Monte Carlo Methods using Matlab

Roberto Casarin
University Ca' Foscari, Venice
Summer School of Bayesian Econometrics
Perugia 2013

September 9, 2013

Contents

Chapter 7

Sequential Monte Carlo

7.1 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.

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 [23] and Kalman and Bucy [24] introduces filtering techniques (Kalman-Bucy filter) for continuous valued,

[1] Part of this work is:

- Billio, M., Casarin, R. and Sartore, D., (2007), Bayesian inference in dynamic models with latent factors, in Mazzi, G. L. and Savio, G., Growth and Cycle in the Eurozone, 25-44, Palgrave MacMillan, 2007.
- Casarin, R. and Marin, J.-M., (2009), Online data processing: Comparison of Bayesian regularized particle filters, Electronic Journal of Statistics, 3, 239-258.
- Billio M. and Casarin, R., (2010), Identifying Business Cycle Turning Points with Sequential Monte Carlo Methods: an on-line and real time application to the Euro area, Journal of Forecasting, 29, 145-167.
- Casarin, R. (2007), Simulation Methods for Bayesian Inference on Latent Variables Models, PhD Thesis, University Paris Dauphine

linear and Gaussian dynamic systems. Another relevant work on dynamic model analysis is due to Maybeck [35], [36], [37]. 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 [21] 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 [20] 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 [22] 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 simulation techniques for filtering and smoothing the state vector 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 [26] analyze Monte Carlo simulation methods for nonlinear 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 [31], 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 [12] 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.
In the economic and financial literature (for example on business cycle anal-

ysis and on asset pricing), dynamic models are used to capture some well known features of the economic time series: comovement, heavy tails and asymmetry. Comovement of economic variables can be modelled by means of dynamic factor models. Heavy tails and asymmetry denotes an heterogeneous dynamic of the economic variable. If the behavior of the economic or financial time series depends on the phase of the economic cycle or of the financial market, then asymmetry arises. Moreover if the frequency of large deviations from the mean of the economic or financial variable is high, then heavy tails appear. In order to model heavy tails, stochastic volatility models and non-Gaussian innovations can be used. We refer to Chapter 6 for a review on stochastic volatility models and briefly review in the following the literature on economic cycle models. In order to capture asymmetry Goldfeld and Quandt [17] introduced Markov Switching (MS) models for serially uncorrelated data, while Hamilton [19] applies MS to serially correlated time series. In their models parameter are allowed to depend on the hidden state of the economic cycle. This state may assume only two values, which are interpreted as: positive growth trend and negative growth trend.

A different way to model asymmetry in time series can be found in Tong [50] and Potter [41]. They introduce threshold autoregressive models (TAR). In this class of model, the phase of the economic cycle is determined by means of a threshold on the level of the observable variable. Parameters depend on the phase of the cycle.

All above cited approaches and in particular the original work due to Hamilton [19], have been successively extended in many directions.

Kim [25] applies Markov Switching to dynamic linear model in a Bayesian approach. Kim and Nelson [26] analyze general Markov Switching dynamic models and provide Bayesian inference tools together with MCMC simulation techniques.

In his switching model Hamilton [19] assumes that the growth rate of real output depend by an unobserved Markov switching variable. This variable can assume only states accordingly to the two phases of the business cycle: positive trend growth and negative trend growth. This hypothesis seems to be too restrictive when looking at data. In particular transitory and permanent components characterize recession phases. Thus Kim and Murray [27] and Kim and Piger [28] divide economic cycle in three phases: recession, high-growth and normal-growth.

Another kind of extension to the basic model of Hamilton [19] concerns the duration of the phases of the business cycle. Sichel [46], Durland and

McCurdy [14], Watson [51] and Diebold and Rudebusch [11] assume that the transition probability of the Markov switching problem depend on the duration of the current phase of the cycle.

Finally, multivariate extensions to the Hamilton [19] univariate MS model have been suggested by Diebold and Rudebusch [11] and Krolzig [30].

7.2 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 [23], Hamilton [20], Harrison and West [22] and in Doucet, Freitas and Gordon [12]. Throughout this chapter, we use a notation similar to that one commonly used in particle filter literature (see Doucet, Freitas and Gordon [12]).

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

$$\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta) \tag{7.1}$$

$$p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1}, \theta) \tag{7.2}$$

$$x_0 \sim p(\mathbf{x}_0|\theta)\,, \quad \text{with } t = 1, \ldots, T \tag{7.3}$$

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

By $\mathbf{x}_{0:t} \stackrel{\Delta}{=} (\mathbf{x}_0, \ldots, \mathbf{x}_t)$ and by $\mathbf{y}_{1:t} \stackrel{\Delta}{=} (\mathbf{y}_1, \ldots, \mathbf{y}_t)$ we denote respectively the collection of state vectors and of observable vectors, up to time t. We denote by $\mathbf{x}_{-t} \stackrel{\Delta}{=} (\mathbf{x}_0, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_T)$ the collection of all the state vectors

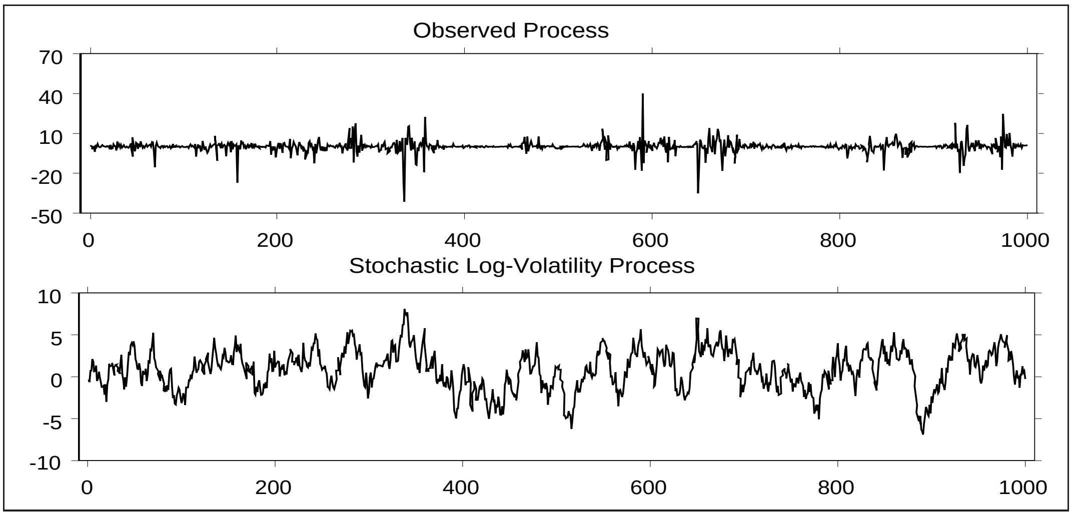

Figure 7.1: Simulated paths of 1,000 observations from the observable process y_t, with time varying volatility and of the stochastic volatility process h_t. We simulate the model given in the SV example, with $\phi = 0.9$, $\alpha = 0.1$ and $\sigma^2 = 1$.

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

$$(\mathbf{y}_t|\mathbf{x}_t) \sim p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta) \tag{7.4}$$

$$(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}, \theta) \tag{7.5}$$

$$x_0 \sim p(\mathbf{x}_0|\theta)\,, \quad \text{with } t = 1, \ldots, T. \tag{7.6}$$

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.

Example - 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

$$y_t \sim \mathcal{N}\left(0, e^{h_t}\right) \tag{7.7}$$
$$h_t \sim \mathcal{N}\left(\alpha + \phi h_{t-1}, \sigma^2\right) \tag{7.8}$$
$$h_0 \sim \mathcal{N}\left(0, \sigma^2\right)\ , \quad \text{with } t = 1, \ldots, T \tag{7.9}$$

where y_t is the observable process, with time varying volatility and h_t represents the stochastic log-volatility process. In Fig. 7.1 we exhibit a simulated path of the observable process y_t and of the stochastic volatility process h_t.

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.

7.2.1 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 (7.4), (7.5) and (7.6) 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*

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \theta) = \int_{\mathcal{X}} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}, \theta) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}, \theta) d\mathbf{x}_{t-1} \tag{7.10}$$

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

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta) = \frac{p(\mathbf{y}_t, \mathbf{x}_t|\mathbf{y}_{1:t-1}, \theta)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \theta)} = \frac{p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta) p(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \theta)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \theta)} \tag{7.11}$$

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

$$p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\theta) = \int_{\mathcal{X}} p(\mathbf{y}_t,\mathbf{x}_t|\mathbf{y}_{1:t-1},\theta)d\mathbf{x}_t = \int_{\mathcal{X}} p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{y}_{1:t-1},\theta)d\mathbf{x}_t. \tag{7.12}$$

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 (7.4), (7.5) and (7.6), the *K-steps-ahead prediction density* of the state vector $\mathbf{x}_t$ can be evaluated iteratively. The first step is

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t},\theta) = \int_{\mathcal{X}} p(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{y}_{1:t},\theta)p(\mathbf{x}_t|\mathbf{y}_{1:t},\theta)d\mathbf{x}_t \tag{7.13}$$

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

$$p(\mathbf{x}_{t+k}|\mathbf{y}_{1:t},\theta) = \int_{\mathcal{X}} p(\mathbf{x}_{t+k}|\mathbf{x}_{t+k-1},\mathbf{y}_{1:t},\theta)p(\mathbf{x}_{t+k-1}|\mathbf{y}_{1:t},\theta)d\mathbf{x}_{t+k-1} \tag{7.14}$$

where

$$p(\mathbf{x}_{t+k}|\mathbf{x}_{t+k-1},\mathbf{y}_{1:t},\theta) = \int_{\mathcal{Y}^{k-1}} p(\mathbf{x}_{t+k}|\mathbf{x}_{t+k-1},\mathbf{y}_{1:t+k-1},\theta)p(d\mathbf{y}_{t+1:t+k-1}|\mathbf{y}_{1:t},\theta) \tag{7.15}$$

where $\mathcal{Y}^k = \otimes_{i=1}^k \mathcal{Y}_i$ is the k-times Cartesian 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

$$p(\mathbf{y}_{t+K}|\mathbf{y}_{1:t},\theta) = \int_{\mathcal{Y}} p(\mathbf{y}_{t+K}|\mathbf{x}_{t+K},\mathbf{y}_{1:t+K-1},\theta)p(d\mathbf{y}_{t+1:t+K-1}|\mathbf{y}_{1:t},\theta)p(d\mathbf{x}_{t+K}|\mathbf{y}_{1:t},\theta) \tag{7.16}$$

Due to the high number of integrals that must be solved, previous densities may be difficult to evaluate with general dynamics. From a numerical

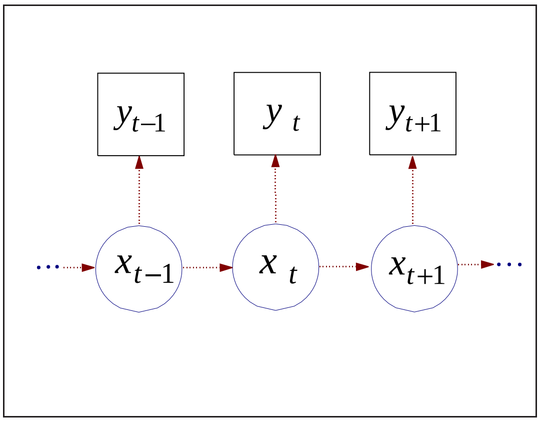

Figure 7.2: The causality structure of a Markov dynamic model with hidden states is represented by means of a *Directed Acyclic Diagram* (DAG). A box around the variable indicates the variable is known, while a circle indicates a hidden variable.

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 (7.3), (7.5) and (7.6) become

$$(\mathbf{y}_t|\mathbf{x}_t) \sim p(\mathbf{y}_t|\mathbf{x}_t, \theta) \tag{7.17}$$
$$(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, \theta) \tag{7.18}$$
$$x_0 \sim p(\mathbf{x}_0|\theta)\,, \quad \text{with } t = 1, \ldots, T. \tag{7.19}$$

The causality structure of this model has been represented through the Directed Acyclic Graph (DAG) exhibited in Fig. 7.2. Under previous assumptions the filtering and prediction densities simplify as follows

$$
\begin{aligned}
(7.20)\; p(\mathbf{x}_t|\mathbf{y}_{1:t-1},\theta) &= \int_{\mathcal{X}} p(\mathbf{x}_t|\mathbf{x}_{t-1},\theta)p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1},\theta)d\mathbf{x}_{t-1} \\
(7.21)\quad p(\mathbf{x}_t|\mathbf{y}_{1:t},\theta) &= \frac{p(\mathbf{y}_t|\mathbf{x}_t,\theta)p(\mathbf{x}_t|\mathbf{y}_{1:t-1},\theta)}{p(\mathbf{y}_t|\mathbf{y}_{0:t-1},\theta)} \\
(7.22) p(\mathbf{x}_{t+K}|\mathbf{y}_{1:t},\theta) &= \int_{\mathcal{X}} p(\mathbf{x}_{t+K}|\mathbf{x}_{t+K-1},\theta)p(\mathbf{x}_{t+K-1}|\mathbf{y}_{1:t},\theta)d\mathbf{x}_{t+K-1} \\
(7.23) p(\mathbf{y}_{t+K}|\mathbf{y}_{1:t},\theta) &= \int_{\mathcal{X}} p(\mathbf{y}_{t+K}|\mathbf{x}_{t+K},\theta)p(\mathbf{x}_{t+K}|\mathbf{y}_{1:t},\theta)d\mathbf{x}_{t+K}.
\end{aligned}
$$

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

$$(7.24)\qquad p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T},\theta) = p(\mathbf{x}_{0:T-1}|\mathbf{y}_{1:T-1},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_T,\theta)p(\mathbf{x}_T|\mathbf{x}_{T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)}.$$

which is particularly useful when processing data sequentially and it is fundamental in implementing Particle Filter algorithms. A proof of this relation is given in Appendix A.

The second recursive relation provides factorization of the *smoothing density* of the state vectors $\mathbf{x}_{0:T}$, given the information, $\mathbf{y}_{1:T}$, available at time T

$$(7.25)\qquad p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T},\theta) = p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)\prod_{t=0}^{T-1} p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta).$$

See the proof in Appendix A. The proof differs from that one given in Carter and Köhn [6] because we consider the general model in equations (7.4)-(7.6), 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

$$(7.26)\qquad p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta) = \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{y}_{1:t},\theta)p(\mathbf{x}_t|\mathbf{y}_{1:t},\theta)}{p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t},\theta)}.$$

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 [6] 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 [22] for a Bayesian analysis of the Kalman filter. See Harvey [21] for a frequentist approach to the Kalman filter.

7.2.2 Conditionally Gaussian Linear Models

A lot of models used in economic and financial literature belong to the class of the conditionally normal dynamic linear models. These are defined as follows

$$\begin{aligned} \mathbf{y}_t &= F(s_t)\mathbf{x}_t + V(s_t)\epsilon_t \quad \epsilon_t \sim \mathcal{N}(0,I) \\ \mathbf{x}_{t+1} &= G(s_t)\mathbf{x}_t + W(s_t)\eta_t \quad \eta_t \sim \mathcal{N}(0,I) \end{aligned} \tag{7.27}$$

where ϵ_t is independent of η_t and where s_t is a sequence of random variables. Looking at the Bayesian literature, Harrison and West [22] call this model *multi-process* model. In the classification proposed by these authors, if $s_t = s_{t-1} = s$, $\forall t$ the model is a multi-process of the first kind, while if s_t is a stochastic process, the model is a multi-process of second kind.

Note that if s_t is a discrete time and finite state Markov chain with known transition probabilities, the model is called *jump Markov linear system* or *Markov switching linear model* with parameters evolving over time. In the following we consider an example of Markov switching normal linear model.

Example - Stochastic Latent Factor Model with Markov Switching
Many business cycle models can be represented as a Markov switching linear model. Let y_t be the observable variable and x_t the latent factor, which has to be extracted. The switching model is

$$y_t = \alpha x_t + \sigma_\epsilon \epsilon_t \qquad \epsilon_t \sim \mathcal{N}(0,1) \tag{7.28}$$

$$x_{t+1} = \mu(s_{t+1}) + \rho\, x_t + \sigma_\eta \eta_{t+1} \qquad \eta_{t+1} \sim \mathcal{N}(0,1) \tag{7.29}$$

$$s_t \sim Markov(\mathbb{P}), \quad \text{with} \qquad s_t \in \{0,1\} \tag{7.30}$$

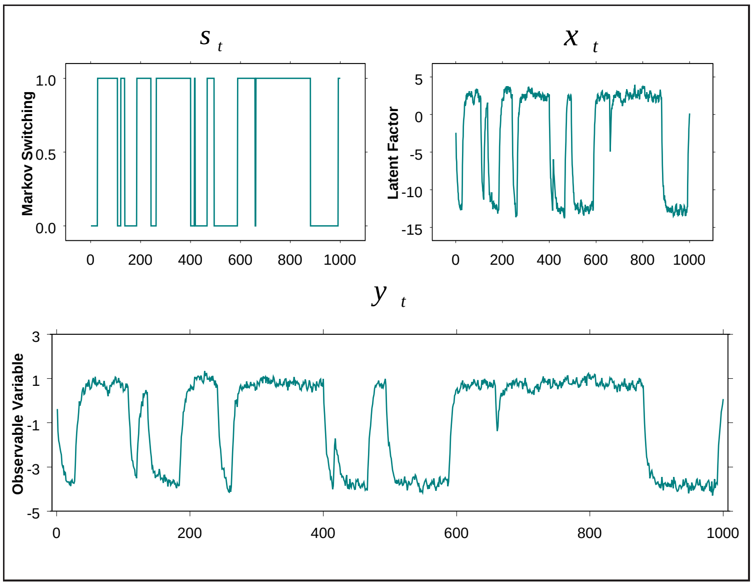

Figure 7.3: Simulation from the Markov switching stochastic trend model given in Example **??**. We set parameters to be $\alpha = 0.3, \sigma_\varepsilon = 0.1, \rho = 0.8, \mu_0 = -2.5, \mu_1 = 0.5, \sigma_\eta = 0.1, p_{11} = 0.97, p_{22} = 0.99$.

where $\mu(s_t) = \mu + \alpha s_t$ and $\mathbb{P}$ is the transition matrix.
This kind of model can be found in Kim and Nelson [26]. The absence of analytical filtering densities makes Bayesian simulation based inference a possible solution to the filtering problem.

We use parameters, estimated in Kim and Nelson [26], to simulate the MS model. Fig. 7.3 exhibits simulation paths of $1,000$ observations of the Markov switching process, the latent factor and the observable variable, respectively.

7.3 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 [5] and to the multi-move Gibbs sampler due to Carter and Köhn [6] and

Frühwirth-Schnatter [16]. Moreover some basic sequential Monte Carlo simulation methods are introduced. We examine the adaptive importance sampling algorithm due to West [52], [53], the sequential importance sampling algorithm and more advanced sequential Monte Carlo algorithms called Particle Filters (see Doucet, Freitas and Gordon [12]).

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 are estimated together with the hidden states of the model, 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

$$\theta_t \sim \delta_{\theta_{t-1}}(\theta_t), \qquad t = 0, \ldots, T \tag{7.31}$$

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. We refer to Chapter **??** for a wider treatment of the parameter estimation problem in a sequential Monte Carlo approach.

7.3.1 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

$$(\mathbf{y}_t|\mathbf{x}_t) \sim p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta) \tag{7.32}$$

$$(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}, \theta) \tag{7.33}$$

$$x_0 \sim p(\mathbf{x}_0|\theta) \tag{7.34}$$

$$\theta \sim p(\theta)\,, \quad \text{with } t = 1, \ldots, T. \tag{7.35}$$

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 [48] 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 [5] and Harrison and West [22]). 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 [6] 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 [5], 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 [33], [34] and Müller [38], 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 [29] and extended to generalized stochastic volatility models by Chib, Nardari and Shephard [8]. Shephard [44] and Shephard and Pitt [45] 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 [10].

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

$$(7.36)\ p(\theta|\mathbf{x}_{0:T},\mathbf{y}_{1:T}) \propto p(\theta)p(\mathbf{x}_0|\theta)\prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{1:t-1},\theta)$$

$$(7.37)\ p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T},\theta) \propto p(\mathbf{x}_0|\theta)\prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{y}_{1:t-1},\theta)p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{1:t-1},\theta).$$

A basic Gibbs sampler is obtained by simulating sequentially from the parameter vector posterior (*parameter simulation step*) in equation (7.36) and from the state vectors posterior (*data augmentation step*) in equation (7.37) 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 [49].

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.

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

Single-move algorithm can be implemented for general dynamic models. Moreover, note that the dynamic model given in equations (7.32)-(7.35) satisfies to the Markov property. In this case the full posterior density of the state vector, given in the single-move Gibbs sampler (see the Algorithm 2), is

$$p(\mathbf{x}_t|\mathbf{x}_{-t}, \mathbf{y}_{1:T}, \theta) \propto p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}, \theta)p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}, \theta)p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{y}_{1:t}, \theta) \tag{7.38}$$

and the implementation of the algorithm becomes easier. For the general model described in Equations (7.32)-(7.35), with measurement and transition densities depending on past values of the observable variable $\mathbf{y}_{1:t-1}$, the proof of Equation (7.38) is given in Appendix B.

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 [6]). 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 [6] and by Frühwirth-Schnatter [16] to dynamic models and the resulting algorithm is the multi-move Gibbs sampler. Furthermore Frühwirth-Schnatter [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.

The algorithm has been derived trough the recursive smoothing relation given in equation (7.25). 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 (7.26) we obtain

$$p(\mathbf{x}_t|\mathbf{x}_{t+1}^{(i+1)}, \mathbf{y}_{1:t}, \theta^{(i+1)}) = \frac{p(\mathbf{x}_{t+1}^{(i+1)}|\mathbf{x}_t, \theta^{(i+1)})\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta^{(i+1)})}{\hat{p}(\mathbf{x}_{t+1}^{(i+1)}|\mathbf{y}_{1:t}, \theta^{(i+1)})} \tag{7.39}$$

We stress once more that the multi-move Gibbs sampler does not easily apply to nonlinear and non-Gaussian models. Thus in a MCMC approach, the single-move Gibbs sampler remains the only numerical solution to the estimation problem.

Sequential sampling algorithm represents an alternative to MCMC methods.

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. $\ldots$

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. $\ldots$

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

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.

7.3.2 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 [52] proposed this technique in order to estimate parameters of static models. West [53] and West and Harrison

[22] 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 [43]) 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 (7.20), can be approximated as follow

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}, \theta) \approx \sum_{i=1}^{n_t} w_t^i p(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \theta) \tag{7.40}$$

The second key idea, implemented in the adaptive importance sampling algorithm of West [53], 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 [53] suggested to use Gaussian or Student-t kernels due to their flexibility in approximating other densities. For example, the Gaussian kernel reconstruction is

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}, \theta) \approx \sum_{i=1}^{n_t} w_t^i \mathcal{N}(\mathbf{x}_{t+1}|m_t a + \mathbf{x}_t^i(1-a), h^2 V_t) \tag{7.41}$$

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.

Algorithm 4 - *Adaptive Sequential Importance Sampling -*
Given a weighted random sample $\{\mathbf{x}_t^i, w_t^i\}_{t=1}^{n_t}$, *for* $i = 1, \dots, n_t$

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*

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

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

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 [32]).

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.

7.3.3 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 [12] 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 (7.17), (7.18) and (7.19) 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 (7.20) and (7.21) 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

$$\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta) = \sum_{i=1}^{N} w_t^i \delta_{\{\mathbf{x}_t^i\}}(d\mathbf{x}_t) \tag{7.42}$$

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. (7.21) as follows

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}, \theta) = \int_{\mathcal{X}} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \theta) p(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta) d\mathbf{x}_t \simeq \sum_{i=1}^{N} w_t^i p(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \theta) \tag{7.43}$$

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. (7.21)

(7.44)
$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1},\theta) \propto p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t},\theta) \simeq \sum_{i=1}^{N} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)p(\mathbf{x}_{t+1}|\mathbf{x}_t^i,\theta)w_t^i$$

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

$$\mathbb{E}(f(\mathbf{x}_{t+1})|\mathbf{y}_{1:t+1}) \simeq \frac{\frac{1}{N}\sum_{i=1}^N f(x_{t+1}^i)w_{t+1}^i}{\frac{1}{N}\sum_{i=1}^N w_{t+1}^i}. \tag{7.45}$$

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 (7.44). This step corresponds to propagate the initial particle set (see Fig. 7.4) 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

$$w_{t+1}^i \propto \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t},\theta)w_t^i}{q(\mathbf{x}_{t+1}|\mathbf{x}_t^i,\mathbf{y}_{t+1},\theta)} = w_t^i\, p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^i,\theta) \tag{7.46}$$

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 [40] this strategy is sensitive to outliers. See also Crisan and Doucet [9], 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.

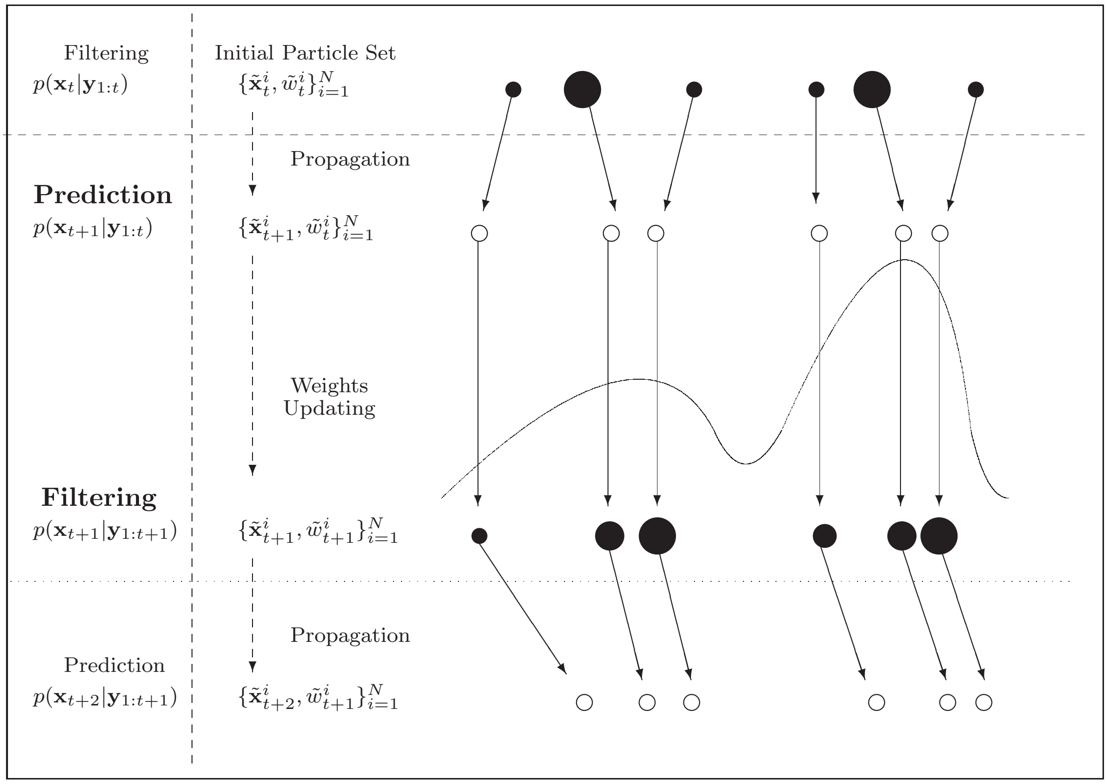

Figure 7.4: Particles evolution in the SIS particle filter.

The 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

$$p(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1}, \theta) \simeq \sum_{i=1}^{N} \tilde{w}_{t+1}^i \delta_{\{\mathbf{x}_{0:t+1}^i\}}(d\mathbf{x}_{0:t+1}) \tag{7.47}$$

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*[1] are defined as follows

$$w_{t+1}^i \triangleq \frac{p(\mathbf{x}_{0:t+1}^i|\mathbf{y}_{1:t+1},\theta)}{q(\mathbf{x}_{0:t+1}^i|\mathbf{y}_{1:t+1},\theta)}. \tag{7.48}$$

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.

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

Assume that the dynamic model of interest is the one described by equations (7.17), (7.18) and (7.19) 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 the weights can be rewritten in a recursive form

$$w_{t+1}^i = w_t^i \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^i,\theta)p(\mathbf{x}_{t+1}^i|\mathbf{x}_t^i,\theta)}{q(\mathbf{x}_{t+1}^i|\mathbf{x}_{t+1}^i,\mathbf{y}_{t+1},\theta)} \tag{7.49}$$

This relation can be proved by using the Bayes rule and the Markov property of the system (see the proof in Appendix B).

SV Model and SIS algorithm, Example

[1] 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

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

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

(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 the SV example, we obtain the filtered log-volatility represented in Fig. 7.5. 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

$$
RMSE_t = \left\{\frac{\sum_{u=1}^t (\tilde{h}_u - h_u)^2}{t}\right\}^{\frac{1}{2}}, \tag{7.50}
$$

measures the distance between the true and the filtered series. In the Fig. 7.6 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.

As evidenced in the SV Example and as it is well known in the literature (see for example Arulampalam, Maskell, Gordon and Clapp [1]), that basic

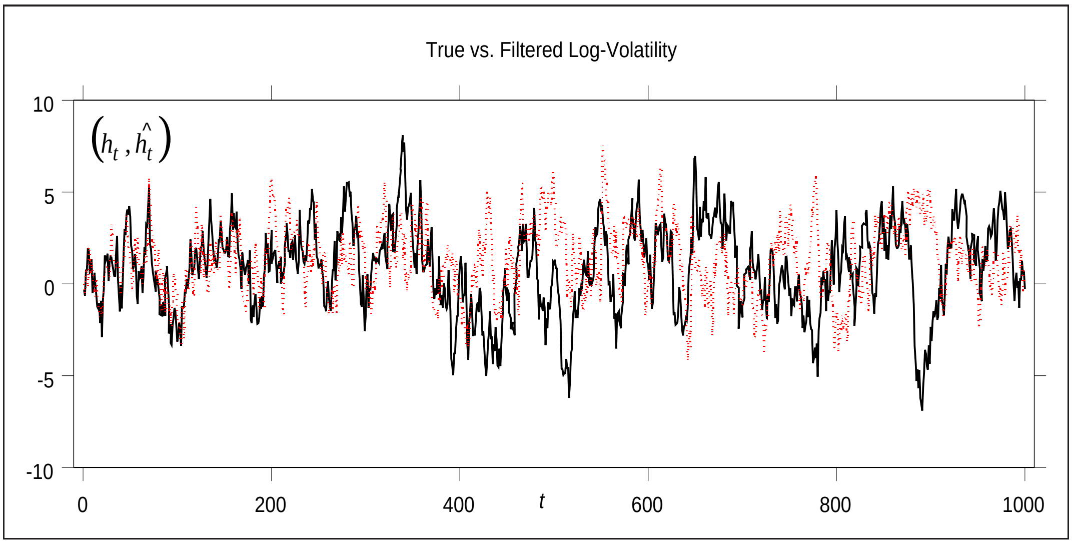

Figure 7.5: True (solid line) versus Filtered (dotted line) log-volatility obtained by applying a SIS algorithm, with $N = 3,000$ particles at each time step.

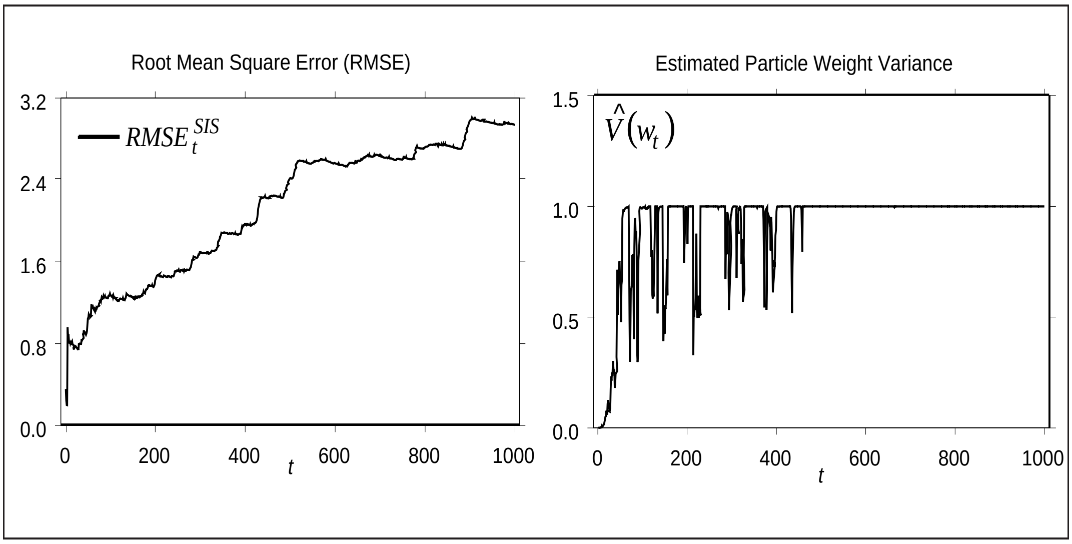

Figure 7.6: The Root Means Square Error between true and filtered log-volatility, and the estimated variance of the particle weights, which allows us to detect degeneracy of the particle weights.

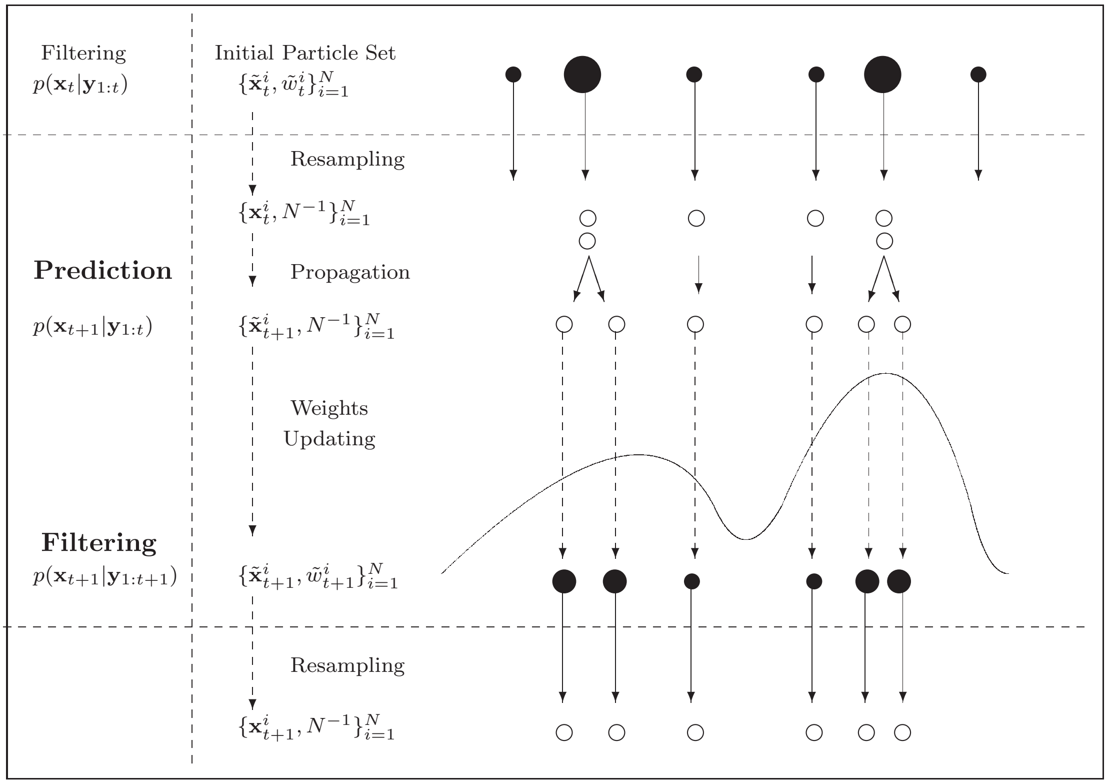

Figure 7.7: Particles evolution in the SIR particle filter.

SIS algorithms have a degeneracy problem. After some iterations the empirical distribution degenerates into a single particle, because the variance of the importance weights is non-decreasing over time (see Doucet et al. [13]). In order to solve the degeneracy problem, the *Sampling Importance Resampling* (SIR) algorithm has been introduced by Gordon, Salmond and Smith [18].

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 [39], which is based on a discretisation of the continuous state space.

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:* $\tilde{w}_{t+1}^i = \bar{w}_{t+1}^i \, (\sum_{j=1}^N \bar{w}_{t+1}^j)^{-1}$, *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* $i = 1, \ldots, N$.

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 [40] 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. Note that the empirical filtering density given in Eq. (7.44) is a mixture of distributions, which can be reparameterised by introducing an auxiliary variable $i \in \{1, \dots, N\}$, which indicates the component of the mixture. The joint distribution of the hidden state and of the index i is

$$\begin{aligned} p(\mathbf{x}_{t+1}, i|\mathbf{y}_{1:t+1}, \theta) &= \frac{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t}, \mathbf{x}_{t+1}, i)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t}, \theta)} p(\mathbf{x}_{t+1}, i|\mathbf{y}_{1:t}, \theta) = \\ &= \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, \theta)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t}, \theta)} p(\mathbf{x}_{t+1}|i, \mathbf{y}_{1:t}, \theta) p(i|\mathbf{y}_{1:t}, \theta) = \\ &= \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}, \theta)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t}, \theta)} p(\mathbf{x}_{t+1}|\mathbf{x}_t^i, \theta) w_t^i. \end{aligned} \tag{7.51}$$

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

$$\begin{aligned} q(\mathbf{x}_{t+1}^j, i^j|\mathbf{y}_{1:t+1}, \theta) &= q(\mathbf{x}_{t+1}^j|\mathbf{y}_{1:t+1}, \theta) q(i^j|\mathbf{y}_{1:t+1}, \theta) \\ &= p(\mathbf{x}_{t+1}^j|\mathbf{x}^{i^j}, \theta)(p(\mathbf{y}_{t+1}|\mu_{t+1}^{i^j}, \theta) w_t^{i^j}) \end{aligned} \tag{7.52}$$

for $j = 1, \dots, N$. 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

$$\begin{aligned} w_{t+1}^j &\triangleq \frac{p(\mathbf{x}_{t+1}^j, i^j|\mathbf{y}_{1:t+1}, \theta)}{q(\mathbf{x}_{t+1}^j, i^j|\mathbf{y}_{1:t+1}, \theta)} \\ &= \frac{p(\mathbf{x}_{t+1}^j|\mathbf{x}^{i^j}, \theta) p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^j, \theta) w_t^{i^j}}{p(\mathbf{x}_{t+1}^j|\mathbf{x}^{i^j}, \theta) p(\mathbf{y}_{t+1}|\mu_{t+1}^{i^j}, \theta) w_t^{i^j}} \\ &= \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^j, \theta)}{p(\mathbf{y}_{t+1}|\mu_{t+1}^{i^j}, \theta)} \end{aligned} \tag{7.53}$$

In Algorithm 7 we give a pseudo-code representation of the Auxiliary Particle Filter.

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

In the following examples 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 and make a comparison between them, in terms of Root Means Square Error.

Example - SV Model and SIR algorithm
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 the SV example.

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

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

(ii) Update the weights

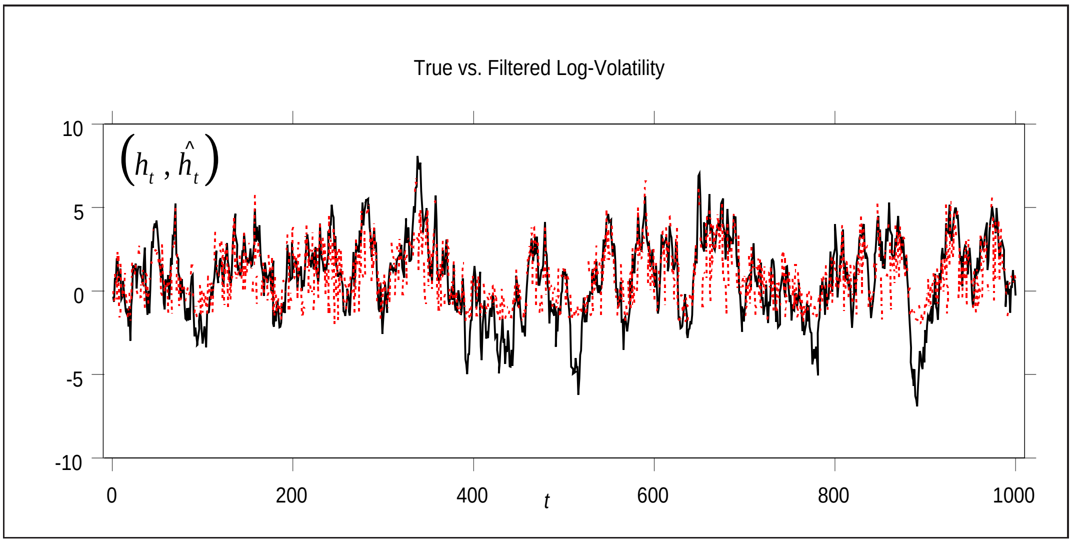

Figure 7.8: True (solid line) versus Filtered (dotted line) log-volatility obtained by applying a SIR algorithm, with $N = 3,000$ particles at each time step.

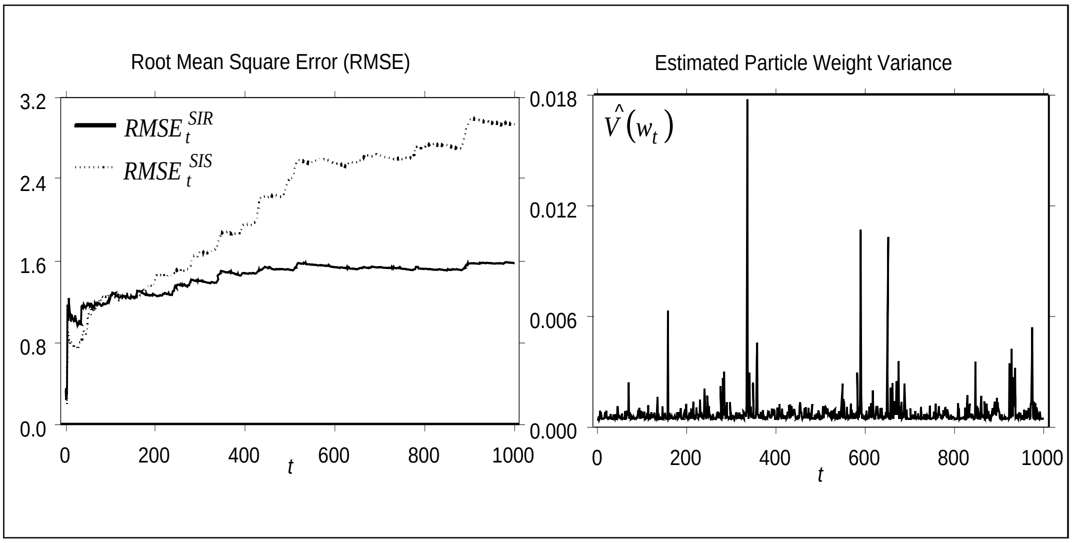

Figure 7.9: The first graph compares SIS and SIR algorithms in terms of Root Means Square Error between true and filtered log-volatility. The second graph shows the estimated variance of the particle weights, which allows us to detect degeneracy of the particle weights.

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

(iii) Normalize the weights

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

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

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

By applying the SIR algorithm to the synthetic data, simulated in the SV example, we obtain the filtered log-volatility represented in Fig. 7.8. In Fig. 7.9, 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.

Example SV Model and APF algorithm
In this example we apply an Auxiliary Particle Filter Algorithm to the SV example.

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

(i) Calculate $\mu^j_{t+1} = \phi h^j_t + \alpha$ for $j =,\ldots,N$

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

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

(iv) Update the weights

$$\bar{w}^j_{t+1} \propto \frac{\mathcal{N}(y_{t+1}|0, e^{\tilde{h}^j_{t+1}})}{\mathcal{N}(y_{t+1}|0, e^{\mu^{ij}_{t+1}})}$$

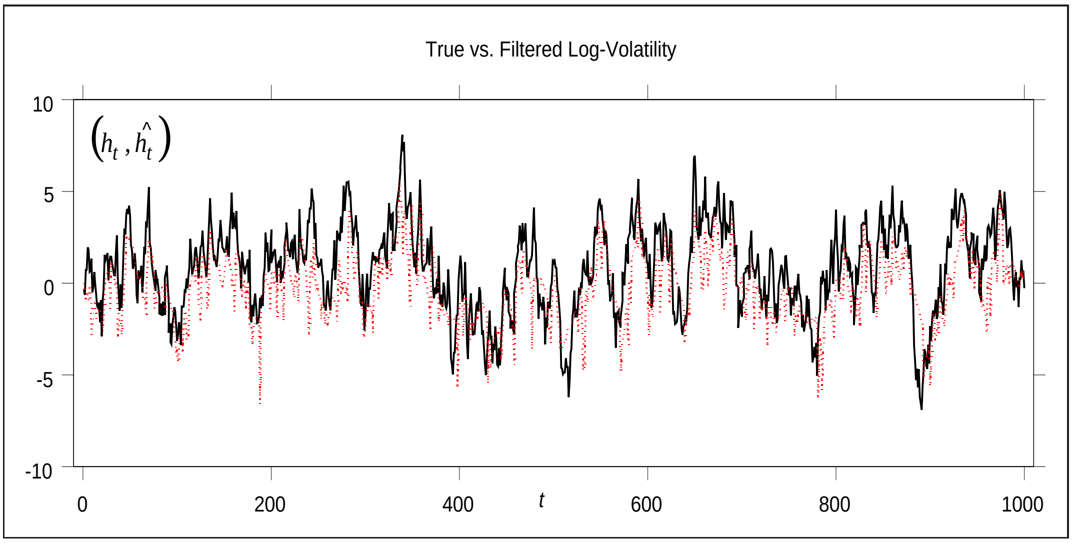

Figure 7.10: True (solid line) versus Filtered (dotted line) log-volatility obtained by applying an APF algorithm, with $N = 3,000$ particles at each time step.

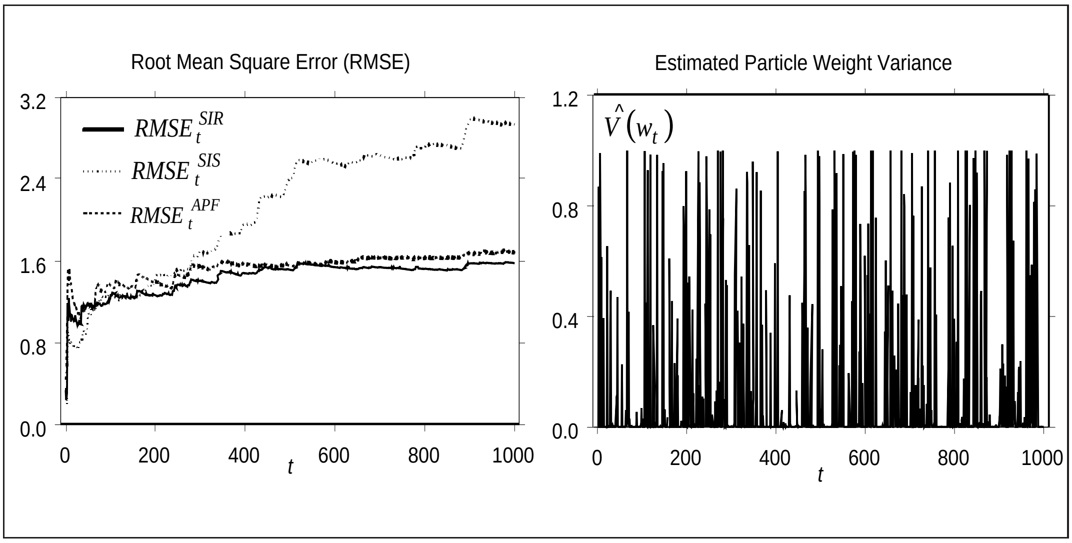

Figure 7.11: The first graph compares SIS, SIR and APF algorithms in terms of Root Means Square Error between true and filtered log-volatility. The second graph shows the estimated variance of the particle weights, which allows us to detect degeneracy of the particle weights.

(iii) Normalize the weights

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

Note that, for simulating auxiliary variables i^j, Pitt and Shephard [40] 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 SV example, we obtain the filtered log-volatility represented in Fig. 7.10. In Fig. 7.11, 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.

We conclude this section with a brief discussion of the problem of parameter estimation (see Chapter 6 for a more detailed analysis), for dynamic models with hidden variables, in a sequential data-processing approach. In principle parameter estimate and state filtering can be treated separately (see Storvik [47]). In many applications of particle filter techniques, parameters are treated as known and MCMC parameter estimates are used instead of the true parameter values. But in this way parameter estimate are not continuously updated as the hidden states. MCMC is typically a off-line approach, it does not allow the sequential updating of parameter estimates, as new observations arrive. Moreover, when applied sequentially, MCMC estimation method is more time consuming than particle filter algorithms.

One of the main issue in researching on particle filter is the inclusion of the parameter estimation procedure in the state filtering algorithm. Some studies have already extended sequential Monte Carlo techniques in order to jointly estimate state vectors and parameter. See Berzuini et al. [3] and Storvik [47] for a general discussion of the problem and Liu and West [32] for the joint application of the adaptive importance sampling and the auxiliary particle filter. In the following we briefly show how joint parameter estimation and states filtering apply to a Bayesian dynamic model. In particular we apply the algorithm of Liu and West to the stochastic latent factor model given in Example **??**. We refer to the next chapter of this manual for a review of the on-line parameter estimation problem, with applications to Markov switching stochastic volatility models.

Example - APF and Latent Factor Models

The aim of this example is to show how particle filter algorithms apply to a widely used class of economic dynamic models: *Markov switching stochastic latent factor models*. In these models latent factor represents the trend of the market, while the switching states are the phases (growth and recession) of the market or of the economy.

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 refer to the Markov switching mode presented in the previous sections and apply the algorithm due to Liu and West [32]. This algorithm combines adaptive importance sampling for sequentially estimating the parameter vector with the auxiliary particle filter for filtering and predicting the hidden state. Observe that the latent structure of the MS model in the example exhibits two levels. The first one is given by the stochastic latent factor x_t and the second one is given by the regime switching process s_t. This stochastic structure makes the inference more difficult than in the simpler Hamilton's MS models.

The adaptation of the algorithm of Liu and West [32] to our MS model give us the following Particle Filter algorithm.

Algorithm 8 - *APF for the Business Cycle Model* -
Given an initial set of particles $\{\mathbf{x}_t^i, s_t^i, \theta_t^i, w_t^i\}_{i=1}^N$*:*

1. *Compute* $V_t = \sum_{i=1}^N (\theta_t^i - \bar{\theta}_t)(\theta_t^i - \bar{\theta}_t)' w_t^i$ *and* $\bar{\theta}_t = \sum_{i=1}^N \theta_t^i w_t^i$
2. *For* $i = 1, \ldots, N$ *and with* a *and* b *well chosen tuning parameters, calculate the following summarizing constant:*
   (a) $\tilde{S}_{t+1}^i = \underset{l \in 1,2}{arg\, max}\, \mathbb{P}(s_{t+1} = l | s_t = s_t^i)$
   (b) $\tilde{X}_{t+1}^i = \mu_t^i + \alpha_t^i \tilde{S}_{t+1}^i + \rho_t^i x_t^i$
   (c) $\tilde{\theta}_t^i = a\theta_t^i + (1 - a)\bar{\theta}_t$, *where* $\tilde{\theta} = (\tilde{\alpha}, \tilde{\sigma}_\varepsilon, \tilde{\rho}, \tilde{\mu}, \tilde{\alpha}, \tilde{\sigma}_\eta, \tilde{p}_{11}, \tilde{p}_{22})$
3. *For* $i = 1, \ldots, N$*:*
   (a) *Simulate* $k^i \propto q(k | \mathbf{y}_{1:t+1}, \theta) = \mathcal{N}(\mathbf{y}_{t+1} | \tilde{\alpha}_t^k \tilde{X}_{t+1}^k, \tilde{\sigma}_{\varepsilon\, t}^k) w_t^k$, *with* $k \in \{1, \ldots, N\}$
   (b) *Simulate* θ_{t+1}^i *from* $\mathcal{N}(\tilde{\theta}_t^{k^i}, b^2 V_t)$
   (c) *Simulate* $s_{t+1}^i \in \{1, 2\}$ *from* $\mathbb{P}(s_{t+1}^i = i | s_t^{k^i})$
   (d) *Simulate* x_{t+1}^i *from* $\mathcal{N}(\mu_{t+1}^i + \alpha_{t+1}^i s_{t+1}^i + \rho_{t+1}^i x_t^{k^i}, \sigma_{\eta\, t+1}^i)$
4. *Update weights*
$$\tilde{w}_{t+1}^i \propto \mathcal{N}(\mathbf{y}_{t+1} | \alpha_{t+1}^i x_{t+1}^i, \sigma_{\varepsilon\, t+1}^i) / \mathcal{N}(\mathbf{y}_{t+1} | \tilde{\alpha}_t^{k^i} \tilde{X}_{t+1}^{k^i}, \tilde{\sigma}_{\varepsilon\, t}^{k^i})$$
5. *Normalize weights* $w_{t+1}^i = \tilde{w}_{t+1}^i (\sum_{i=1}^N \tilde{w}_{t+1}^i)^{-1}$, *for* $i = 1, \ldots, N$.

The tuning parameters a and b are equal to $\frac{3\delta - 1}{2\delta}$ and $\sqrt{1 - a^2}$ respectively, where we chose $\delta = 0.99$ as suggested in West [53].

Figure 7.12 shows on-line estimation of parameters α, σ_ε, ρ, μ_0, μ_1, σ_η, p_{11}, p_{22} obtained by running APF algorithm on the synthetic dataset exhibited in Fig. 7.3. We use a set of $N = 1000$ particles to obtain empirical filtering and prediction densities. Figure 7.13 shows on-line estimation of the latent factor x_t and of the switching process s_t.

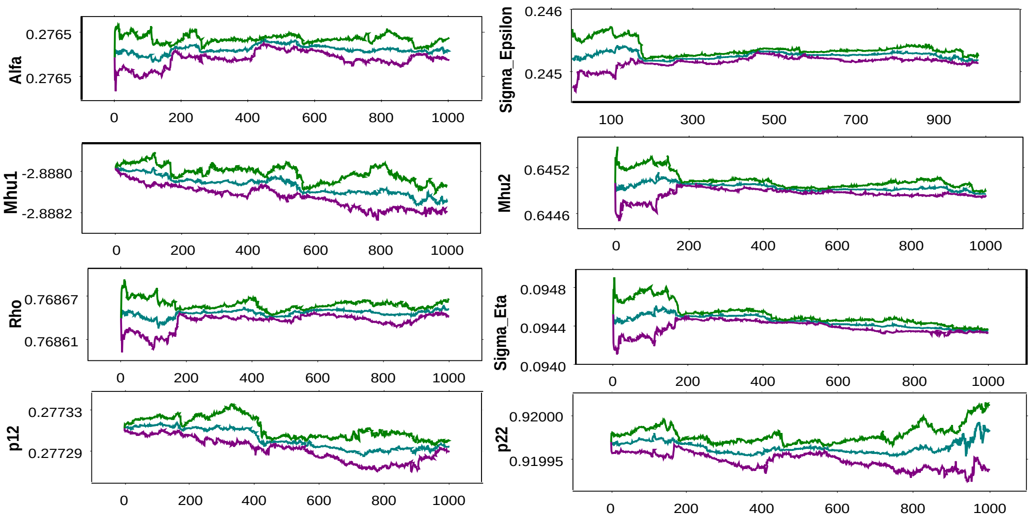

Figure 7.12: On-line parameter estimates. Graphs exhibit at each date the empirical mean and the quantiles at 0.275 and 0.925 for each parameter.

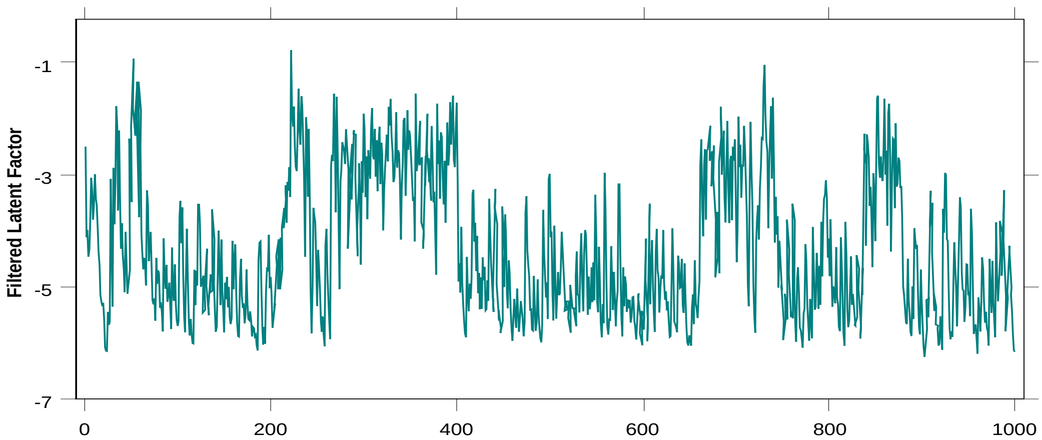

Figure 7.13: Sequentially filtered latent factor, x_t, over $T = 1,000$ observations.

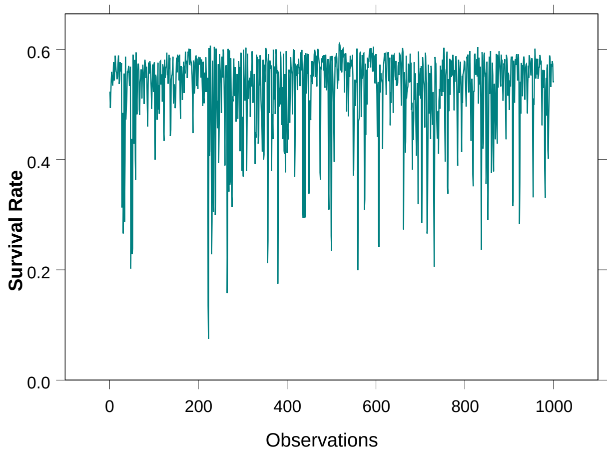

Figure 7.14: Survival Rate of the particle set at each time step.

In order to detect the absence of degeneracy in the output of the APF algorithm we evaluate at each time step the *Survival Rate*. It is defined as the number of particles survived to the selection step over the total number of particles. Particles set degenerates when persistently exhibiting a high number of dead particles from a generation to the subsequent one. Survival rate is calculated as follow

$$SR_t = \{N - \sum_{i=1}^{N} \mathbb{I}_{\{0\}}(Card(I_{i,t}))\}N^{-1} \tag{7.54}$$

where $I_{i,t} = \{j \in \{1, \ldots, N\} | i_t^j = i\}$ is the set of all random index values, which are selecting, at time t, the i-th particle. If at time t the particle k does not survive to the selection step then the set $I_{k,t}$ becomes empty. Graph 7.14 shows the survival rate at each time step. The rate does not decrease thus we conclude that the APF algorithm does not degeneracy in our study.

Appendix A - General Filtering

proof- **Recursive filtering relation given in Equation (7.24) -**
Consider the joint posterior density of the state vectors, conditional on the

information available at time T

$$\begin{aligned}
p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T},\theta) &= \frac{p(\mathbf{x}_{0:T},\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)} = \\
&= p(\mathbf{x}_{0:T-1}|\mathbf{y}_{1:T-1},\theta)\frac{p(\mathbf{x}_T,\mathbf{y}_T|\mathbf{x}_{0:T-1},\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)} = \\
&= p(\mathbf{x}_{0:T-1}|\mathbf{y}_{1:T-1},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_{0:T},\mathbf{y}_{1:T-1},\theta)p(\mathbf{x}_T|\mathbf{x}_{0:T-1},\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)} = \\
&= p(\mathbf{x}_{0:T-1}|\mathbf{y}_{1:T-1},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)p(\mathbf{x}_T|\mathbf{x}_{T-1},\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)} = \\
&= p(\mathbf{x}_{0:T-1}|\mathbf{y}_{1:T-1},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_T,\theta)p(\mathbf{x}_T|\mathbf{x}_{T-1},\theta)}{p(\mathbf{y}_T|\mathbf{y}_{1:T-1},\theta)}.
\end{aligned} \tag{7.55}$$

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

$\square$

proof- **Recursive smoothing density given in Equation (7.25) -** Consider the joint posterior density of the state vectors, conditional on the available information $\mathbf{y}_{1:T}$

$$\begin{aligned}
& p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T},\theta) = \\
=\; & p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{0:T-1}|\mathbf{x}_T,\mathbf{y}_{1:T},\theta) = \\
=\; & p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{0:T-2}|\mathbf{x}_{T-1:T},\mathbf{y}_{1:T},\theta) = \\
\overset{Bayes}{=}\; & p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_{T-1:T},\mathbf{y}_{1:T-1},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)}p(\mathbf{x}_{0:T-2}|\mathbf{x}_{T-1:T},\mathbf{y}_{1:T},\theta) = \\
\overset{Markov}{=}\; & p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)\frac{p(\mathbf{y}_T|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)}{p(\mathbf{y}_T|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)}p(\mathbf{x}_{0:T-2}|\mathbf{x}_{T-1:T},\mathbf{y}_{1:T},\theta) = \\
=\; & p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)p(\mathbf{x}_{0:T-2}|\mathbf{x}_{T-1},\mathbf{y}_{1:T},\theta).
\end{aligned} \tag{7.56}$$

By applying iteratively Bayes theorem and Markov property of the dynamic model we obtain the recursive smoothing relation

$$
\begin{aligned}
& p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T},\theta) = \\
= \;& p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)\prod_{t=0}^{T-2} p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:T},\theta) \\
\overset{(7.57)}{=} \;& p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)\prod_{t=0}^{T-2} p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\mathbf{y}_{t+1:T},\theta) \\
\overset{Bayes}{=} \;& p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)\prod_{t=0}^{T-2} \frac{p(\mathbf{y}_{t+1:T}|\mathbf{x}_{t:t+1},\mathbf{y}_{1:t},\theta)p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)}{p(\mathbf{y}_{t+1:T}|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)} \\
\overset{Markov}{=} \;& p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)p(\mathbf{x}_{T-1}|\mathbf{x}_T,\mathbf{y}_{1:T-1},\theta)\prod_{t=0}^{T-2} \frac{p(\mathbf{y}_{t+1:T}|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)}{p(\mathbf{y}_{t+1:T}|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta)} \\
= \;& p(\mathbf{x}_T|\mathbf{y}_{1:T},\theta)\prod_{t=0}^{T-1} p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t},\theta).
\end{aligned}
$$

$\square$

Appendix B - Simulation Based Filtering

proof **- Single-Move Gibbs Sampler, Equation (7.38) -**
Consider the full posterior density of the t-th state vector and apply the independence assumption between $\mathbf{y}_{t+1:T}$ and $\mathbf{x}_t$

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

We can simplify the last density as follow

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

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

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

□

proof- **Recursive weights updating relation given in Equation (7.49) -**

Starting from the definition of importance weights

$$
\begin{aligned}
w_{t+1} &\overset{\Delta}{=} \frac{p(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)}{q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)} = \\
&\overset{Bayes}{=} \frac{p(\mathbf{x}_{0:t+1},\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)}{q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)} = \\
&= \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)p(\mathbf{x}_{t+1},\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\mathbf{x}_{0:t},\theta)}{q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)} = \\
&= \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)}{q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)} \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{0:t+1},\mathbf{y}_{1:t},\theta)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)} p(\mathbf{x}_{t+1}|\mathbf{x}_{0:t},\mathbf{y}_{1:t},\theta) = \\
&\overset{Markov}{=} \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)}{q(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1},\theta)} \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)} p(\mathbf{x}_{t+1}|\mathbf{x}_{t},\theta) = \\
&= \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t},\theta)} \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)p(\mathbf{x}_{t+1}|\mathbf{x}_{t},\theta)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)q(\mathbf{x}_{t+1}|\mathbf{x}_{0:t},\mathbf{y}_{1:t+1},\theta)} = \\
&= w_t \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)p(\mathbf{x}_{t+1}|\mathbf{x}_{t},\theta)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t},\theta)q(\mathbf{x}_{t+1}|\mathbf{x}_{0:t},\mathbf{y}_{1:t+1},\theta)}.
\end{aligned}
\tag{7.58}
$$

Thus particle weights updating recursive relation is

$$
w_{t+1} \propto w_t \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta)p(\mathbf{x}_{t+1}|\mathbf{x}_{t},\theta)}{q(\mathbf{x}_{t+1}|\mathbf{x}_{t},\mathbf{y}_{t+1},\theta)}. \tag{7.59}
$$

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 (7.59) simplifies to

$$
w_{t+1} \propto w_t \, p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1},\theta). \tag{7.60}
$$

□

Bibliography

[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] 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.

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

[7] Casarin R., (2003), Bayesian Inference for Markov Switching Stochastic Volatility Models, *working paper*, CEREMADE, forthcoming.

[8] 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.

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

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

[11] Diebold F.X. and Rudebusch G.D., (1996), Measuring Business Cycles: A Modern Perspective, *The Review of Economics and Statistics*, 78, 67-77.

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

[13] 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.

[14] Durland J. and McCurdy T., (1994), Duration-dependent transitions in a markov model of u.s. gnp growth, *Journal of Business and Economic Statistics*, 12, 279-288.

[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] Goldfeld S.M. and Quandt R.E., (1973), A Markov Model for Switching Regression, *Journal of Econometrics*, 1, 3-16.

[18] 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.

[19] Hamilton J.D., (1989), A new approach to the economic analysis of nonstationary time series and the business cycle, *Econometrica*, 57, 357-384.

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

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

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

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

[24] 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.

[25] Kim C.J., (1994), Dynamic linear models with Markov switching, *Journal of Econometrics*, 60, 1-22.

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

[27] Kim C.J. and Murray C.J., (2001), Permanent and Transitory Components of Recessions, forthcoming, *Empirical Economics*.

[28] Kim C.J. and Piger J., (2000), Common stochastic trends, common cycles, and asymmetry in economic fluctuations, Working paper, n. 681, International Finance Division, Federal Reserve Board,Semptember 2000.

[29] 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.

[30] Krolzig H.M., (1997), *Markov Switching Vector Autoregressions. Modelling, Statistical Inference and Application to Business Cycle Analysis*, Springer-Verlag.

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

[32] 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.

[33] 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.

[34] 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.

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

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

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

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

[39] 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.

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

[41] Potter S.M., (1995), A Nonlinear Approach to U.S. GNP, *Journal of Applied Econometrics*, 10, 109-125.

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

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

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

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

[46] Sichel D.E., (1991), Business cycle duration dependence: A parametric approach, *Review of Economics and Statistics*, 73, 254-256.

[47] 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.

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

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

[50] Tong H., (1983), *Threshold Models in Non-Linear Time-Series Models*, New York, Springer-Verlag, 1983.

[51] Watson J., (1994), Business cycle durations and postwar stabilization of the u.s. economy, *American Economic Review*, 84, 24-46.

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

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

Chapter 8

Sequential Monte Carlo and Parameter Estimation

[1]Part of this work is in:

- Casarin, R. and Marin, J.-M., (2009), Online data processing: Comparison of Bayesian regularized particle filters, Electronic Journal of Statistics, 3, 239-258.
- Casarin, R. (2007), Simulation Methods for Bayesian Inference on Latent Variables Models, PhD Thesis, University Paris Dauphine