuss the three main issues which arise when making inference on a dynamic model, i.e.: filtering, predicting and smoothing. We present general solutions to these problems, but note that, without further assumptions on the densities, which characterize the dynamic model, these solutions are not yet analytical.
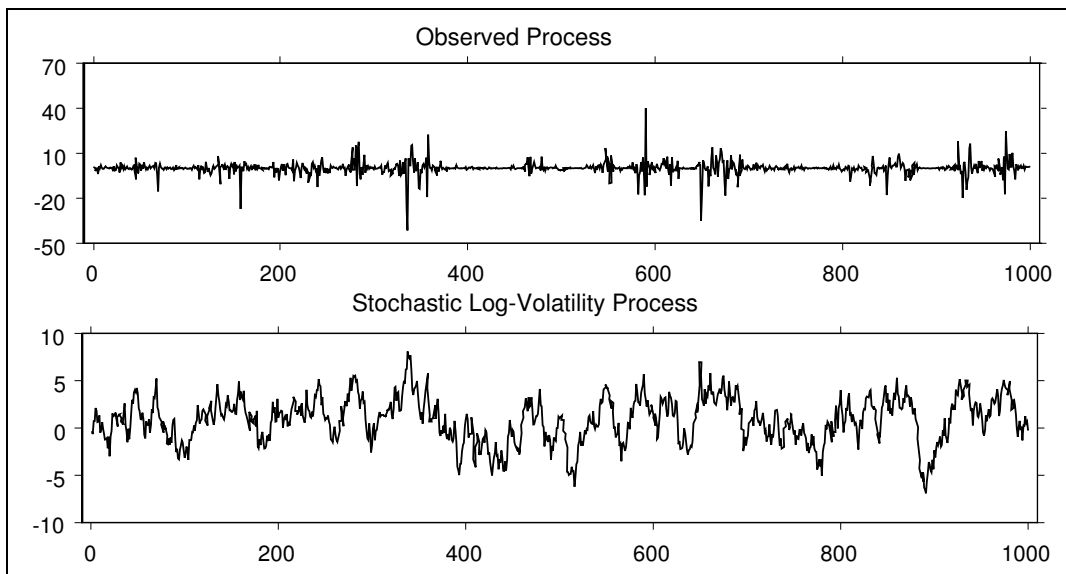
Figure 1: Simulated paths of 1,000 observations from the observable process y_t, with time varying volatility and of the stochastic volatility process h_t. We simulate the model given in Example 1 by setting $\phi = 0.9$, $\nu = 0.1$ and $\sigma^2 = 1$.

2.1 State Estimation

We treat here the problem of estimation of the hidden state vector when parameters are known. We are interested in estimating the density $p(\mathbf{x}_t|\mathbf{y}_{1:s}, \theta)$. If $t = s$ the density of interest is called *filtering density*, if $t < s$ it is called *smoothing density*, finally if $t > s$ it is called *prediction density*.

For the dynamic model given in equations (4), (5) and (6) we assume that at time t the density $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}, \theta)$ is known. Observe that if $t = 1$ the density $p(\mathbf{x}_0|\mathbf{y}_0, \theta) = p(\mathbf{x}_0|\theta)$ is the initial distribution of the dynamic model.

Applying the Chapman-Kolmogorov transition density, we obtain the one step ahead *prediction density*

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \theta) = \int_{\mathcal{X}} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}, \theta) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}, \theta) d\mathbf{x}_{t-1} \tag{10}$$

As the new observation $\mathbf{y}_t$ becomes available, it is possible, using the Bayes theorem, to update the prediction density and to filter the current state of the system. The *filtering density* is

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta) = \frac{p(\mathbf{y}_t, \mathbf{x}_t|\mathbf{y}_{1:t-1}, \theta)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \theta)} = \frac{p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta) p(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \theta)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \theta)} \tag{11}$$

5

where $p(\mathbf{x}_t|\mathbf{y}_{1:t-1},\theta)$ is the prediction density determined at the previous step and the density at the denominator is the marginal of the current state and observable variable joint density

$$p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\theta) = \int_{\mathcal{X}} p(\mathbf{y}_t,\mathbf{x}_t|\mathbf{y}_{1:t-1},\theta)d\mathbf{x}_t = \int_{\mathcal{X}} p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{y}_{1:t-1},\theta)d\mathbf{x}_t. \quad (12)$$

At each date t, it is possible to determine the K-steps-ahead prediction density, conditional on the available information $\mathbf{y}_{1:t}$. Given the dynamic model described by equations (4), (5) and (6), the *K-steps-ahead prediction density* of the state vector $\mathbf{x}_t$ can be evaluated iteratively. The first step is

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t},\theta) = \int_{\mathcal{X}} p(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{y}_{1:t},\theta)p(\mathbf{x}_t|\mathbf{y}_{1:t},\theta)d\mathbf{x}_t \quad (13)$$

and the k-th step, with $k = 1,\ldots,K$, is

$$p(\mathbf{x}_{t+k}|\mathbf{y}_{1:t},\theta) = \int_{\mathcal{X}} p(\mathbf{x}_{t+k}|\mathbf{x}_{t+k-1},\mathbf{y}_{1:t},\theta)p(\mathbf{x}_{t+k-1}|\mathbf{y}_{1:t},\theta)d\mathbf{x}_{t+k-1} \quad (14)$$

where

$$p(\mathbf{x}_{t+k}|\mathbf{x}_{t+k-1},\mathbf{y}_{1:t},\theta) = \int_{\mathcal{Y}^{k-1}} p(\mathbf{x}_{t+k}|\mathbf{x}_{t+k-1},\mathbf{y}_{1:t+k-1},\theta)p(d\mathbf{y}_{t+1:t+k-1}|\mathbf{y}_{1:t},\theta) \quad (15)$$

where $\mathcal{Y}^k = \bigotimes_{i=1}^{k} \mathcal{Y}_i$ is the k-times product of the state space. Similarly, the *K-steps-ahead prediction density* of the observable variable $\mathbf{y}_{t+K}$ conditional on the information available at time t is determined as follow

$$p(\mathbf{y}_{t+K}|\mathbf{y}_{1:t},\theta) = \int_{\mathcal{Y}} p(\mathbf{y}_{t+K}|\mathbf{x}_{t+K},\mathbf{y}_{1:t+K-1},\theta)p(d\mathbf{y}_{t+1:t+K-1}|\mathbf{y}_{1:t},\theta)p(d\mathbf{x}_{t+K}|\mathbf{y}_{1:t},\theta) \quad (16)$$

Due to the high number of integrals that must be solved, previous densities may be difficult to evaluate with general dynamics. From a numerical point of view simulation methods, like MCMC algorithms or Particle Filters allow us to overcome these difficulties; while from an analytical point of view to obtain simpler relations we need to introduce some simplifying hypothesis on the dynamics of the model. For example if we assume that the evolution of the dynamic model does not depend on the past values of the observable variable $\mathbf{y}_{1:t}$, then equations (4), (5) and (6) become

$$(\mathbf{y}_t|\mathbf{x}_t) \sim p(\mathbf{y}_t|\mathbf{x}_t,\theta) \quad (17)$$
$$(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim p(\mathbf{x}_t|\mathbf{x}_{t-1},\theta) \quad (18)$$
$$x_0 \sim p(\mathbf{x}_0|\theta), \quad \text{with } t = 1,\ldots,T. \quad (19)$$
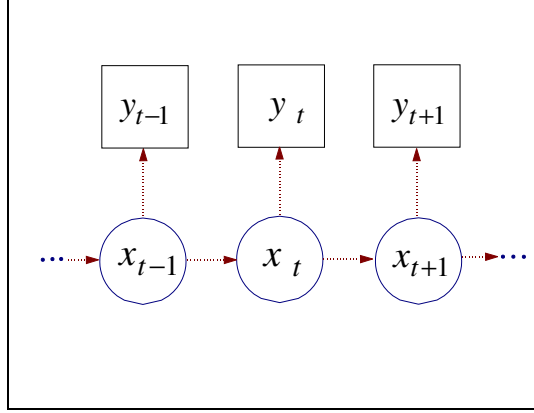
Figure 2: The causality structure of a Markov dynamic model with hidden states is represented by means of a *Directed Acyclic Diagram* (DAG). A box around the variable indicates the variable is known, while a circle indicates a hidden variable.

The causality structure of this model has been represented through the Directed Acyclic Graph (DAG) exhibited in Fig. 2. Under previous assumptions the filtering and prediction densities simplify as follows

$$
p(\mathbf{x}_t|\mathbf{y}_{1:t-1},\theta) \;=\; \int_{\mathcal{X}} p(\mathbf{x}_t|\mathbf{x}_{t-1},\theta)p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1},\theta)d\mathbf{x}_{t-1} \tag{20}
$$

$$
p(\mathbf{x}_t|\mathbf{y}_{1:t},\theta) \;=\; \frac{p(\mathbf{y}_t|\mathbf{x}_t,\theta)p(\mathbf{x}_t|\mathbf{y}_{1:t-1},\theta)}{p(\mathbf{y}_t|\mathbf{y}_{0:t-1},\theta)} \tag{21}
$$

$$
p(\mathbf{x}_{t+K}|\mathbf{y}_{1:t},\theta) \;=\; \int_{\mathcal{X}} p(\mathbf{x}_{t+K}|\mathbf{x}_{t+K-1},\theta)p(\mathbf{x}_{t+K-1}|\mathbf{y}_{1:t},\theta)d\mathbf{x}_{t+K-1} \tag{22}
$$

$$
p(\mathbf{y}_{t+K}|\mathbf{y}_{1:t},\theta) \;=\; \int_{\mathcal{X}} p(\mathbf{y}_{t+K}|\mathbf{x}_{t+K},\theta)p(\mathbf{x}_{t+K}|\mathbf{y}_{1:t},\theta)d\mathbf{x}_{t+K}. \tag{23}
$$

We conclude this section with two important recursive relations. Both these relations can be proved starting from the definition of joint smoothing density and assuming that the Markov property holds.

The first relation is the sequential filtering equation, which reveals particularly useful when processing data sequentially and becomes fundamental in implementing Particle Filter algorithms. Consider the joint posterior density of the state vectors, conditional on the information available at time T

$$
\begin{aligned}
p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T},\theta) &= \frac{p(\mathbf{x}_{0:T},\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)} = \\
&= p(\mathbf{x}_{0:T-1}|\mathbf{y}_{1:T-1},\theta)\frac{p(\mathbf{x}_T,\mathbf{y}_T|\mathbf{x}_{0:T-1},\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)} = \\
&= p(\mathbf{x}_{0:T-1}|\mathbf{y}_{1:T-1},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_{0:T},\mathbf{y}_{1:T-1},\theta)p(\mathbf{x}_T|\mathbf{x}_{0:T-1},\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)} = \quad (24)\\
&= p(\mathbf{x}_{0:T-1}|\mathbf{y}_{1:T-1},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)p(\mathbf{x}_T|\mathbf{x}_{T-1},\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)} = \\
&= p(\mathbf{x}_{0:T-1}|\mathbf{y}_{1:T-1},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_T,\theta)p(\mathbf{x}_T|\mathbf{x}_{T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)}.
\end{aligned}
$$

where the last line is due to the Markov property of the measurement and transition densities.

Consider again the joint posterior density of the state vectors, $\mathbf{x}_{0:T}$, given the information, $\mathbf{y}_{1:T}$, available at time T

$$
\begin{aligned}
&p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T},\theta) = \\
&= p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{0:T-1}|\mathbf{x}_T,\mathbf{y}_{1:T},\theta) = \\
&= p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{0:T-2}|\mathbf{x}_{T-1:T},\mathbf{y}_{1:T},\theta) = \\
&\overset{Bayes}{=} p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_{T-1:T},\mathbf{y}_{1:T-1},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)}p(\mathbf{x}_{0:T-2}|\mathbf{x}_{T-1:T},\mathbf{y}_{1:T},\theta) = \\
&\overset{Markov}{=} p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)}p(\mathbf{x}_{0:T-2}|\mathbf{x}_{T-1:T},\mathbf{y}_{1:T},\theta) = \quad (25)\\
&= p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)p(\mathbf{x}_{0:T-2}|\mathbf{x}_{T-1},\mathbf{y}_{1:T},\theta).
\end{aligned}
$$

By applying iteratively Bayes theorem and Markov property of the dynamic model we obtain the second recursive relation, which provides a factorization of the state vector *smoothing density*

$$
\begin{aligned}
&p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T},\theta) = \\
&= p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)\prod_{t=0}^{T-2}p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:T},\theta) = \\
&= p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)\prod_{t=0}^{T-2}p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\mathbf{y}_{t+1:T},\theta) = \quad (26)\\
&\overset{Bayes}{=} p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)\prod_{t=0}^{T-2}\frac{p(\mathbf{y}_{t+1:T}|\mathbf{x}_{t:t+1},\mathbf{y}_{1:t},\theta)p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)}{p(\mathbf{y}_{t+1:T}|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)} =
\end{aligned}
$$

$$\overset{Markov}{=} p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)\prod_{t=0}^{T-2}\frac{p(\mathbf{y}_{t+1:T}|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)}{p(\mathbf{y}_{t+1:T}|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)} =$$

$$= p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)\prod_{t=0}^{T-1}p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta).$$

The way this relation is obtained differs from that one given in Carter and Köhn [7] because we consider the general model in equations (4)-(6), with transition and measurement densities depending on past values of $\mathbf{y}_t$. Note that the density $p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)$, which appears in the joint smoothing density, can be represented through the filtering and the prediction densities

$$p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta) = \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{y}_{1:t},\theta)p(\mathbf{x}_t|\mathbf{y}_{1:t},\theta)}{p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t},\theta)}. \tag{28}$$

This factorization of the smoothing density is also relevant when inference is carried out through simulation methods. See for example the multi-move Gibbs sampler of Carter and Köhn [7] and the particle filter algorithms.

Only in some well known cases, these filtering densities admit an analytical form. For the normal linear dynamic model, filtering and smoothing density are given by the Kalman filter. See Harrison and West [21] for a Bayesian analysis of the Kalman filter. See Harvey [20] for a frequentist approach to the Kalman filter.

A class of models which do not admit a tractable analytical representation of the filtering, prediction and smoothing densities is the class of *conditional normal linear models* (also called *multi-process* models, in Harrison and West [21]), which are widely used in economic analysis (see Kim and Nelson [24]). In the next section, we introduce another example of nonlinear dynamic models with jumps, which are used in financial applications.

2.2 Stochastic Volatility Models with Jumps

Using a Bayesian representation, stochastic volatility model with jumps can be defined as follow

$$(y_t|h_t) \sim \mathcal{N}(0, e^{h_t}) \tag{29}$$

$$(h_t|h_{t-1},s_t) \sim \mathcal{N}(h_t|\alpha_{s_t} + \phi\, h_{t-1}, \sigma^2) \tag{30}$$

for $t = 1,\ldots,T$, where s_t is the jump process. If s_t is a discrete time and finite state Markov chain with transition probabilities

$$\mathbb{P}(s_t = j|s_{t-1} = i, s_{t-2} = i_2..., s_0 = i_t) = \mathbb{P}(s_t = j|s_{t-1} = i) = p_{ij} \tag{31}$$
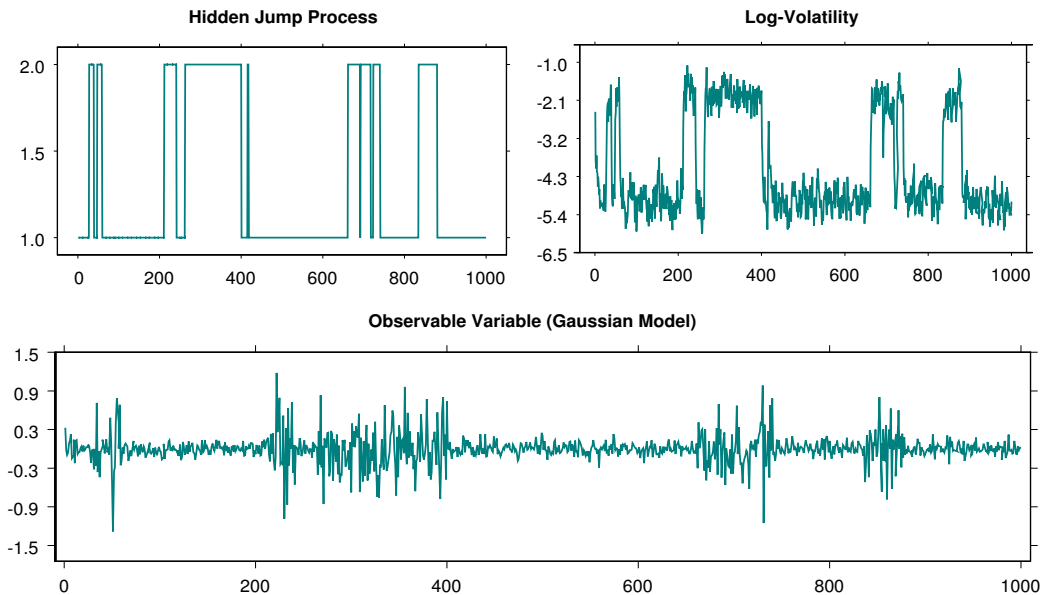
9

Figure 3: Simulation of the Markov switching stochastic volatility model $\mathcal{M}_1$ ($\alpha_1 = -2.5$, $\alpha_2 = -1$, $\phi = 0.5$, $\sigma^2 = 0.1$, $p_{11} = 0.99$, $p_{22} = 0.975$). The left upper graph exhibits the evolution of the hidden jump process, the right upper graph shows the log-volatility of the observable process, which is represented in the third graph.

$i, j = 1, \ldots, L$, L denoting the number of unobservable states, then the model is called *Markov switching*. Fig. 3 exhibits simulation paths of $1,000$ observations of the Markov switching process, the latent factor and the observable variable, respectively. In order to simulate the MS stochastic volatility model we use parameters estimated in So, Lam and Li [40].

3 Simulation Based Filtering

The main aim of this section is to review both some traditional and recently developed inference methods for nonlinear and non-Gaussian models. We focus on the Bayesian approach and on simulation based methods. First Markov Chain Monte Carlo methods are reviewed with particular attention to the single-move Gibbs sampler due to Carlin, Polson and Stoffer [6] and to the multi-move Gibbs sampler due to Carter and Köhn [7] and Frühwirth-Schnatter [16]. Moreover some basic sequential Monte Carlo simulation methods are introduced. We examine the adaptive importance sampling algorithm due to West [44], [45], the sequential importance sampling algorithm and more advanced sequential Monte Carlo algorithms called Particle Filters (see Doucet, Freitas and Gordon [13]).

10

Finally, note that another important issue in making inference on dynamic models is the estimation of the parameter vector. In a Bayesian MCMC based approach, parameters and hidden states are jointly estimated by simulating from the posterior distribution of the model. Also in a sequential importance sampling approach and following the engineering literature, a common way to solve the parameter estimation problem is to treat parameters θ as hidden state of the system (see Berzuini et al. [3]). The model is restated assuming time dependent parameter vectors θ_t, and imposing a constraint on the evolution: $\theta_t = \theta_{t-1}$. The constraint can be expressed also in terms of transition probability as follows

$$\theta_t \sim \delta_{\theta_{t-1}}(\theta_t), \qquad t = 0, \ldots, T \tag{32}$$

where $\delta_{x^*}(x)$ denotes the Dirac delta function. The Bayesian model is then completed by assuming a prior distribution $p(\theta_0)$ on the parameter vector.

3.1 The Gibbs Sampler

In previous sections we examine some estimation algorithms for filtering, predicting and smoothing the state vector of a quite general probabilistic dynamic model. In order to examine Gibbs sampling methods, we consider the following dynamic model

$$(\mathbf{y}_t|\mathbf{x}_t) \sim p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta) \tag{33}$$

$$(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}, \theta) \tag{34}$$

$$x_0 \sim p(\mathbf{x}_0|\theta) \tag{35}$$

$$\theta \sim p(\theta), \quad \text{with } t = 1, \ldots, T. \tag{36}$$

The estimation problem is solved in a Bayesian perspective by evaluating the mean of the joint posterior density of the state and parameter vectors $p(\mathbf{x}_{0:T}, \theta|\mathbf{y}_{1:T})$. Tanner and Wong [42] motivates this solution by the *data augmentation principle*, which consists in considering the hidden state vectors as nuisance parameters.

If an analytical evaluation of the posterior mean is not possible, then simulation methods and in particular Monte Carlo Markov Chain apply. The most simple solution is to implement a *single-move Gibbs sampler* (see Carlin, Polson and Stoffer [6] and Harrison and West [21]). This method generates the states one at a time using the Markov property of the dynamic model to condition on the neighboring states. However the first order Markov dependence between adjacent states induces a high correlation between outputs of the Gibbs sampler and causes a slow convergence of the algorithm. To solve this problem Carter and Köhn [7] and Frühwirth-Schnatter [16]

simultaneously proposes *multi-move Gibbs sampler*. The main idea of this method is to generate simultaneously all the state vectors using analytical filtering and smoothing relations.

Their approach is less general than that of Carlin, Polson and Stoffer [6], but for linear dynamic models with Gaussian mixture innovations in the observation equation, their approach is more efficient. In particular the multi-move Gibbs sampler has a faster convergence to the posterior distribution and the posterior moment estimates have smaller variance. These results are supported theoretically by Liu, Wong and Kong [28], [29] and Müller [33], who shows that generating variables simultaneously produces faster convergence than generating them one at a time.

The idea of grouping parameters (or hidden states) when simulating is now commonly in Bayesian inference on stochastic models, with latent factors. For example, multi-move MCMC algorithms have been used for stochastic volatility models by Kim, Shephard and Chib [25] and extended to generalized stochastic volatility models by Chib, Nardari and Shephard [10]. Shephard [38] and Shephard and Pitt [39] discussed multi-move MCMC algorithms to non-Gaussian time series models . Finally, an alternative way of simulating from the smoothing density of the state vectors is discussed in De Jong and Shephard [12].

In the following we briefly present how to implement the single-move Gibbs sampler for parameters and states estimation. On the time interval $\{1, \ldots, T\}$, the conditional posterior distributions of the parameter vector and of the state vectors are

$$p(\theta|\mathbf{x}_{0:T}, \mathbf{y}_{1:T}) \propto p(\theta)p(\mathbf{x}_0|\theta) \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta)p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}, \theta) \qquad (37)$$

$$p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}, \theta) \propto p(\mathbf{x}_0|\theta) \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta)p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}, \theta). \qquad (38)$$

A basic Gibbs sampler is obtained by simulating sequentially from the parameter vector posterior (*parameter simulation step*) in equation (37) and from the state vectors posterior (*data augmentation step*) in equation (38) conditionally on the parameter vector simulated at the previous step. When conditional distributions cannot be directly simulated, the corresponding steps in the Gibbs algorithm can be replaced by Metropolis-Hastings steps. The resulting algorithms are called hybrid sampling algorithms and they are validated in Tierney [43]. A generic Gibbs sampler can be used for simulating the posterior of the parameter vector conditional on the simulated state vectors. The single-move Gibbs sampler for the state vectors, conditional on the simulated parameter vector, is then obtained by drawing each state vector conditionally on the other simulated state vectors.

Single-move algorithm can be implemented for general dynamic models. Moreover, note that the dynamic model given in equations (33)-(36) satisfies to the Markov property. For the general

model described in Equations (33)-(36), with measurement and transition densities depending on past values of the observable variable $\mathbf{y}_{1:t-1}$, the full posterior density of the state vector, given in the single-move Gibbs sampler (see the Algorithm 2) can be derived as follow.

Algorithm 1 - *Gibbs sampler for the parameter vector*

Given simulated vectors $\theta^{(i)}$ and $\mathbf{x}_{0:T}^{(i)}$, generate the parameter vector

1. $\theta_1^{(i+1)} \sim p(\theta_1 | \theta_2^{(i)}, \ldots, \theta_{n_\theta}^{(i)}, \mathbf{x}_{0:T}^{(i)}, \mathbf{y}_{1:T})$

2. $\theta_2^{(i+1)} \sim p(\theta_2 | \theta_1^{(i+1)}, \theta_3^{(i)}, \ldots, \theta_{n_\theta}^{(i)}, \mathbf{x}_{0:T}^{(i)}, \mathbf{y}_{1:T})$

3. ...

4. $\theta_k^{(i+1)} \sim p(\theta_k | \theta_1^{(i+1)}, \ldots, \theta_{k-1}^{(i+1)}, \theta_{k+1}^{(i)}, \ldots, \theta_{n_\theta}^{(i)}, \mathbf{x}_{0:t}^{(i)}, \mathbf{y}_{1:T})$

5. ...

6. $\theta_{n_\theta}^{(i+1)} \sim p(\theta_{n_\theta} | \theta_1^{(i+1)}, \ldots, \theta_{n_\theta-1}^{(i+1)}, \mathbf{x}_{0:T}^{(i)}, \mathbf{y}_{1:T})$

Algorithm 2 - *Single-Move Gibbs Sampler* -

(i) Simulate $\theta^{(i)}$ through Algorithm 1;
(ii) Given $\theta^{(i)}$ and $\mathbf{x}_{0:T}^{(i)}$, simulate state vectors

7. $\mathbf{x}_0^{(i+1)} \sim p(\mathbf{x}_0 | \mathbf{x}_{2:T}^{(i)}, \mathbf{y}_{1:T}, \theta^{(i+1)})$

8. $\mathbf{x}_1^{(i+1)} \sim p(\mathbf{x}_1 | \mathbf{x}_0^{(i+1)}, \mathbf{x}_{2:T}^{(i)}, \mathbf{y}_{1:T}, \theta^{(i+1)})$

9. ...

10. $\mathbf{x}_t^{(i+1)} \sim p(\mathbf{x}_t | \mathbf{x}_{0:t-1}^{(i+1)}, \mathbf{x}_{t+1:T}^{(i)}, \mathbf{y}_{1:T}, \theta^{(i+1)})$

11. ...

12. $\mathbf{x}_T^{(i+1)} \sim p(\mathbf{x}_T | \mathbf{x}_{0:T-1}^{(i+1)}, \mathbf{y}_{1:T}, \theta^{(i+1)})$

Consider the full posterior density of the t-th state vector and apply the independence assumption between $\mathbf{y}_{t+1:T}$ and $\mathbf{x}_t$

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

then the last density simplifies as follow

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{x}_{-t},\mathbf{y}_{1:t},\theta) &= p(\mathbf{x}_t|\mathbf{x}_{0:t-1},\mathbf{x}_{t+1:T},\mathbf{y}_{1:t},\theta) = \\
&= \frac{p(\mathbf{x}_{t:T},\mathbf{y}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t-1},\theta)}{p(\mathbf{x}_{t+1:T},\mathbf{y}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t-1},\theta)} = \\
&= \frac{p(\mathbf{x}_{t+1:T},\mathbf{y}_t|\mathbf{x}_{0:t},\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t-1},\theta)}{p(\mathbf{x}_{t+1:T},\mathbf{y}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t-1},\theta)} = \\
&= \frac{p(\mathbf{x}_{t+1:T}|\mathbf{x}_{0:t},\mathbf{y}_{1:t},\theta)p(\mathbf{y}_t|\mathbf{x}_{0:t},\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t-1},\theta)}{p(\mathbf{x}_{t+1:T},\mathbf{y}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t-1},\theta)} = \\
&\overset{Markov}{=} \frac{p(\mathbf{x}_{t+1:T}|\mathbf{x}_{0:t},\mathbf{y}_{1:t},\theta)p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{1:t-1},\theta)}{p(\mathbf{x}_{t+1:T},\mathbf{y}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t-1},\theta)}.
\end{aligned}
$$

The full posterior density of the t-th state vector is thus proportional to

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{x}_{-t},\mathbf{y}_{1:t},\theta) &\propto \\
\propto\;& p(\mathbf{x}_{t+1:T}|\mathbf{x}_{0:t},\mathbf{y}_{1:t},\theta)p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{1:t-1},\theta) = \\
=\;& p(\mathbf{x}_{t+2:T}|\mathbf{x}_{0:t+1},\mathbf{y}_{1:t},\theta)p(\mathbf{x}_{t+1}|\mathbf{x}_{0:t},\mathbf{y}_{1:t},\theta)p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{1:t-1},\theta) = \\
\overset{Markov}{=}\;& p(\mathbf{x}_{t+2:T}|\mathbf{x}_{0:t+1},\mathbf{y}_{1:t},\theta)p(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{y}_{1:t},\theta)p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{1:t-1},\theta) \propto \\
\propto\;& p(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{y}_{1:t},\theta)p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{1:t-1},\theta).
\end{aligned}
$$

and due to this property of the posterior density, the implementation of the single-move Gibbs sampling algorithm becomes easier.

Example 2 - *(Stochastic Volatility Model, Example 1 continued)*
We illustrate, through this example, how simulation based Bayesian inference applies to dynamic models. In particular, we develop a single-move Gibbs sampler for the stochastic volatility model given in Example 1. Note that the stochastic volatility model can be rewritten as follows

$$y_t = e^{h_t/2}\varepsilon_t, \qquad\qquad \varepsilon_t \sim \mathcal{N}(0,1) \tag{39}$$

$$h_t = \nu + \phi h_{t-1} + \sigma\eta_t, \quad \eta_t \sim \mathcal{N}(0,1) \tag{40}$$

$$\text{with} \quad t = 1,\ldots,T. \tag{41}$$

The completed likelihood function of this model is

$$
\begin{aligned}
&L(\phi,\nu,\sigma^2|y_{1:T},h_{0:T}) = \\
&= \prod_{t=1}^{T} \frac{1}{(2\pi)^{1/2}e^{h_t/2}} \exp\left\{-\frac{y_t^2 e^{-h_t}}{2}\right\} \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{(h_t - \nu - \phi h_{t-1})^2}{2\sigma^2}\right\} = \\
&= \frac{1}{(2\pi)^T(\sigma^2)^{\frac{T}{2}}} \exp\left\{-\frac{1}{2}\sum_{t=1}^{T}(y_t^2 e^{-h_t} + h_t) - \frac{1}{2\sigma^2}\sum_{t=1}^{T}(h_t - \nu - \phi h_{t-1})^2\right\}
\end{aligned}
\tag{42}
$$

In order to conclude the description of the Bayesian dynamic model we take the following prior distributions on the parameters ϕ, ν, σ^2 and h_0

$$\phi \sim \mathcal{N}(a,b^2)\mathbb{I}_{(-1,+1)}(\phi) \tag{43}$$

$$\nu \sim \mathcal{N}(c,d^2) \tag{44}$$

$$\sigma^2 \sim \mathcal{IG}(\alpha,\beta) \tag{45}$$

$$h_0 \sim \mathcal{N}(0,\sigma^2) \tag{46}$$

where $\mathcal{IG}(\cdot,\cdot)$ indicates the *Inverse Gamma* distribution. Note that for the parameter ϕ we use a truncated Normal distribution, but a Uniform distribution $\mathcal{U}_{[-1,1]}$ could alternatively be used. In the first part of the single-move Gibbs sampler, the parameters are simulated from the posterior distributions, through the following three steps
(i) Simulate $\phi \sim \pi(\phi|\sigma^2,\nu,y_{1:T},h_{0:T})$, where

$$\pi(\phi|\sigma^2,\nu,y_{1:T},h_{0:T}) \propto \tag{47}$$

$$\propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{t=1}^{T}(h_t - \nu - \phi h_{t-1})^2\right\} \frac{1}{(2\pi)^{1/2}b} \exp\left\{-\frac{(\phi - a)^2}{2b^2}\right\}\mathbb{I}_{(-1,+1)}(\phi) \propto$$

$$\propto \exp\left\{-\frac{1}{2}\left[\phi^2\left(\frac{1}{b^2} + \frac{\sum_{t=1}^{T} h_{t-1}^2}{\sigma^2}\right) - 2\phi\left(\frac{a}{b^2} + \frac{\sum_{t=1}^{T} h_{t-1}(h_t - \nu)}{\sigma^2}\right)\right]\right\}\mathbb{I}_{(-1,+1)}$$

$$\propto \mathcal{N}(\tilde{a},\tilde{b}^2)\mathbb{I}_{(-1,+1)}(\phi)$$

with

$$\tilde{a} = \tilde{b}^2 \left(\frac{a}{b^2} + \frac{1}{\sigma^2} \sum_{t=1}^{T} h_{t-1}(h_t - \nu) \right), \quad \tilde{b}^2 = \left(\frac{1}{b^2} + \frac{1}{\sigma^2} \sum_{t=1}^{T} h_{t-1}^2 \right)^{-1}$$

(ii) Simulate $\nu \sim \pi(\nu|\phi, \sigma^2, y_{1:T}, h_{0:T})$, where

$$\pi(\nu|\phi, \sigma^2, y_{1:T}, h_{0:T}) \propto \tag{48}$$
$$\propto \exp\left\{ -\frac{1}{2\sigma^2} \sum_{t=1}^{T} (h_t - \nu - \phi h_{t-1})^2 \right\} \exp\left\{ -\frac{(\nu - c)^2}{2d^2} \right\} \frac{1}{(2\pi)^{1/2}d} \propto$$
$$\propto \exp\left\{ -\frac{1}{2} \left[\nu^2 (\frac{1}{d^2} + \frac{T}{\sigma^2}) - 2\nu \left(\frac{c}{d^2} + \frac{\sum_{t=1}^{T}(h_t - \phi h_{t-1})}{\sigma^2} \right) \right] \right\} \propto$$
$$\propto \mathcal{N}(\tilde{c}, \tilde{d}^2)$$

with

$$\tilde{c} = \tilde{d}^2 \left(\frac{c}{d^2} + \frac{1}{\sigma^2} \sum_{t=1}^{T} (h_t - \phi h_{t-1}) \right), \quad \tilde{d}^2 = \left(\frac{1}{d^2} + \frac{T}{\sigma^2} \right)^{-1}$$

(iii) Simulate $\sigma^2 \sim \pi(\sigma^2|\phi, \nu, y_{1:T}, h_{0:T})$, where

$$\pi(\sigma^2|\phi, \nu, y_{1:T}, h_{0:T}) \propto \tag{49}$$
$$\propto \left(\frac{1}{\sigma^2} \right)^{\frac{(T+1)}{2}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{t=1}^{T} (h_t - \nu - \phi h_{t-1})^2 - \frac{h_0^2}{2\sigma^2} \right\} \frac{\beta^\alpha e^{-\beta/\sigma^2}}{\Gamma(\alpha)(\sigma^2)^{\alpha+1}} \mathbb{I}_{[0,\infty)}(\sigma^2) \propto$$
$$\propto \exp\left\{ -\frac{1}{2\sigma^2} (2\beta + \sum_{t=1}^{T} (h_t - \nu - \phi h_{t-1})^2 + h_0^2) \right\} \left(\frac{1}{\sigma^2} \right)^{\alpha + \frac{T+1}{2} + 1} \mathbb{I}_{[0,\infty)}(\sigma^2) \propto$$
$$\propto \mathcal{IG}(\tilde{\alpha}, \tilde{\beta})$$

with

$$\tilde{\alpha} = \alpha + \frac{T+1}{2}, \qquad \tilde{\beta} = \beta + \frac{1}{2} \left[\sum_{t=1}^{T} (h_t - \nu - \phi h_{t-1})^2 + h_0^2 \right]$$

In the second part of the single-move Gibbs sampler, the hidden variables, $\{h_t\}_{t=1}^{T}$, are sampled once at time, from the following posterior distribution

16

$$\pi(h_t|h_{-t}, y_{1:T}, \nu, \phi, \sigma^2) \propto \exp\left\{-\frac{(y_t^2 e^{-h_t} + h_t)}{2}\right\}$$

$$\exp\left\{-\frac{(h_t - \nu - \phi h_{t-1})^2}{2\sigma^2} - \frac{(h_{t+1} - \nu - \phi h_t)^2}{2\sigma^2}\right\} \tag{50}$$

for $t = 1, \ldots, T$. Due to the presence of the quantity e^{-h_t} in the exponential any standard simulation method easily applies here. Moreover the posterior distribution is known up to a normalising constant, thus the M.-H. algorithm can be used. The proposal distribution can be obtained by replacing the function $\left(y_t^2 e^{-h_t} + h_t\right)$ in equation (50), with $(\log y_t^2 + 1) + (h_t - \log y_t^2)^2/2$, which is its the second order Taylor expansion, around $\log\left(y_t^2\right)$. We simulate h_t from the following proposal density

$$q(h_t|h_{-t}, y_{1:T}, \nu, \phi, \sigma^2) \propto \tag{51}$$

$$\propto \exp\left\{-\frac{(h_t - \log y_t^2)^2}{4} - \frac{(h_t - \nu - \phi h_{t-1})^2}{2\sigma^2} - \frac{(h_{t+1} - \nu - \phi h_t)^2}{2\sigma^2}\right\} \propto$$

$$\propto \exp\left\{-\frac{h_t^2}{2}\left(\frac{1+\phi^2}{\sigma^2} + \frac{1}{2}\right) + h_t\left(\frac{\nu(1-\phi) + \phi(h_{t-1} + h_{t+1})}{\sigma^2} + \frac{\log y_t^2}{2}\right)\right\} \propto$$

$$\propto \mathcal{N}(\mu, \tilde{\sigma}^2)$$

with

$$\mu = \left((\nu(1-\phi) + \phi(h_{t-1} + h_{t+1}))\,\sigma^{-2} + \log y_t^2/2\right)\tilde{\sigma}^2, \quad \tilde{\sigma}^2 = \left(\frac{1+\phi^2}{\sigma^2} + \frac{1}{2}\right)^{-1}$$

We apply the single-move Gibbs sampler to the synthetic dataset exhibited in Fig. 1. We use a burn-in sample of 10,000 observations to reduce the dependence on the initial conditions of the sampler and take the remaining 40,000 values to calculate the parameter estimates, which are represented in Table 1.

Table 1: Bayesian estimates of the parameters ϕ, ν and σ^2. We use a burn-in sample of 10,000 observations and calculate the parameter estimates by averaging 40,000 raw values simulated from the posterior distributions, by means of the Gibbs sampler.

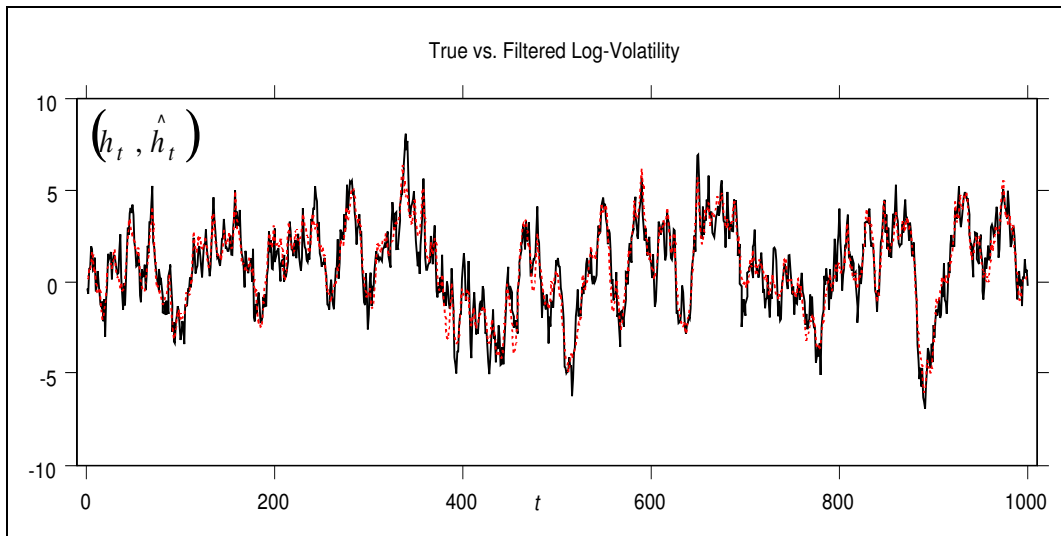| Parameters	True	Estimates	St. Dev.	2.5% percentile	97.5% percentile
ϕ	0.9	0.9096	0.0146	0.8813	0.9370
ν	0.1	0.0754	0.0350	0.0048	0.1382
σ^2	1	1.0075	0.0313	0.9487	1.0712

True vs. Filtered Log-Volatility

Figure 4: Graph of the true (solid line) and filtered (dotted line) densities. The filtered density has been obtained by averaging 50,000 values, simulated through the single-move Gibbs sampler, given in example 2.

Figure 4 shows the true stochastic log-volatility versus the filtered volatility. Figures 5 and 6 show the raw output of the single-move Gibbs sampler and the ergodic means respectively. Figures 7 and 8 show the autocorrelation of the simulated values and the histograms of the posterior distributions, respectively.

Simulations have been carried out on a PC with Intel 2400 MHz processor, using routines implemented in Gauss 3.2.8.

$\square$

Although the simplification due to the first order Markov property of the dynamic model makes the single-move Gibbs sampler easier to implement, some problems arise. In particular, the Markovian dependence between neighboring states generates correlation between outputs of the Gibbs sampler and origins slower convergence to the posterior distribution (see Carter and Köhn [7]). A consequence is that if an adaptive importance sampling is carried out by running parallel single-move Gibbs samplers, the number of replications before convergence of the parameter estimates is high.

A general method to solve this autocorrelation problem in the output of the Gibbs sampler is to group parameters (or states) and to simulate them simultaneously. This idea has been independently applied by Carter and Köhn [7] and by Frühwirth-Schnatter [16] to dynamic models and the resulting algorithm is the multi-move Gibbs sampler. Furthermore Frühwirth-Schnatter
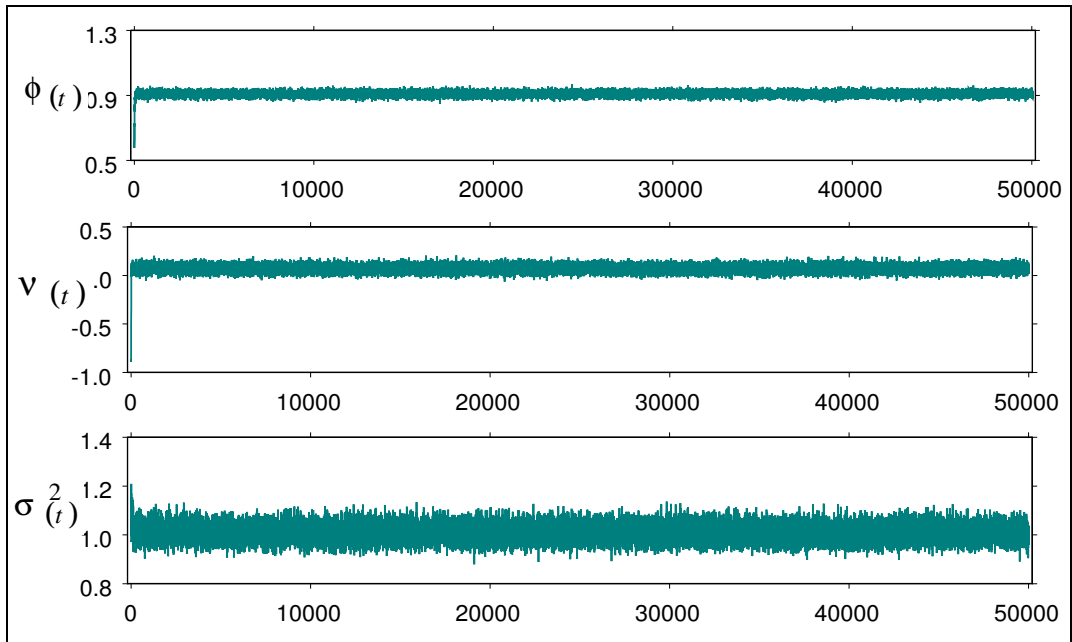
Figure 5: Sample of 50,000 parameter values simulated from the posterior distributions of the stochastic volatility model, through the single-move Gibbs sampler.
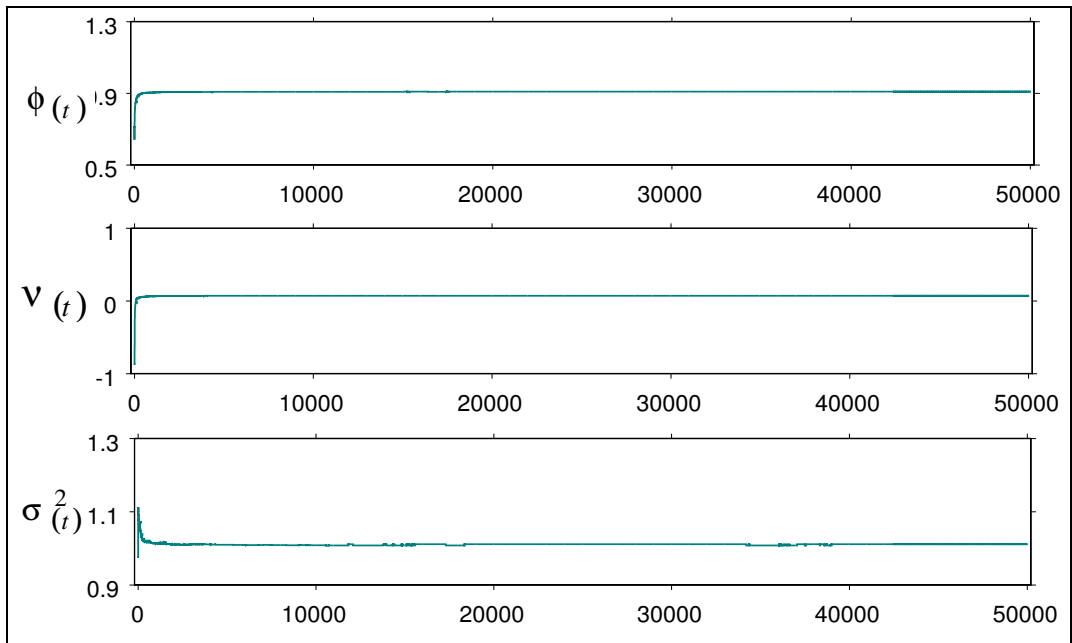

Figure 6: Ergodic averages of the parameter values simulated from the posterior distributions of the stochastic volatility model, through the single-move Gibbs sampler.
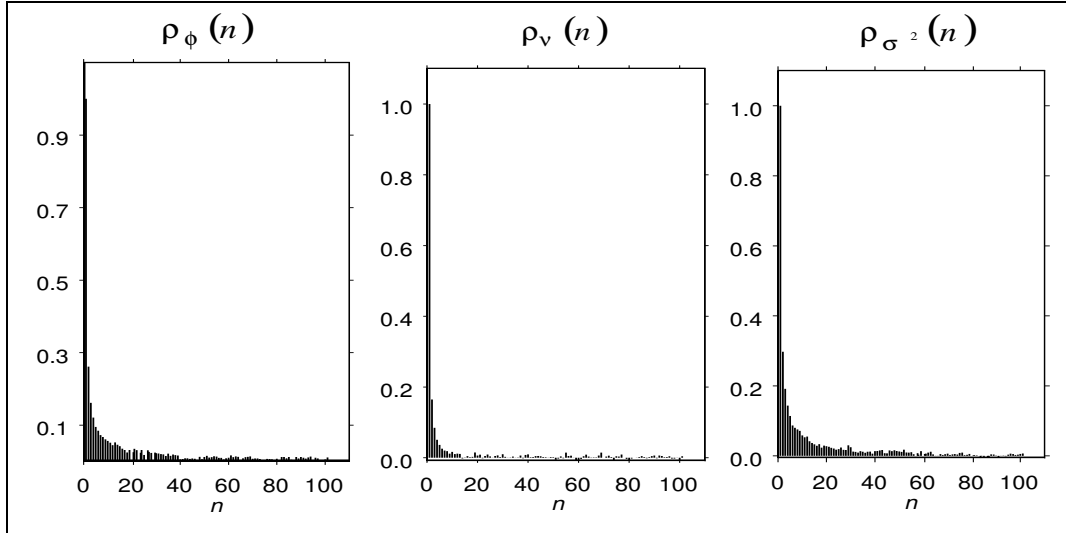
Figure 7: The graph exhibits the autocorrelation functions, ρ_ϕ, ρ_ν and ρ_{σ^2}, with lags $n = 1, \ldots, 100$, calculated on the 50,000 parameter values simulated from the posterior distribution.
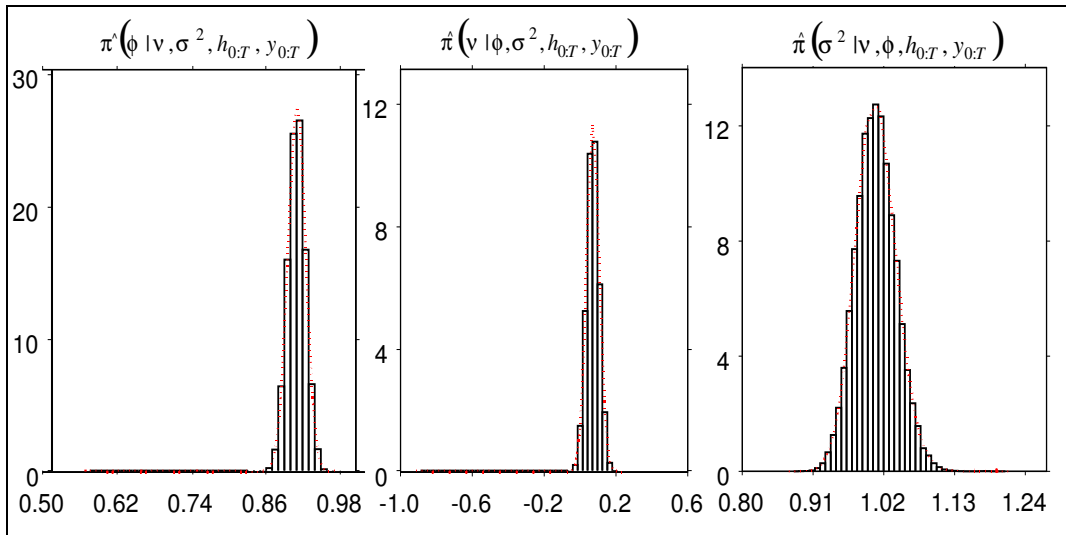

Figure 8: The figure show the histogram of 50,000 parameter values simulated from the posterior distributions and the Gaussian kernel density reconstruction.

[16] shows how the use of the multi-move Gibbs sampler improves the convergence rate of an adaptive importance sampling algorithm and makes a comparison with a set of parallel single-move Gibbs samplers. The implementation of the multi-move Gibbs sampler depends on the availability of the analytical form of filtering and smoothing densities.

We give here a general representation of the algorithm, but its implementation is strictly related to the specific analytical representation of the dynamic model. Given the simulated parameter vector obtained through the Gibbs sampler in the Algorithm 1, the multi-move Gibbs sampler is in Algorithm 3.

Algorithm 3 - *Multi-Move Gibbs Sampler* -
(i) Simulate $\theta^{(i)}$ through Algorithm 1;
(ii) Given $\theta^{(i)}$ and $\mathbf{x}_{0:T}^{(i)}$, run analytical filtering relations in order to estimate prediction and filtering densities for each $t = 0, \ldots, T$

    7. $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \theta^{(i+1)})$

    8. $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta^{(i+1)})$

(iii) Simulate state vectors by means of the recursive factorization of the smoothing density

    9. $\mathbf{x}_T^{(i+1)} \sim p(\mathbf{x}_T|\mathbf{y}_{1:T}, \theta^{(i+1)})$

    10. $\mathbf{x}_{T-1}^{(i+1)} \sim p(\mathbf{x}_{T-1}|\mathbf{x}_T^{(i+1)}, \mathbf{y}_{1:T-1}, \theta^{(i+1)})$

    11. ...

    12. $\mathbf{x}_t^{(i+1)} \sim p(\mathbf{x}_t|\mathbf{x}_{t+1}^{(i+1)}, \mathbf{y}_{1:t}, \theta^{(i+1)})$

    13. ...

    14. $\mathbf{x}_1^{(i+1)} \sim p(\mathbf{x}_1|\mathbf{x}_2^{(i+1)}, \mathbf{y}_1, \theta^{(i+1)})$

The algorithm has been derived trough the recursive smoothing relation given in equation (26). Moreover, at each simulation step the posterior density is obtained by means of estimated prediction and filtering densities. By applying the fundamental relation given in equation (28) we obtain

$$p(\mathbf{x}_t|\mathbf{x}_{t+1}^{(i+1)}, \mathbf{y}_{1:t}, \theta^{(i+1)}) = \frac{p(\mathbf{x}_{t+1}^{(i+1)}|\mathbf{x}_t, \theta^{(i+1)})\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta^{(i+1)})}{\hat{p}(\mathbf{x}_{t+1}^{(i+1)}|\mathbf{y}_{1:t}, \theta^{(i+1)})} \tag{52}$$

We stress once more that the multi-move Gibbs sampler does not easily apply to nonlinear and non-Gaussian models. Thus in a Bayesian MCMC approach, the single-move Gibbs sampler remains the only numerical solution to the estimation problem. Moreover estimation performances of both the single-move and multi-move Gibbs samplers depends on the model complexity, the parameter setting and on the specific time series. To overcome these difficulties in the literature adaptive sampling schemes have been proposed.

Sequential sampling algorithm represents a first alternative to MCMC methods. Sequential Monte Carlo algorithms allow us to make inference on general dynamic models. One of the early sequential methods proposed in the literature is Adaptive Importance Sampling, which will be discussed in the next section.

As alternative to MCMC methods, recently Cappé et al. [5] propose an adaptive importance sampling algorithm, called Population Monte Carlo (PMC). Moreover Guillin et al. [18] and Celeux, Marin, Robert [9] also apply both Gibbs sampler and PMC algorithm to latent variables models and in particular also to the basic continuous stochastic volatility model, evidencing how filtering through PMC produces better results.

3.2 Adaptive Importance Sampling

The adaptive sequential importance sampling scheme is a sequential stochastic simulation method which adapts progressively to the posterior distribution also by means of the information contained in the samples, which are simulated at the previous steps. The adaptation mechanism is based on the discrete posterior approximation and on the kernel density reconstruction of the prior and posterior densities. West [44] proposed this technique in order to estimate parameters of static models. West [45] and West and Harrison [21] successively extended the method in order to estimate parameters and states of dynamic models.

The first key idea is to use importance sampling (see Robert and Casella [37]) in order to obtain a weighted random grid of evaluation points in the state space. Let $\{\mathbf{x}_t^i, w_t^i\}_{t=1}^{n_t}$ be a sample drawn from the posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta)$ through an importance density g_t. The prediction density, given in equation (20), can be approximated as follow

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}, \theta) \approx \sum_{i=1}^{n_t} w_t^i p(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \theta) \tag{53}$$

The second key idea, implemented in the adaptive importance sampling algorithm of West [45], is to propagate points of the stochastic grid by means of the transition density and to build a smoothed approximation of the prior density. This approximation is obtained through a kernel density estimation. West [45] suggested to use Gaussian or Student-t kernels due to their flexibility in approximating other densities. For example, the Gaussian kernel reconstruction is

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}, \theta) \approx \sum_{i=1}^{n_t} w_t^i N(\mathbf{x}_{t+1}|m_t a + \mathbf{x}_t^i(1-a), h^2 V_t) \tag{54}$$

The final step of the algorithm consists in updating the prior density and in producing a random sample, $\{\mathbf{x}_{t+1}^i, w_{t+1}^i\}_{i=1}^{n_{t+1}}$, from the resulting posterior density.

The sample is obtained by using the kernel density estimate as importance density. Adaptive importance sampling is represented through Algorithm 4.

The main advantage of this algorithms relies in the smoothed reconstruction of the prior density. This kernel density estimate of the prior allows to obtain adapting importance densities and to avoid the information loss, which comes from cumulating numerical approximation over time. Furthermore this technique can be easily extended to account for a sequential estimation of the parameter (see the recent work due to Liu and West [27]).

Algorithm 4 - *Adaptive Sequential Importance Sampling* -

Given a weighted random sample $\{\mathbf{x}_t^i, w_t^i\}_{t=1}^{n_t}$, for $i = 1, \ldots, n_t$

1. *Simulate $\tilde{\mathbf{x}}_{t+1}^i \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \theta)$*

2. *Calculate $m_t = \sum_{i=1}^{n_t} w_t^i \tilde{\mathbf{x}}_{t+1}^i$, $V_t = \sum_{i=1}^{n_t} w_t^i (\tilde{\mathbf{x}}_{t+1}^i - m_t)'(\tilde{\mathbf{x}}_{t+1}^i - m_t)$*

3. *Generate from the Gaussian kernel*

$$\mathbf{x}_{t+1}^i \sim \sum_{i=1}^{n_t} w_t^i N(\mathbf{x}_{t+1}|(m_t a + \mathbf{x}_t^i(1-a)), h^2 V_t)$$

4. *Update the weights $w_{t+1}^i \propto w_t^i \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^i)p(\mathbf{x}_{t+1}^i|\mathbf{x}_t^i)}{N(\mathbf{x}_{t+1}^i|(m_t a + (1-a)\mathbf{x}_t^i), h^2 V_t)}$*

However adaptive importance sampling requires the calibration of parameters a and h, which determines the behavior of the kernel density estimate. The choice of that shrinkage parameters influences the convergence of the algorithm and heavily depends on the complexity of the model studied.

Finally, adaptive importance sampling belongs to a more general class of sequential simulation algorithms, which are particulary efficient for on-line data processing and which have some common problems like sensitivity to outliers and degeneracy. In next paragraph we review some general particle filters.

3.3 Particle Filters

In the following we focus on *Particle filters*, also referred in the literature as *Bootstrap filters*, *Interacting particle filters*, *Condensation algorithms*, *Monte Carlo filters* and on the estimation of the states of the dynamic model. See also Doucet, Freitas and Gordon [13] for an updated review on the particle filter techniques, on their applications and on the main convergence results for this kind of algorithms.

The main advantages of particle filters are that they can deal with nonlinear and non-Gaussian noise and can be easily implemented, also in a parallel mode. Moreover in contrast to Hidden Markov Model filters, which work on a state space discretised to a fixed grid, particle filters focuse sequentially on the higher probable regions of the state space. This is feature is common to Adaptive Importance Sampling algorithm exhibited in the previous section.

Assume that the parameter vector θ of the dynamic model given in equations (17), (18) and (19) is known. Different versions of the particle filter exist in the literature and different simulation approaches like rejection sampling, MCMC and importance sampling, can be used for the construction of a particle filter. We introduce particle filters applying the importance sampling reasoning.

At each time step $t + 1$, as a new observation $\mathbf{y}_{t+1}$ arrives, we are interested in predicting and filtering the hidden variables and the parameters of a general dynamic model. In particular we search how to approximate prediction an filtering densities given in Equations (20) and (21) by means of sequential Monte Carlo methods.

Assume that the weighted sample $\{\mathbf{x}_t^i, w_t^i\}_{i=1}^N$ has been drawn from the filtering density at time t

$$\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta) = \sum_{i=1}^N w_t^i \delta_{\{\mathbf{x}_t^i\}}(d\mathbf{x}_t) \tag{55}$$

Each simulated value $\mathbf{x}_t^i$ is called *particle* and the particles set, $\{\mathbf{x}_t^i, w_t^i\}_{i=1}^N$, can be viewed as a random discretization of the state space $\mathcal{X}$, with associated probabilities weights w_t^i. It is possible to approximate, by means of this particle set, the prediction density given in Eq. (21) as follows

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}, \theta) = \int_{\mathcal{X}} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \theta) p(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta) d\mathbf{x}_t \simeq \sum_{i=1}^N w_t^i p(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \theta) \tag{56}$$

which is called *empirical prediction density* and is denoted by $\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}, \theta)$. By applying the Chapman-Kolmogorov equation it is also possible to obtain an approximation of the filtering density given in Eq. (21)
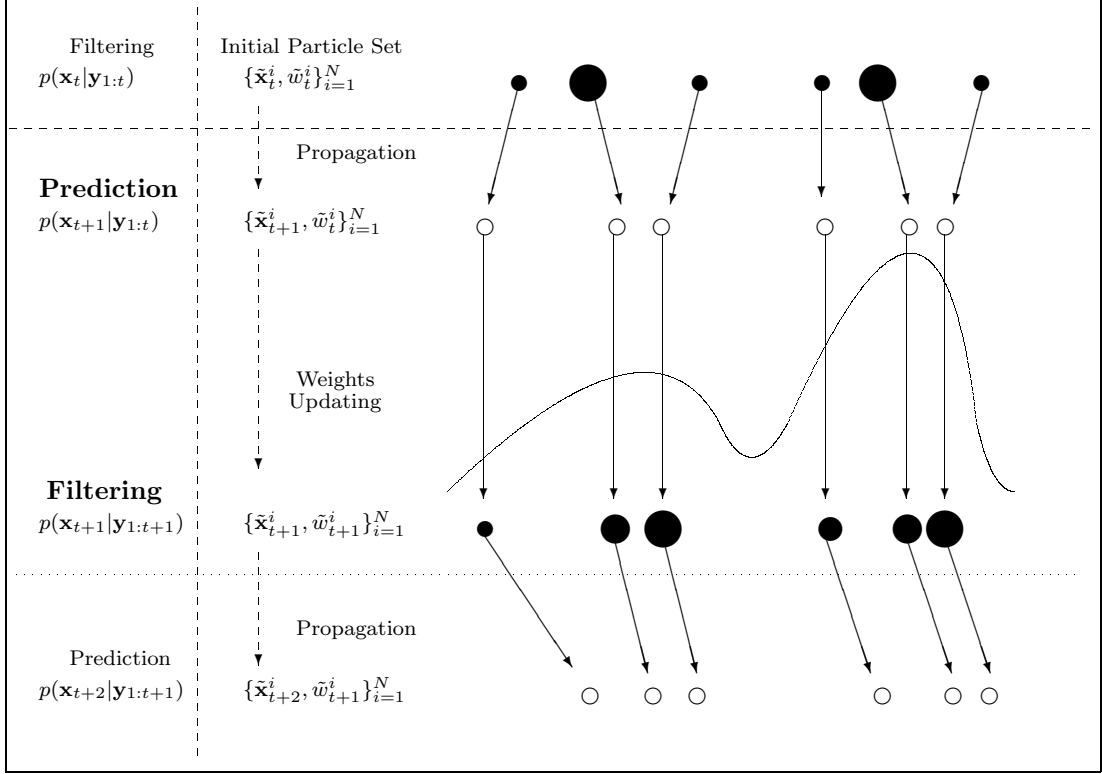
Figure 9: Particles evolution in the SIS particle filter.

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}, \theta) \propto p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, \theta)p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}, \theta) \simeq \sum_{i=1}^{N} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, \theta)p(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \theta)w_t^i \qquad (57)$$

which is called *empirical filtering density* and is denoted by $\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}, \theta)$.

Assume now that the quantity $\mathbb{E}(f(\mathbf{x}_{t+1})|\mathbf{y}_{1:t+1})$ is of interest. It can be evaluated numerically by a Monte Carlo sample $\{\mathbf{x}_{t+1}^i, w_{t+1}^i\}_{i=1}^{N}$, simulated from the filtering distribution

$$\mathbb{E}(f(\mathbf{x}_{t+1})|\mathbf{y}_{1:t+1}) \simeq \frac{\frac{1}{N}\sum_{i=1}^{N} f(x_{t+1}^i)w_{t+1}^i}{\frac{1}{N}\sum_{i=1}^{N} w_{t+1}^i}. \qquad (58)$$

A simple way to obtain a weighted sample from the filtering density at time $t+1$ is to apply importance sampling to the empirical filtering density given in equation (57). This step corresponds to propagate the initial particle set (see Fig. 9) through the importance density $q(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \mathbf{y}_{t+1}, \theta)$. Moreover if we propagate each particle of the set through the transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1}^i, \theta)$ of the dynamic model, then particle weights update as follows

$$w_{t+1}^i \propto \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t},\theta)w_t^i}{q(\mathbf{x}_{t+1}|\mathbf{x}_t^i,\mathbf{y}_{t+1},\theta)} = w_t^i\, p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^i,\theta) \tag{59}$$

This is the natural choice for the importance density, because the transition density represents a sort of prior at time t for the state x_{t+1}. However, as underlined in Pitt and Shephard [35] this strategy is sensitive to outliers. See also Crisan and Doucet [11], for a discussion on the choice of the importance densities. They focused on the properties of the importance density, which are necessary for the a.s. convergence of the sequential Monte Carlo algorithm.

The basic particle filter developed through previous equations is called *Sequential Importance Sampling* (SIS). In Algorithm 5, we give a pseudo-code representation of this method.

Algorithm 5 - *SIS Particle Filter* -
Given the initial set of particles $\{\mathbf{x}_t^i, w_t^i\}_{i=1}^N$, for $i = 1, \ldots, N$:

1. *Simulate $\mathbf{x}_{t+1}^i \sim q(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \mathbf{y}_{t+1}, \theta)$*

2. *Update the weights: $w_{t+1}^i \propto w_t^i \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^i,\theta)\, p(\mathbf{x}_{t+1}|\mathbf{x}_t^i;\theta)}{q(\mathbf{x}_{t+1}|\mathbf{x}_t^i,\mathbf{y}_{t+1},\theta)}$*

Sequential importance sampling permits to obtain recursive updating of the particles weights and is based on the sequential decomposition of the joint filtering density and on a particular choice of the importance density. To evidence these aspects we consider the following smoothing problem.

We want to approximate the smoothing density $p(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)$, of the state vectors as follows

$$p(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta) \simeq \sum_{i=1}^N \tilde{w}_{t+1}^i \delta_{\{\mathbf{x}_{0:t+1}^i\}}(d\mathbf{x}_{0:t+1}) \tag{60}$$

by simulating $\{\mathbf{x}_{0:t+1}^i\}_{i=1}^N$ from a proposal distribution $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)$ and by correcting the weights of the resulting empirical density. The correction step comes from an importance sampling argument, thus the *unnormalized particles weights*[§] are defined as follows

---

[§]Note that importance sampling requires to know the importance and the target distributions up to a proportionality constant, thus the unnormalized weights may not sum to one. However *normalized importance sampling weights* can be easily obtained as follows

$$\tilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j} \quad i = 1, \ldots, N \text{ and } t = 1, \ldots, T.$$

$$w_{t+1}^i \triangleq \frac{p(\mathbf{x}_{0:t+1}^i|\mathbf{y}_{1:t+1},\theta)}{q(\mathbf{x}_{0:t+1}^i|\mathbf{y}_{1:t+1},\theta)}. \tag{61}$$

The key idea used in SIS consists in obtaining a recursive relation for the weights updating. This property makes them particulary appealing for on-line applications.

Assume that the dynamic model of interest is the one described by equations (17), (18) and (19) and choose the importance density to factorize as follows: $q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta) = q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)q(\mathbf{x}_{t+1}|\mathbf{x}_{0:t},\mathbf{y}_{1:t+1},\theta)$, then, using the Bayes rule and the Markov property of the system, the weights can be rewritten in a recursive form

$$
\begin{aligned}
w_{t+1} \quad &\triangleq \quad \frac{p(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)}{q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)} = \\
&\overset{Bayes}{=} \quad \frac{p(\mathbf{x}_{0:t+1},\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)}{q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)} = \\
&= \quad \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)p(\mathbf{x}_{t+1},\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\mathbf{x}_{0:t},\theta)}{q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)} = \\
&= \quad \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)}{q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)} \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{0:t+1},\mathbf{y}_{1:t},\theta)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)} p(\mathbf{x}_{t+1}|\mathbf{x}_{0:t},\mathbf{y}_{1:t},\theta) = \\
&\overset{Markov}{=} \quad \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)}{q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)} \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)} p(\mathbf{x}_{t+1}|\mathbf{x}_t,\theta) = \\
&= \quad \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)} \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)p(\mathbf{x}_{t+1}|\mathbf{x}_t,\theta)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)q(\mathbf{x}_{t+1}|\mathbf{x}_{0:t},\mathbf{y}_{1:t+1},\theta)} = \\
&= \quad w_t \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)p(\mathbf{x}_{t+1}|\mathbf{x}_t,\theta)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)q(\mathbf{x}_{t+1}|\mathbf{x}_{0:t},\mathbf{y}_{1:t+1},\theta)}.
\end{aligned}
\tag{62}
$$

Thus for the i-th particle, the weight updating recursive relation is

$$w_{t+1}^i \propto w_t^i \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^i,\theta)p(\mathbf{x}_{t+1}^i|\mathbf{x}_t^i,\theta)}{q(\mathbf{x}_{t+1}^i|\mathbf{x}_t^i,\mathbf{y}_{t+1},\theta)} \tag{63}$$

Moreover, if we assume that the importance density for the state $\mathbf{x}_{t+1}$ is the transition density: $q(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{y}_{t+1},\theta) = p(\mathbf{x}_{t+1}|\mathbf{x}_t,\theta)$, then equation (63) simplifies to

$$w_{t+1}^i \propto w_t^i\, p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^i,\theta). \tag{64}$$

Example 3 - *(SV Model and SIS algorithm, Example 1 continued)*

The normalization procedure causes the loss of the unbiasness property because the quantity at the denominator is a random variable.

The first aim of this example is to show how sequential importance sampling applies to a specific Bayesian dynamic model. We consider the SV model introduced through the example 1. In the following, we assume that the parameters are known, because we want to study how SIS algorithms work just for filtering the log-volatility.

The second aim of this example, is to evidence the degeneracy problem, which arises in using SIS algorithms. Given the initial weighted particle set $\{h_t^i, w_t^i\}$, the SIS filter performs the following steps

(i) Simulate $h_{t+1}^i \sim \mathcal{N}\left(h_{t+1}|\nu + \phi h_t^i, \sigma^2\right)$

(ii) Update the weights as follow

$$
\begin{aligned}
\tilde{w}_{t+1}^i &\propto w_t^i p(y_{t+1}|h_{t+1}^i, \theta) \propto \\
&\propto w_t^i \exp\left\{-\frac{1}{2}\left[y_{t+1}^2 e^{-h_{t+1}^i} + h_{t+1}^i\right]\right\}
\end{aligned}
$$

(iii) Normalize the weights

$$
w_{t+1}^i = \frac{\tilde{w}_{t+1}^i}{\sum_{j=1}^N \tilde{w}_{t+1}^j}
$$

By applying the SIS algorithm to the synthetic data, simulated in Example 1, we obtain the filtered log-volatility represented in Fig. 10. Note that after some iterations the filtered log-volatility does not fit well to the true log-volatility. The *Root Mean Square Error* (RMSE) defined as

$$
RMSE_t = \left\{\frac{\sum_{u=1}^t (\tilde{h}_u - h_u)^2}{t}\right\}^{\frac{1}{2}}, \tag{65}
$$

measures the distance between the true and the filtered series. In the Fig. 11 the RMSE cumulates rapidly over time. Moreover, the same figure exhibits the estimated variance of the particle weights. This indicator shows how the SIS algorithm degenerates after 430 iterations. The discrete probability mass concentrates on one particle, the others particle having null probability.

$\square$

As evidenced in Example 3 and as it is well known in the literature (see for example Arulampalam, Maskell, Gordon and Clapp [1]), basic SIS algorithms have a degeneracy problem. After some iterations the empirical distribution degenerates into a single particle, because the variance of
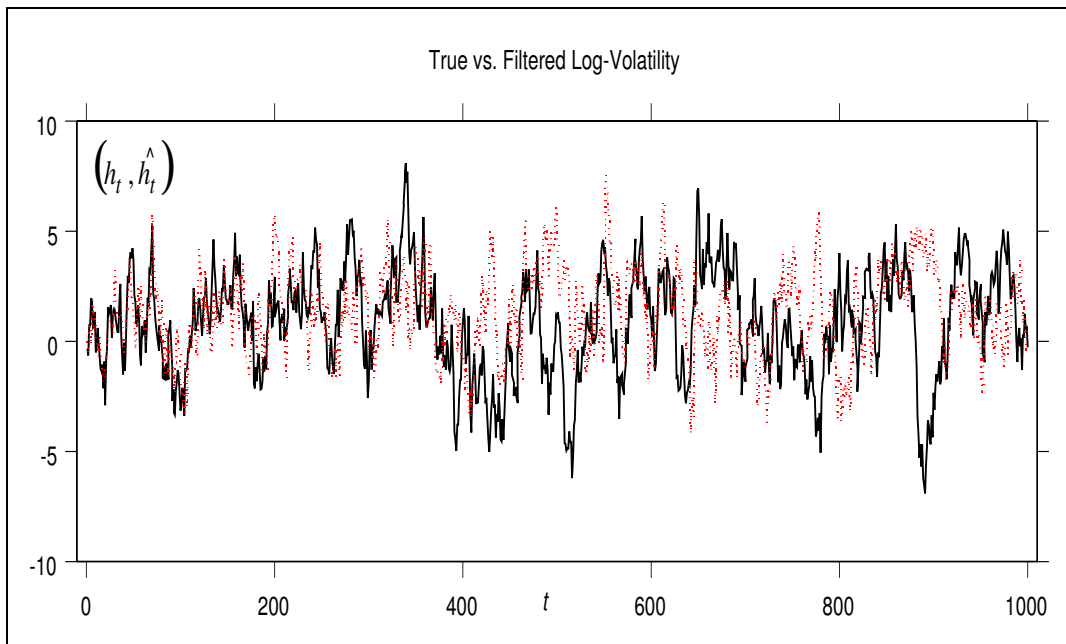
Figure 10: True (solid line) versus Filtered (dotted line) log-volatility obtained by applying a SIS algorithm, with $N = 3,000$ particles at each time step.
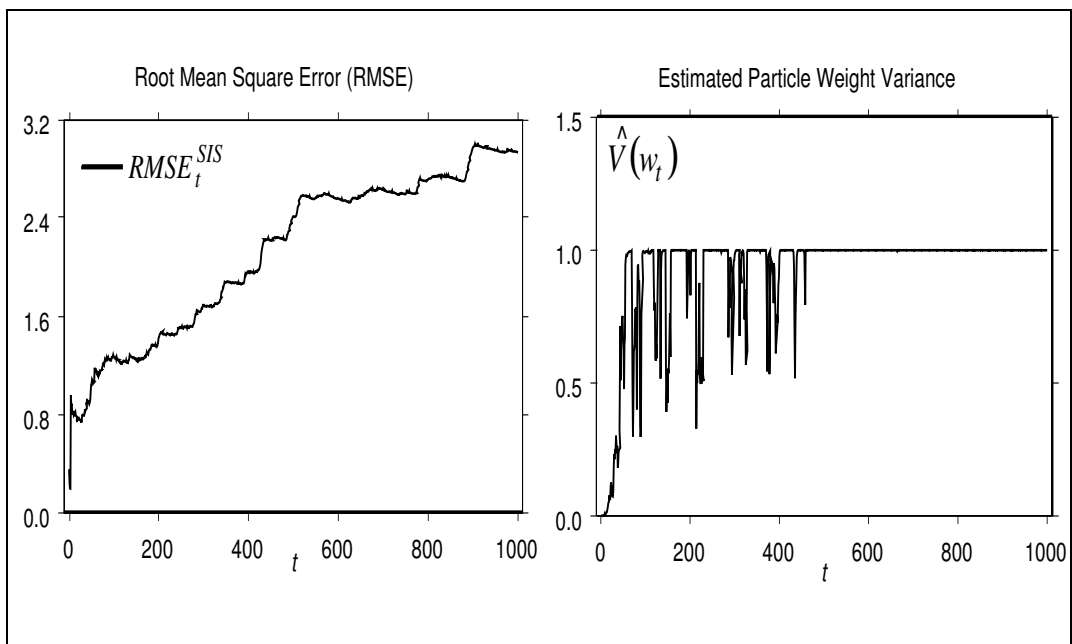

Figure 11: The Root Means Square Error between true and filtered log-volatility, and the estimated variance of the particle weights, which allows us to detect degeneracy of the particle weights.
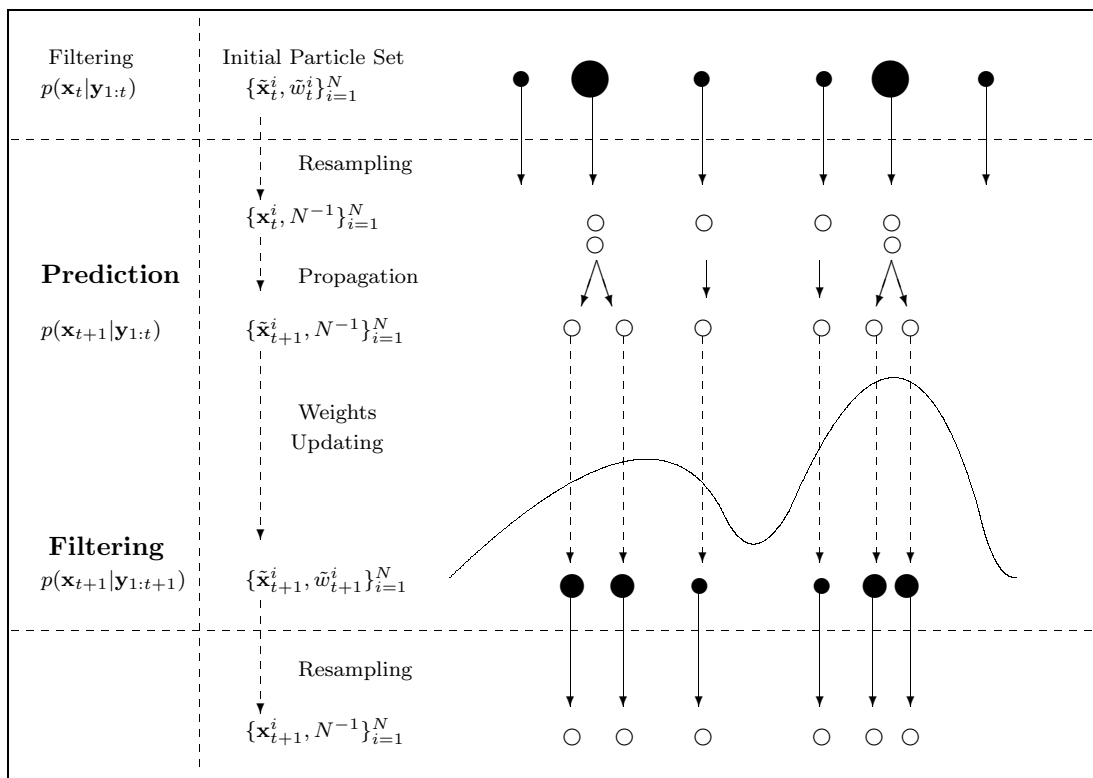
Figure 12: Particles evolution in the SIR particle filter.

the importance weights is non-decreasing over time (see Doucet et al. [14]). In order to solve the degeneracy problem, the *Sampling Importance Resampling* (SIR) algorithm has been introduced by Gordon, Salmond and Smith [17].

This algorithm belongs to a wider class of bootstrap filters, which use a re-sampling step to generate a new set of particles with uniform weights. This step introduces diversity in particle set, avoiding degeneracy. In Algorithm 6, we give a pseudo-code representation of this method.

Note that in the SIR particle filter, we assumed $q(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \mathbf{y}_{t+1}, \theta) = p(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \theta)$. Moreover, due to the resampling step, the weights are uniformly distributed over the particle set: $w_t^i = 1/N$, thus the weights updating relation becomes: $\tilde{w}_{t+1}^i \propto w_t^i \, p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^i, \theta) \propto p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^i, \theta)$.

However, the basic SIR algorithm produces a progressive impoverishment (loss of diversity) of the information contained in the particle set, because of the resampling step and of the fact that particles do not change over filter iterations.

Many solutions have been proposed in literature. We recall the *Regularised Particle Filter* proposed by Musso, Oudjane and LeGland [34], which is based on a discretisation of the continuous state space. Gilks and Berzuini [4] propose the SIR-Move algorithm, which moves

particles after the re-sampling step. Thus, particle value changes and the impoverishment is partially avoided. Finally, Pitt and Shephard [35] introduce the *Auxiliary Particle Filter* (APF) and applied it to a Gaussian ARCH-type stochastic volatility model. They find that the auxiliary particle filter works well and that the sensibility to outliers is lower than in the basic filters. In the following we focus on the APF algorithm. In order to avoid re-sampling, the APF algorithm uses an auxiliary variable to select most representative particles and to mutate them through a simulation step. Then weights of the regenerated particles are updated through an importance sampling argument. In this way particles with low probability do not survive to the selection and the information contained in the particle set is not wasted. In particular the auxiliary variable μ_t^i contains and resumes the information on the previous particle set and it is used in the selection step to sample the random particle index.

Algorithm 6 - *SIR Particle Filter* -
Given the initial set of particles $\{\mathbf{x}_t^i, w_t^i\}_{i=1}^N$, for $i = 1, \ldots, N$:

1. *Simulate $\tilde{\mathbf{x}}_{t+1}^i \sim q(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \mathbf{y}_{t+1}, \theta)$*

2. *Update the weights: $\bar{w}_{t+1}^i \propto p(\mathbf{y}_{t+1}|\tilde{\mathbf{x}}_{t+1}^i, \theta)$*

3. *Normalize the weights: $\tilde{w}_{t+1}^i = \bar{w}_{t+1}^i \left(\sum_{j=1}^N \bar{w}_{t+1}^j\right)^{-1}$, for $i = 1, \ldots, N$.*

4. *Simulate $\{\mathbf{x}_{t+1}^i\}_{i=1}^N$ from the empirical density $\{\tilde{\mathbf{x}}_t^i, \tilde{w}_t^i\}_{i=1}^N$*

5. *Assign $w_{t+1}^i = 1/N$, for $i = 1, \ldots, N$.*

Algorithm 7 - *Auxiliary Particle Filter* -
Given the initial set of particles $\{\mathbf{x}_t^j, w_t^j\}_{j=1}^N$, for $j = 1, \ldots, N$:

1. *Calculate $\mu_{t+1}^j = \mathbb{E}(\mathbf{x}_{t+1}|\mathbf{x}_t^j, \theta)$*

2. *Simulate $i^j \sim q(i|\mathbf{y}_{1:t+1}, \theta) \propto w_t^i \, p(\mathbf{y}_{t+1}|\mu_{t+1}^i, \theta)$ with $i \in \{1, \ldots, N\}$*

3. *Simulate $\mathbf{x}_{t+1}^j \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t^{i^j}, \theta)$*

4. *Update particles weights: $\tilde{w}_{t+1}^j \propto \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^j, \theta)}{p(\mathbf{y}_{t+1}|\mu_{t+1}^{i^j}, \theta)}$.*

5. *Normalize the weights: $w_{t+1}^i = \tilde{w}_{t+1}^i \left(\sum_{j=1}^N \tilde{w}_{t+1}^j\right)^{-1}$, for $i = 1, \ldots, N$.*

Note that the empirical filtering density given in Eq. (57) is a mixture of distributions, which can be reparameterised by introducing an auxiliary variable $i \in \{1, \ldots, N\}$, which indicates the component of the mixture. The joint distribution of the hidden state and of the index i is

$$
\begin{aligned}
p(\mathbf{x}_{t+1}, i | \mathbf{y}_{1:t+1}, \theta) &= \frac{p(\mathbf{y}_{t+1} | \mathbf{y}_{1:t}, \mathbf{x}_{t+1}, i)}{p(\mathbf{y}_{t+1} | \mathbf{y}_{1:t}, \theta)} p(\mathbf{x}_{t+1}, i | \mathbf{y}_{1:t}, \theta) = \\
&= \frac{p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, \theta)}{p(\mathbf{y}_{t+1} | \mathbf{y}_{1:t}, \theta)} p(\mathbf{x}_{t+1} | i, \mathbf{y}_{1:t}, \theta) p(i | \mathbf{y}_{1:t}, \theta) =
\end{aligned}
\tag{66}
$$

$$
= \frac{p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}, \theta)}{p(\mathbf{y}_{t+1} | \mathbf{y}_{1:t}, \theta)} p(\mathbf{x}_{t+1} | \mathbf{x}_t^i, \theta) w_t^i.
$$

The basic idea of the APF is to refresh the particle set while reducing the loss of information due to this operation. Thus, the algorithm generates a new set of particles by jointly simulating the particle index i (*selection step*) and the selected particle value $\mathbf{x}_{t+1}$ (*mutation step*) from the reparameterised empirical filtering density, according to the following importance density

$$
\begin{aligned}
q(\mathbf{x}_{t+1}^j, i^j | \mathbf{y}_{1:t+1}, \theta) &= q(\mathbf{x}_{t+1}^j | \mathbf{y}_{1:t+1}, \theta) q(i^j | \mathbf{y}_{1:t+1}, \theta) \\
&= p(\mathbf{x}_{t+1}^j | \mathbf{x}^{i^j}, \theta)(p(\mathbf{y}_{t+1} | \mu_{t+1}^{i^j}, \theta) w_t^{i^j})
\end{aligned}
\tag{67}
$$

for $j = 1, \ldots, N$. Note that the index is sampled using weights which are proportional to the observation density conditional on a summary statistics on the initial particle set. In this way, less informative particles are discarded. The information contained in each particle is evaluated with respect to both the observable variable and the initial particle set. By following the usual importance sampling argument, the updating relation for the particle weights is

$$
\begin{aligned}
w_{t+1}^j &\triangleq \frac{p(\mathbf{x}_{t+1}^j, i^j | \mathbf{y}_{1:t+1}, \theta)}{q(\mathbf{x}_{t+1}^j, i^j | \mathbf{y}_{1:t+1}, \theta)} \\
&= \frac{p(\mathbf{x}_{t+1}^j | \mathbf{x}^{i^j}, \theta) p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}^j, \theta) w_t^{i^j}}{p(\mathbf{x}_{t+1}^j | \mathbf{x}^{i^j}, \theta) p(\mathbf{y}_{t+1} | \mu_{t+1}^{i^j}, \theta) w_t^{i^j}} \\
&= \frac{p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}^j, \theta)}{p(\mathbf{y}_{t+1} | \mu_{t+1}^{i^j}, \theta)}
\end{aligned}
\tag{68}
$$

In Algorithm 7 we give a pseudo-code representation of the Auxiliary Particle Filter. In Examples 4 and 5 we show how resampling can improve the performance of the basic SIS algorithm. In

particular, we apply SIR and APF algorithms to the stochastic volatility model, illustrated in Example 1 and make a comparison between them, in terms of Root Means Square Error and of variance of the particle weights. Finally in Example 6 we show ho APF applies to the double latent structure of the MSSV model given in sectioin 2.2.

Example 4 - *(SV Model and SIR algorithm, Example 3 continued)*

In this example we show how the selection (or resampling) step can improve the performance of the basic Sequential Importance Sampling algorithm when applied to the Stochastic Volatility model given in Example 1.

In order to implement SIR algorithm we introduce a resampling step after the propagation of the initial set of particle $\left\{ h_t^i, w_t^i = 1/N \right\}_{i=1}^N$. The steps of the SIR are

(i) Simulate $\tilde{h}_{t+1}^i \sim \mathcal{N}\left(h_{t+1} | \nu + \phi h_t^i, \sigma^2 \right)$

(ii) Update the weights

$$
\begin{aligned}
\bar{w}_{t+1}^i &\propto w_t^i p(y_{t+1} | \tilde{h}_{t+1}^i, \theta) \propto \\
&\propto w_t^i \exp\left\{ -\frac{1}{2} \left[y_{t+1}^2 e^{-\tilde{h}_{t+1}^i} + \tilde{h}_{t+1}^i \right] \right\}
\end{aligned}
$$

(iii) Normalize the weights

$$
\tilde{w}_{t+1}^i = \frac{\bar{w}_{t+1}^i}{\sum_{j=1}^N \bar{w}_{t+1}^j}
$$

(iv) Simulate $h_{t+1}^i \sim \sum_{i=1}^N \tilde{w}_{t+1}^i \delta_{\tilde{h}_{t+1}^i}$

(v) Set $w_{t+1}^i = 1/N, \quad \forall i = 1, \ldots, N$

By applying the SIR algorithm to the synthetic data, simulated in Example 1, we obtain the filtered log-volatility represented in Fig. 13. In Fig. 14, the first graph on the left compares SIS and SIR algorithm performances evaluated in terms of RMSE. Note how resampling, (selection step), effectively improves the performance of the basic SIS filter, avoiding the degeneracy of the particle weights and reducing the RMSE.
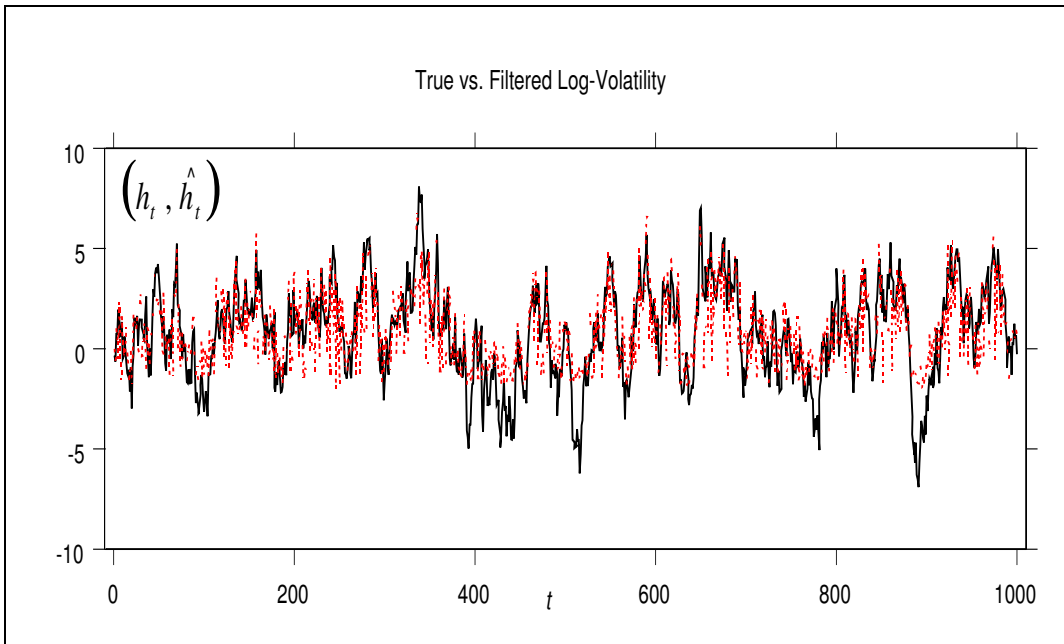
$\square$

Figure 13: True (solid line) versus filtered (dotted line) log-volatility obtained by applying a SIR algorithm, with $N = 3,000$ particles at each time step.
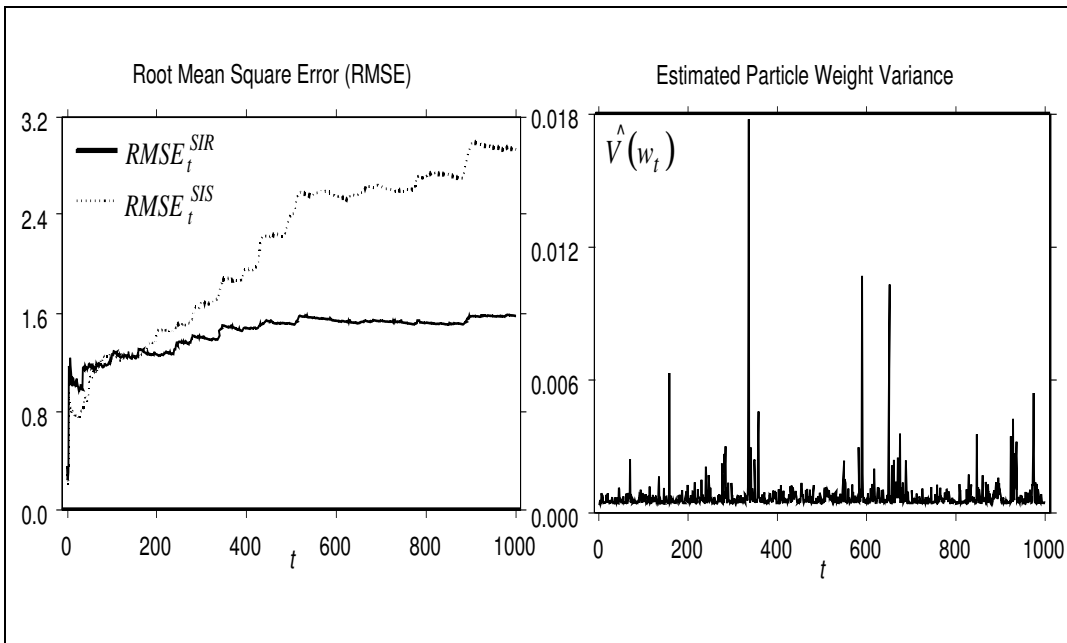

Figure 14: The first graph on the left compares SIS and SIR algorithms in terms of Root Means Square Error between true and filtered log-volatility. The second graph shows the estimated variance of the particle weights, which allows us to detect degeneracy of the particle weights.
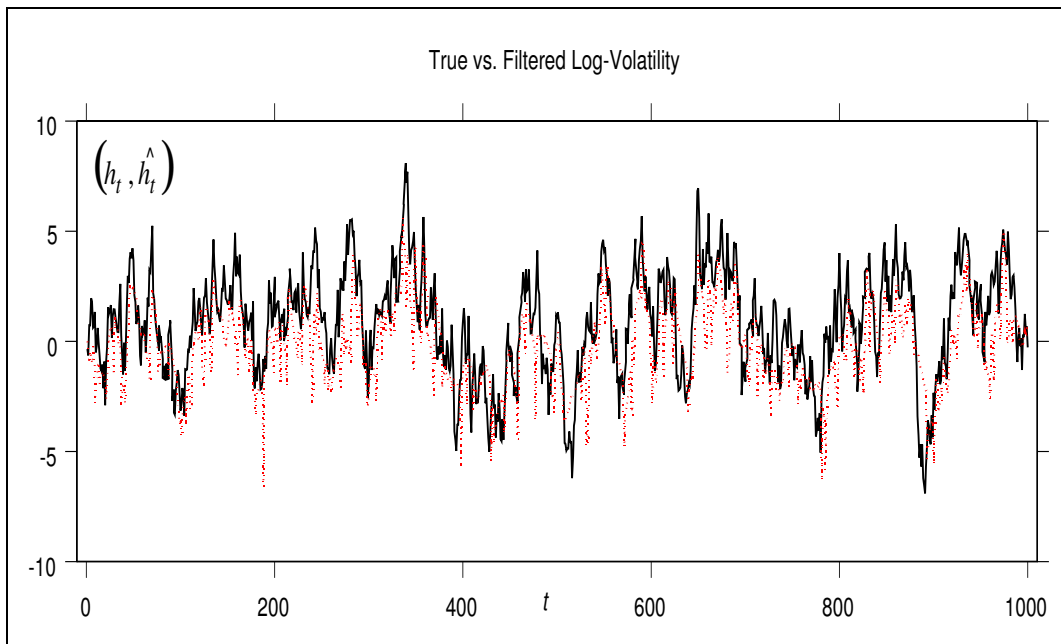
Figure 15: True (solid line) versus filtered (dotted line) log-volatility obtained by applying an APF algorithm, with $N = 3,000$ particles at each time step.
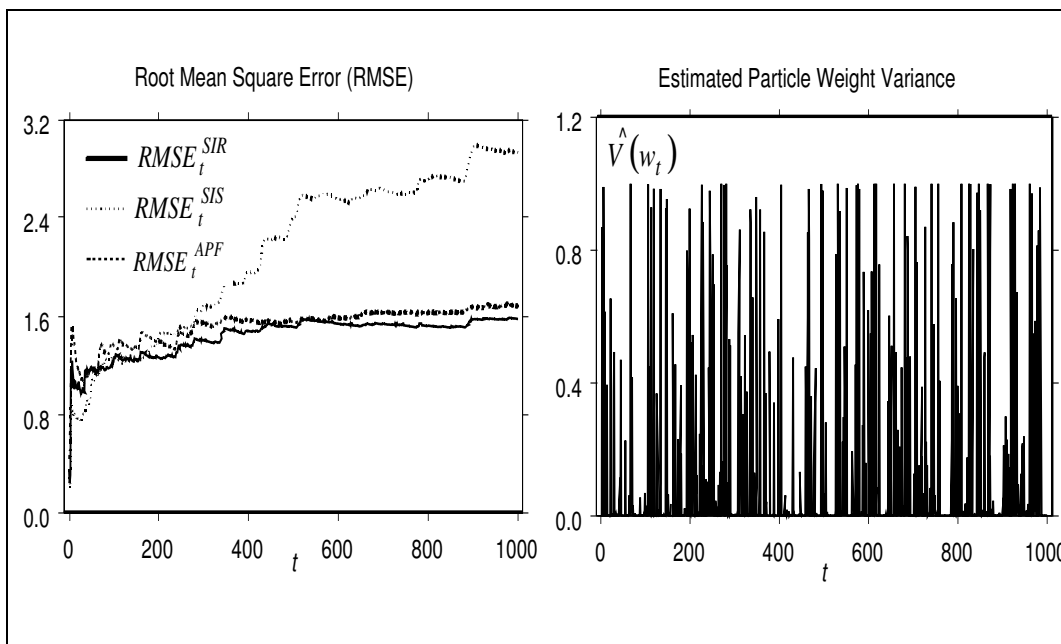

Figure 16: The first graph on the left compares SIS, SIR and APF algorithms in terms of Root Means Square Error between true and filtered log-volatility. The second graph shows the estimated particle weight variance, which allows us to detect degeneracy of the filter.

Example 5 - *(SV Model and APF algorithm, Example 3 continued)*

In this example we apply an Auxiliary Particle Filter Algorithm to the Stochastic Volatility model given in Example 1.

In order to implement the algorithm we introduce a selection step before the propagation of the set of particles $\left\{ h_t^j, w_t^j = 1/N \right\}_{j=1}^{N}$. The steps of the APF are

(i) Calculate $\mu_{t+1}^j = \phi h_t^j + \nu$ for $j =, \ldots, N$

(ii) Simulate $i^j \sim \sum_{j=1}^{N} w_t^j \mathcal{N}(y_{t+1}|\mu_{t+1}^j)$ for $j =, \ldots, N$

(iii) Simulate $\tilde{h}_{t+1}^j \sim \mathcal{N}\left(h_{t+1}|\nu + \phi h_t^{i^j}, \sigma^2 \right)$ for $j =, \ldots, N$

(iv) Update the weights

$$\bar{w}_{t+1}^j \quad \propto \quad \frac{\mathcal{N}(y_{t+1}|0, e^{\tilde{h}_{t+1}^j})}{\mathcal{N}(y_{t+1}|0, e^{\mu_{t+1}^{i^j}})}$$

(iii) Normalize the weights

$$\tilde{w}_{t+1}^j = \frac{\bar{w}_{t+1}^i}{\sum_{j=1}^{N} \bar{w}_{t+1}^j}$$

Note that, for simulating auxiliary variables i^j, Pitt and Shephard [35] suggest to use another proposal distributions, based on the Taylor expansion of the measurement distribution.

By applying the APF algorithm to the synthetic data, simulated in Example 1, we obtain the filtered log-volatility represented in Fig. 15. In Fig. 16, the first graph on the left compares SIS, SIR and APF algorithms in terms of RMSE. Although the variability in the weights of the particle set and the RMSE are greater than in the SIR algorithm, there is not degeneracy and the performance of APF algorithm is superior than that one of the basic SIS algorithm. The poor performance of the APF with respect to the SIR algorithm evidence a well known problem of this kind of algorithm. When the transition density exhibits a high noise variance the use of APF does not improve filtering results.

$\square$

We conclude this section another example of application of APF to a stochastic volatility model with jumps. As done in previous examples we assume that parameters are known. One of the main issues in particle filtering is the inclusion of the parameter estimation in the sequential

learning process. We refer to Berzuini et al. [3] and Storvik [41] for a general discussion of the problem and to Liu and West [27] for an example of joint use of the adaptive importance sampling and auxiliary particle filter.

Example 6 - *(APF and MSSV Models, example in Section 2.2 continued)*
The aim of this example is to show how particle filter algorithms apply to *Markov switching stochastic volatility models.* In these models latent factor represents the log-volatility of observed variable and the switching states are the volatility phases (high or low) of the financial market.
We apply APF algorithm to synthetic data in order to show the efficiency of the algorithm and to detect possible degeneracy of the weights.
We use a set of $N = 3,000$ particles to obtain empirical filtering and prediction densities. All computation have been carried out on a Pentium IV 2.4 Ghz, and the APF algorithm has been implemented in GAUSS 3.2.8. Figure 17 shows on-line estimation of the latent factor x_t and of the switching process s_t.
In order to detect the absence of degeneracy in the output of the APF algorithm we evaluate at each time step particle weights variance. Graph 18 shows the RMSE and the weights variance at each time step.

□

4   Conclusion

In this work we describe the Bayesian approach to general dynamic models analysis. We review the literature on the latent factor dynamic models, focusing on the Bayesian approach and discussing general filtering, predicting and smoothing relations. Moreover we recognize the importance of simulation based methods in Bayesian inference. Furthermore, we provide a review of the Bayesian simulation based methods for inference and develop some examples on simulation based filtering for stochastic volatility models. The main features of traditional MCMC methods, like single-move and multi-move Gibbs sampler have been discussed. Moreover, sequential Monte Carlo methods, such as adaptive importance sampling algorithm have been introduced. Finally we compare the basic sequential importance sampling algorithm and other recently developed sequential techniques, like particle filters.
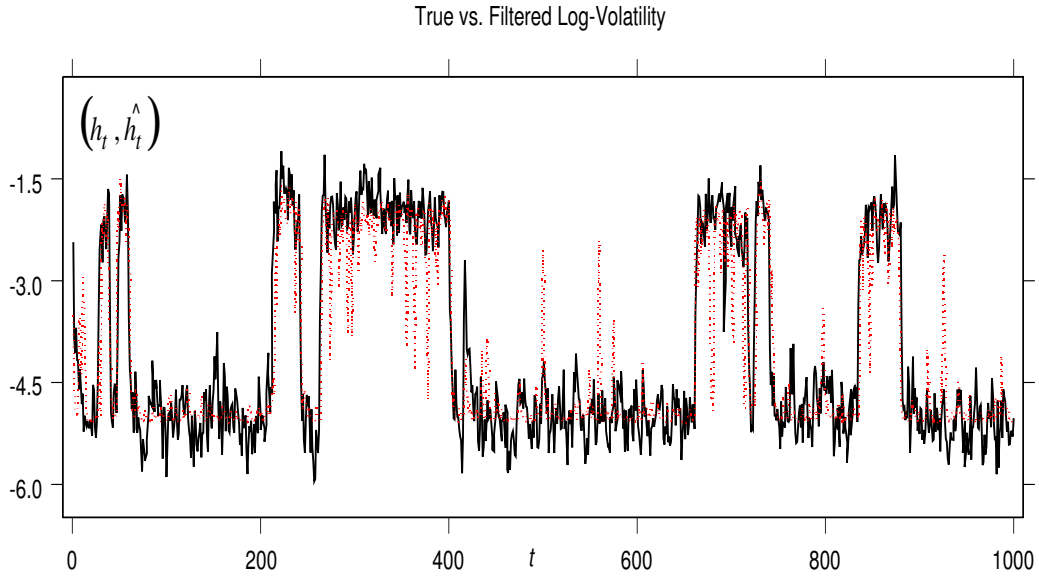
Figure 17: True (solid line) versus filtered (dotted line) log-volatility obtained by applying an APF algorithm, with $N = 3,000$ particles.
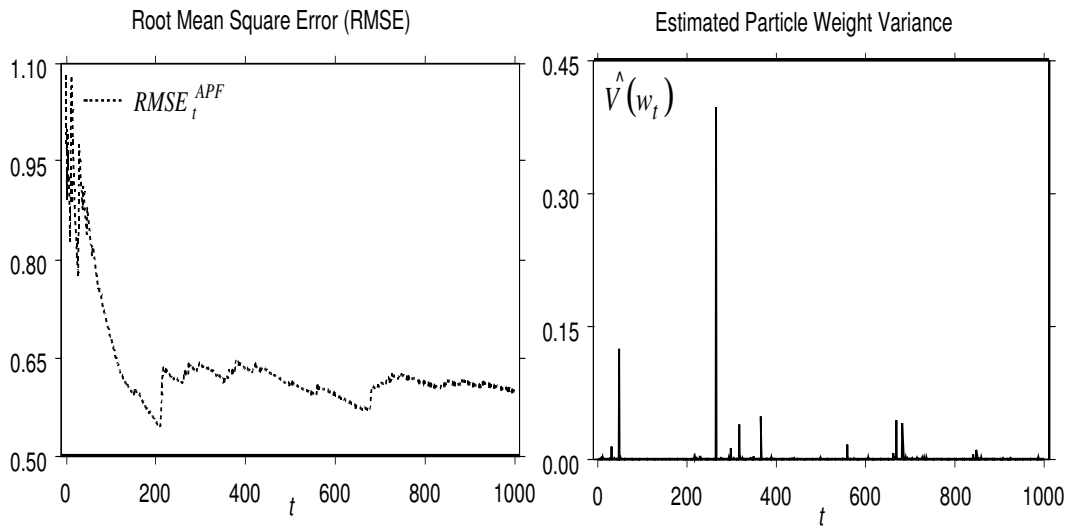

Figure 18: Root Means Square Error between true and filtered log-volatility and the estimated variance of the particle weights, which allows us to detect degeneracy of the filter.

38

References

[1] Arulampalam S., Maskell S., Gordon N. and Clapp T. (2001), A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking, *Technical Report*, QinetiQ Ltd., DSTO, Cambridge.

[2] Bauwens L., Lubrano M. and Richard J.F., (1999), *Bayesian Inference in Dynamic Econometric Models*, Oxford University Press, New York.

[3] Berzuini C., Best N.G, Gilks W.R. and Larizza C., (1997), Dynamic conditional independence models and Markov chain Monte Carlo Methods, *Journal of the American Statistical Association*, Vol. 92, pp. 1403-1441.

[4] Berzuini C., Gilks W.R. (2001), Following a moving average-Monte Carlo inference for dynamic Bayesian models, *J.R. Statist. Soc. B*, vol. 63, pp.127-146.

[5] Capp O., Guillin A., Marin J.M., and Robert C.P. (2004), Population Monte Carlo for ion channel restoration, *J. Comput. Graph. Stat.*, to appear.

[6] Carlin B.P., Polson N.G. and Stoffer D.S., (1992), A Monte Carlo Approach to Nonnormal and Nonlinear State-Space Modelling, *Journal of American Statistical Association*, Vol. 87, n.418, pp.493-500.

[7] Carter C.K. and Köhn R., (1994), On Gibbs Sampling for State Space Models, *Biometrika*, Vol. 81, n.3, 541-553.

[8] Casarin R., (2004), Bayesian Inference for Generalised Markov Switching Stochastic Volatility Models, *working paper*, CEREMADE, forthcoming.

[9] Celeux, G., Marin, J.M. and Robert, C.P. (2003),Iterated importance sampling in missing data problems, *Cahiers du Ceremade*, n. 0326.

[10] Chib S., Nardari F. and Shephard N. (2002), Markov chains Monte Carlo methods for stochastic volatility models, *Journal of Econometrics*, 108(2002), pp. 281-316.

[11] Crisan D. and Doucet A. (2000), Convergence of sequential Monte Carlo methods, *Technical Report 381*, CUED-F-INFENG.

[12] De Jong P. and Shephard N. (1995), The Simulation Smoother for Time Series Models, *Biometrika*, Vol. 82, Issue 2, pp. 339-350.

[13] Doucet A., Freitas J.G. and Gordon J., *Sequential Monte Carlo Methods in Practice*, Springer Verlag, New York.

39

[14] Doucet A., Godsill S. and Andrieu C. (2000), On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing*,Vol. 10, pp. 197-208.

[15] Durbin J. and Koopman S.J., (2001), *Time Series Analysis by State Space Methods*, Oxford University Press.

[16] Frühwirth-Schnatter S., (1994), Data augmentation and dynamic linear models, *Journal of Time Series Analysis*, Vol. 15, n.2, 183-202.

[17] Gordon N., Salmond D. and Smith A.F.M., (1993), Novel Approach to Nonlinear and Non-Gaussian Bayesian State Estimation, *IEE Proceedings-F*, 1993, Vol. 140, pp. 107-113.

[18] Guillin, A., Marin, J.M. and Robert, C.P. (2004), Estimation bayesienne approximative par echantillonnage preferentiel, *Revue de Statistique Applique*, (to appear).

[19] Hamilton J.D., (1994), *Time Series Analysis*, Princeton University Press.

[20] Harvey A.C., (1989), *Forecasting, structural time series models and the Kalman filter*, Cambridge University Press.

[21] Harrison J. and West M., (1997), *Bayesian Forecasting and Dynamic Models* 2nd Ed., Springer Verlag, New York.

[22] Kalman R.E., (1960), A new approach to linear filtering and prediction problems, *Journal of Basic Engineering, Transaction ASME, Series D*, 82, 35-45.

[23] Kalman R.E. and Bucy R.S., (1960), New results in linear filtering and prediction problems, *Transaction of the ASME-Journal of Basic Engineering, Series D*, 83, 95-108.

[24] Kim C.J. and Nelson C.R., (1999), *State-Space Models with Regime Switching*, Cambridge, MIT press.

[25] Kim S., Shephard N. and Chib S. (1998), Stochastic volatility: likelihood inference and comparison with arch models, *Review of Economic Studies*, 65, pp. 361-393.

[26] Liu J.S. and Chen R., (1998), Sequential Monte Carlo Methods for Dynamical System., *Journal of the American Statistical Association*, 93, pp. 1032-1044.

[27] Liu J. and West M., (2001), Combined Parameter and State Estimation in Simulation Based Filtering, in *Sequential Monte Carlo Methods in Practice* eds. Doucet A., Freitas J.G. and Gordon J., (2001), Springer-Verlag, New York.

[28] Liu J.S., Wong W.H. and Kong A., (1994), Covariance structure of the Gibbs sampler with applications to the comparison of estimators and augmentation schemes, *Biometrika*, 81, 27-40.

[29] Liu J.S., Wong W.H. and Kong A., (1995), Correlation structure and convergence rate of the Gibbs sampler with various scans, *Journal of the Royal Statistical Society B*, 57, 157-169.

[30] Maybeck P.S., (1982), *Stochastic Models, Estimation and Control*, vol. 1, Academic Press.

[31] Maybeck P.S., (1982), *Stochastic Models, Estimation and Control*, vol. 2, Academic Press.

[32] Maybeck P.S., (1982), *Stochastic Models, Estimation and Control*, vol. 3, Academic Press.

[33] Müller P., (1992), Alternatives to the Gibbs sampling scheme, *Tech. report*, Institute of Statistics and Decision Sciences, Duke University.

[34] Musso C, Oudjane N. and LeGland F., (2001), Improving Regularised Particle Filters, in *Sequential Monte Carlo in Practice, eds Doucet A., Freitas J.G. and Gordon J.*, (2001), Springer Verlag, New York.

[35] Pitt M. and Shephard N., (1999), Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, Vol. 94(446), pp. 590-599.

[36] Robert C.P., (2001), *The Bayesian Choice*, 2nd ed. Springer Verlag, New York.

[37] Robert C.P. and Casella G. (1999), *Monte Carlo Statistical Methods*, Springer Verlag, New York.

[38] Shephard N., (1994),Partial non-Gaussian state space,*Biometrika*, 81, 115-131.

[39] Shephard N. and Pitt M. K. (1997),Likelihood Analysis of Non-Gaussian Measurement Time Series, *Biometrika*, Vol. 84, Issue 3, pp. 653-667.

[40] So M.K.P., Lam K. and Li W.K. (1998), A stochastic volatility model with Markow switching, *Journal of Business & Economic Statistics*, 16, pp. 244-53.

[41] Storvik G. (2002), Particle filters for state space models with the presence of unknown static parameters, *IEEE Trans. on Signal Processing*, 50, pp. 281-289.

[42] Tanner M. and Wong W., (1987), The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82, 528-550.

[43] Tierney L., (1994), Markov chains for exploring posterior distributions, Ann. of Statist., 22, 1701-1786.

[44] West M., (1992), Mixture models, Monte Carlo, Bayesian updating and dynamic models, *Computer Science and Statistics*, 24, pp. 325-333.

[45] West M., (1993), Approximating posterior distribution by mixtures, *Journal of Royal Statistical Society*, B, 55, pp. 409-442.