Online Bayesian learning in dynamic models: An illustrative introduction to particle methods*

Hedibert F. Lopes
University of Chicago

Carlos M. Carvalho
University of Texas at Austin

September 2, 2011

Abstract

This chapter reviews the main advances, over the last two decades, in the *particle filter* (PF) literature for dynamic models. We focus the discussion around the bootstrap filter (BF) and the auxiliary particle filter (APF), as these are the basis for most of the contributions in the literature. Both filters are then extended to accommodate sequential parameter learning, an area that has gained renewed attention over the last couple of years.

The chapter is mainly intended for those researchers and practitioners with little or no practical experience with PF and are looking for a hands-on approach to the subject. With that in mind, we implement and compare the discussed particle filters in two well known contexts: the AR(1) plus noise model and the stochastic volatility model with AR(1) dynamics, or simply SV-AR(1) model. The AR(1) plus noise model is used as a benchmark since all sequential distributions are available in closed-form when parameters are kept fixed. The SV-AR(1) provides an illustration of the ability of PF to deal with traditionally challenging non-linear models.

*Hedibert F. Lopes is Associate Professor of Econometrics and Statistics, Booth School of Business, University of Chicago, 5807 S. Woodlawn Ave, Chicago, IL 60637, `hlopes@chicagobooth.edu`. Carlos M. Carvalho is Assistant Professor of Statistics, IROM Department, University of Texas at Austin, `carlos.carvalho@mccombs.utexas.edu`.

Contents

1 Introduction

In this chapter, we provide an introductory step-by-step review of Monte Carlo methods for filtering in general nonlinear and non-Gaussian dynamic models, also known as state-space models or hidden Markov models (see West and Harrison, 1997, Durbin and Koopman, 2001, Cappé *et al.*, 2005, and Gamerman and Lopes, 2006). These MC methods are commonly referred to as *sequential Monte Carlo*, or simply *particle filters*. The standard Markovian dynamic model for observation y_t is

$$\begin{aligned} y_t &\sim f(y_t|x_t,\theta), & (1)\\ x_t &\sim g(x_t|x_{t-1},\theta), & (2) \end{aligned}$$

where, for $t = 1, \ldots, n$, x_t is the latent state of the dynamic system and θ is the set of fixed parameters defining the system. Equation (1) is referred to as the observation equation that relates the observed series y_t to the state vector x_t. Equation (2) is the state transition equation that governs the time evolution of the latent state. For didactical reasons, we assume throughout this chapter that y_t and x_t are both scalars. Multidimensional extensions are, in principle, straightforward and out of our scope.

The central problem in many state-space models, is the sequential derivation of the filtering distribution. By Bayes' theorem

$$p(x_t|y_{1:t},\theta) = \frac{f(y_t|x_t,\theta)p(x_t|y_{1:t-1},\theta)}{p(y_t|y_{1:t-1},\theta)} \tag{3}$$

where $y_{1:t} = (y_1, \ldots, y_t)$ (the same for $x_{1:t}$). The problem translates, in part, to deriving the prior distribution of the latent state x_t given data up to time $t-1$:

$$p(x_t|y_{1:t-1},\theta) = \int g(x_t|x_{t-1},\theta)p(x_{t-1}|y_{1:t-1},\theta)dx_{t-1}. \tag{4}$$

Even when θ is assumed to be known, sequential inference about x_t becomes analytically intractable, but when dealing with Gaussian dynamic linear models (DLM) (detailed in Section 2.1).

Most of the early contributions to the literature on the Bayesian estimation of state-space models boils down to the design of Markov chain Monte Carlo (MCMC) schemes that iteratively sample from states and parameters full conditional distributions:

$$p(x_{1:n}|y_{1:n},\theta) \quad \text{and} \quad p(\theta|x_{1:n},y_{1:n}). \tag{5}$$

The main references include, amongst others, Carlin, Polson and Stoffer (1992), Carter and Kohn (1994), Frühwirth-Schnatter (1994) and Gamerman (1998). See Migon *et al.* (2005) for a thorough review of dynamic models.

On the one hand, MCMC methods gave the researcher means to free herself from the (usually unrealistic) assumptions of normality and linearity for both observation equation (1) and state transition equation (2). On the other hand, however, they took from the researcher the ability to sequentially learn about states and parameters.

Particle filters are Monte Carlo schemes designed to sequentially approximate the densities in Equations (3) and (4) over time. The seminal *bootstrap filter* of Salmond, Gordon and Smith (1993), for example, uses the sampling importance resampling algorithm to first propagate particles from time $t-1$, i.e. draws from $p(x_{t-1}|y_{1:t-1})$, via Equation (4), and then resample the discrete set of propagated particles with weights proportional to the likelihood (Bayes' theorem from Equation (3)). Sections 3 and 4 provides additional details about the bootstrap filter as well as many other particles filters for state filtering or state and parameter learning.

The remainder of the chapter is organized as follows. Section 2 introduces the basic notation, results and references for the general class of Gaussian DLMs, the AR(1) plus noise model and for the standard stochastic volatility model with AR(1) dynamics. Particle filters for state learning with fixed parameters (aka pure filtering) and particle filters for state and parameter learning are discussed in Sections 3 and 4, respectively. Section 5 deals with general issues, such as MC error, sequential model checking, particle smoothing and the interaction between particle filters and MCMC schemes.

2 Dynamic models

In what follows we provide basic notation and results, as well as key references, for the general class of Gaussian DLMs, the AR(1) plus noise model and for the standard stochastic volatility model with AR(1) dynamics.

2.1 Dynamic linear models

A Gaussian *dynamic linear model* (DLM) can be written as

$$\begin{aligned} y_t|x_t,\theta &\sim N(\mu + F_t'x_t, \sigma_t^2) && (6)\\ x_t|x_{t-1},\theta &\sim N(\alpha + G_t x_{t-1}, \tau_t^2), && (7)\end{aligned}$$

where intercepts μ and α are added for notational reasons related the stochastic volatility model of Section 2.3. Conditionally on $\theta = (F_{1:n}, G_{1:n}, \sigma^2_{1:n}, \tau^2_{1:n}, \mu, \alpha)$ and assuming the initial distribution $(x_0|y_0) \sim N(m_0, C_0)$, it is straightforward to show that

$$\begin{aligned} x_t|y_{1:t-1},\theta &\sim N(a_t, R_t), && (8)\\ y_t|y_{1:t-1},\theta &\sim N(f_t, Q_t), && (9)\\ x_t|y_{1:t},\theta &\sim N(m_t, C_t), && (10)\end{aligned}$$

for $t = 1, \ldots, n$, where $N(a, b)$ denotes the normal distribution with mean a and variance b. The three densities in Equations (8) to (10) are referred to as the *propagation density*, the *predictive density* and the *filtering density*, respectively. In fact, the propagation and filtering densities are the prior density of x_t given $y_{1:t-1}$ and the posterior density of x_t given $y_{1:t}$. The means and variances of the three densities are provided by the *Kalman recursions*:

$$\begin{aligned} a_t &= \alpha + G_t m_{t-1} \text{ and } R_t = G_t C_{t-1} G_t' + \tau_t^2, && (11) \\ f_t &= \mu + F_t' a_t \text{ and } Q_t = F_t' R_t F_t + \sigma_t^2, && (12) \\ m_t &= a_t + A_t e_t \text{ and } C_t = R_t - A_t Q_t A_t', && (13) \end{aligned}$$

where $e_t = y_t - f_t$ is the prediction error and $A_t = R_t F_t Q_t^{-1}$ is the *Kalman gain*. Two other useful densities are the conditional and marginal *smoothed densities*

$$\begin{aligned} x_t | x_{t+1}, y_t, \theta &\sim N(h_t, H_t), && (14) \\ x_t | y_{1:n}, \theta &\sim N(m_t^n, C_t^n), && (15) \end{aligned}$$

where

$$\begin{aligned} h_t &= m_t + B_t(x_{t+1} - a_{t+1}) \text{ and } H_t = C_t - B_t R_{t+1} B_t', && (16) \\ m_t^n &= m_t + B_t(m_{t+1}^n - a_{t+1}) \text{ and } C_t^n = C_t + B_t^2(C_{t+1}^n - R_{t+1}), && (17) \end{aligned}$$

and $B_t = C_t G_{t+1}' R_{t+1}^{-1}$ (see West and Harrison, 1997, Chapter 4, for additional details).

2.2 AR(1) plus noise model

The AR(1) plus noise model is a Gaussian DLM where the state follows a standard AR(1) process and y_t is observed with measurement error:

$$\begin{aligned} y_t | x_t, \theta &\sim N(x_t, \sigma^2) && (18) \\ x_t | x_{t-1}, \theta &\sim N(\alpha + \beta x_{t-1}, \tau^2). && (19) \end{aligned}$$

Conditional on $\theta = (\sigma^2, \alpha, \beta, \tau^2)$, the whole state vector $x_{1:n}$ can be marginalized out analytically (see (9)):

$$p(y_{1:n}|\theta) = \prod_{t=1}^{n} p_N(y_t; f_t, Q_t), \qquad (20)$$

where $p_N(x; \mu, \sigma^2)$ is the density of a normal random variable with mean μ and variance σ^2 evaluated at x. Notice that here f_t and Q_t are both nonlinear functions of θ. The density in Equation (20) is commonly known as prior predictive density or integrated likelihood.

2.2.1 MC sampling from the posterior.

Posterior draws from $p(x_{1:n}, \theta|y_{1:n})$ can be directly and jointly obtained:

Step (i): Draw $\{\theta^{(i)}\}_{i=1}^N$ from $p(\theta|y_{1:n}) \propto p(\theta)p(y_{1:n}|\theta)$. The likelihood $p(y_{1:n}|\theta)$ comes from (20). This can be performed by sampling importance resampling, acceptance-rejection algorithm or Metropolis-Hastings-type algorithms.

Step (ii): Draw $x_{1:n}^{(i)}$ from $p(x_{1:n}|\theta^{(i)}, y_{1:n})$, for $i = 1, \ldots, N$, by first computing forward moments via Equations (11)-(13) and (16), and then sampling backwards x_t conditional on x_{t+1} and y_t via Equations (14). This step is known as the *forward filtering, backward sampling* (FFBS) algorithm (Carter and Kohn, 1994; Frühwirth-Schnatter, 1994).

Alternatively, θ from step (i) could be sampled, via a Gibbs sampler step, for instance, from $p(\theta|y_{1:n}, x_{1:n})$. In this case, iterating between steps (i) and (ii) would lead to a MCMC scheme whose target, stationary distribution is the posterior distribution $p(x_{1:n}, \theta|y_{1:n})$.

2.2.2 Prior specification and sufficient statistics.

Assume that the prior distribution of (α, β, τ^2) is decomposed into $\tau^2 \sim IG(\nu_0/2, \nu_0\tau_0^2/2)$ and $(\alpha, \beta)|\tau^2 \sim N(d_0, \tau^2 D_0)$, for known hyperparameters ν_0, τ_0^2, d_0 and D_0. It follows immediately, from basic Bayesian derivations for conditionally conjugate families, that $\tau^2|y_{1:t}, x_{1:t} \sim IG(\nu_t/2, \nu_t\tau_t^2/2)$ and $(\alpha, \beta)|\tau^2, y_{1:t}, x_{1:t} \sim N(d_t, \tau^2 D_t)$, where

$$\begin{aligned}
D_t^{-1} &= D_{t-1}^{-1} + z_t z_t' \\
D_t^{-1} d_t &= D_{t-1}^{-1} d_{t-1} + z_t x_t \\
\nu_t &= \nu_{t-1} + 1 \\
\nu_t \tau_t^2 &= \nu_{t-1}\tau_{t-1}^2 + (x_t - z_t' d_t)x_t + (d_{t-1} - d_t)' D_{t-1}^{-1} d_{t-1},
\end{aligned} \tag{21}$$

and $z_t = (1, x_t)'$. The relevance of these conditional conjugacy results will become apparent when dealing with some of the particles filter with parameter learning in Section 4. See Prado and Lopes (2011) for particle methods applied to AR models with structured priors.

2.3 SV-AR(1) model

Univariate stochastic volatility (SV) in asset price dynamics results from the movements of an equity index S_t and its stochastic volatility v_t via a continuous time diffusion by a Brownian motion: $d \log S_t = \mu dt + \sqrt{v_t} dB_t^P$ and $d \log v_t = \kappa(\gamma - \log v_t)dt + \tau dB_t^V$, where the parameters governing the volatility evolution are $(\mu, \kappa, \gamma, \tau)$ and (B_t^P, B_t^V) are (possibly correlated) Brownian motions (Rosenberg, 1972, Taylor, 1986, Hull and White, 1987, Ghysels, Harvey and Renault, 1996, Johannes and Polson, 2010).

Data arises in discrete time so it is natural to take an Euler discretization of the above equations. This is then commonly referred to as the *stochastic volatility autoregressive*, SV-AR(1), model and is described by the following non-linear dynamic model:

$$
\begin{aligned}
y_t|x_t, \theta &\sim N(0, \exp\{x_t/2\}) \qquad (22)\\
x_t|x_{t-1}, \theta &\sim N(\alpha + \beta x_{t-1}, \tau^2) \qquad (23)
\end{aligned}
$$

where y_t are log-returns and x_t are log-variances. See Jacquier, Polson and Rossi (1994) and Kim, Shephard and Chib (1998) for the original Bayesian papers on MCMC estimation of the above SV-AR(1) model. In addition, Lopes and Polson (2010b) provides an extensive review of Bayesian inference in the SV-AR(1) model, as well as other univariate and multivariate SV models.

2.3.1 Sampling parameters.

The SV model is completed with a conjugate prior distribution for $\theta = (\alpha, \beta, \tau^2)$, i.e. $p(\theta) = p(\alpha, \beta|\tau^2)p(\tau^2)$, where $(\alpha, \beta|\tau^2) \sim N(d_0, \tau^2 D_0)$ and $\tau^2 \sim IG(\nu_0/2, \nu_0\tau_0^2/2)$, for known hyperparameters d_0, D_0, ν_0 and τ_0^2. Apart from the nonlinear relationship between y_t and x_t in Equation (22), notice the similarity between the above SV-AR(1) model and the AR(1) plus noise model of Section 2.2. Therefore, sampling (α, β, τ^2) given $x_{1:t}$ can be done via Equations (21).

2.3.2 Sampling states.

Sampling from $x_{1:t}|y_{1:t}, \theta$ jointly is performed by a FFBS scheme introduced by Kim, Shephard and Chib (1998) for the SV-AR(1) model. They approximate the distribution of $\log y_t^2$ by a carefully tuned mixture of normals with seven components. More precisely, the observation equation (22) is rewritten by $z_t = \log y_t^2 = x_t + \epsilon_t$, where $\epsilon_t = \log \varepsilon_t^2$ follows a $\log \chi_1^2$ distribution, a parameter-free left skewed distribution with mean -1.27 and variance 4.94. They argue that $\epsilon = \log \chi_1^2$ can be well approximated by $\sum_{i=1}^7 \pi_i p_N(\epsilon_t; \mu_i, v_i^2)$, where

$$
\begin{aligned}
\pi &= (0.0073, 0.10556, 0.00002, 0.04395, 0.34001, 0.24566, 0.2575)\\
\mu &= (-11.40039, -5.24321, -9.83726, 1.50746, -0.65098, 0.52478, -2.35859)\\
v^2 &= (5.79596, 2.61369, 5.17950, 0.16735, 0.64009, 0.34023, 1.26261).
\end{aligned}
$$

Therefore, a standard data augmentation argument allows the mixture of normals to be transformed into individual normals, i.e. $(\epsilon_t|k_t) \sim N(\mu_{k_t}, v_{k_t}^2)$ and $Pr(k_t) = q_{k_t}$. Conditionally on $k_{1:t}$, the SV model can be rewritten as a standard Gaussian DLM:

$$
\begin{aligned}
(z_t|x_t, k_t, \theta) &\sim N(\mu_{k_t} + x_t, v_{k_t}^2) \qquad (24)\\
(x_t|x_{t-1}, \theta) &\sim N(\beta_0 + \beta_1 x_{t-1}, \tau^2). \qquad (25)
\end{aligned}
$$

The FFBS algorithm is then used to sample from $p(x_{1:n}|y_{1:n}, k_{1:n}, \theta)$. Given $x_{1:n}$, k_t is sampled from $\{1, \dots, 7\}$ with $Pr(\kappa_t = i|z_t) \propto \pi_i p_N(z_t; \mu_i + x_t, v_i^2)$, for $i = 1, \dots, 7$ and $t = 1, \dots, n$.

The above two steps, i.e. sampling parameters and sampling states, will be both very useful in the next two section when deriving particle filters for both state and fixed parameters.

3 Particle filters

Particle filters use Monte Carlo methods, mainly the sampling importance resampling (SIR), to sequentially reweigh and resample draws form the propagation density. The nonlinear Kalman filter is summarized by the prior and posterior densities in Equations (4) and (3):

$$\begin{aligned} p(x_t|y_{1:t-1}) &= \int g(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \\ p(x_t|y_{1:t}) &\propto f(y_t|x_t)p(x_t|y_{1:t-1}), \end{aligned}$$

where the vector of fixed parameters θ is assumed to be known and dropped from the notation and reappearing when necessary. The following joint densities will become useful in Sections 3.1 and 3.2:

$$\begin{aligned} p(x_t, x_{t-1}|y_{1:t-1}) &= g(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1}) \quad (26) \\ p(x_t, x_{t-1}|y_{1:t}) &\propto f(y_t|x_t)g(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1}). \quad (27) \end{aligned}$$

Particle filters, loosely speaking, combine the sequential estimation nature of Kalman-like filters with the flexibility for modeling of MCMC samplers, while avoiding some of their shortcomings. On the one hand, like MCMC samplers and unlike Kalman-like filters, particle filters are designed to allow for more flexible observational and evolutional dynamics and distributions. On the other hand, like Kalman-like filters and unlike MCMC samplers, particle filters provide online filtering and smoothing distributions of states and parameters. Advanced readers are refereed to, for instance, Cappé, Moulines and Rydén (2005, Chapters 7 to 9) for a more formal, theoretical discussions of sequential Monte Carlo methods.

The goal of most particle filters is to draw a set of i.i.d. particles $\{x_t^{(i)}\}_{i=1}^N$ that approximates $p(x_t|y_{1:t})$ by starting with a set of i.i.d. particles $\{x_{t-1}^{(i)}\}_{i=1}^N$ that approximates $p(x_{t-1}|y_{1:t-1})$. To simplify the notation, from now on we will simply refer to "particles x_{t-1}" when describing a "set of i.i.d. particles $\{x_{t-1}^{(i)}\}_{i=1}^N$". The most popular filters are the bootstrap filter (BF), also known as sequential importance sampling with resampling (SISR) filter, proposed by Gordon, Salmond and Smith (1993), and the auxiliary particle filter (APF), also known as auxiliary SIR (ASIR) filter, proposed by Pitt and Shephard (1999b). We introduce both of them in the next Section along with their optimal counterparts.

3.1 Bootstrap filter

The bootstrap filter (BF) is the seminal and perhaps the most implemented of the particle filters. It can be basically thought of as the repetition of the sampling importance resampling (SIR) over time. More precisely, let $p(x_{t-1}|y_{1:t-1})$ be the posterior density of the latent state x_{t-1} at time $t-1$. From Equations (4) and (3) and Bayes' theorem, it is easy to verify that

$$p(x_t, x_{t-1}|y_{1:t}) \propto \underbrace{f(y_t|x_t)}_{2.Resample} \underbrace{g(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})}_{1.Propagate}. \tag{28}$$

In words, BF combines old particles x_{t-1}, generated from $p(x_{t-1}|y_{1:t-1})$, and new particles x_t, generated from $g(x_t|x_{t-1})$, so the combined particles (x_t, x_{t-1}) are draws from $p(x_t, x_{t-1}|y_{1:t-1})$. This step is labeled "*1.Propagate*" in the above expression. BF then resamples the combined particles (x_t, x_{t-1}) with SIR weights proportional to the likelihood

$$\omega_t \propto \frac{f(y_t|x_t)g(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})}{g(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})} = f(y_t|x_t). \tag{29}$$

This step is labeled "*2.Resample*" in the above expression. These combined resampled particles approximate $p(x_{t-1}, x_t|y_{1:t})$ and, in particular, the marginal filtering density $p(x_t|y_{1:t})$. Figure 1 shows a diagram representation of the BF.

3.1.1 Particle impoverishment.

The overall SIR proposal density (the denominator of (29)) is $q(x_t, x_{t-1}|y_{1:t}) = p(x_t, x_{t-1}|y_{1:t-1}) = g(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})$. The particles x_t from (x_t, x_{t-1}) are, in fact, particles from the prior density $p(x_t|y_{1:t-1})$. It is well known that the SIR algorithm can perform badly when the prior is used as proposal density. The main reason is that in most cases either the prior is too flat relative to the likelihood or vice-versa. Small overlap between the prior and the posterior leads to unbalanced weights, that is a small number of particles will have dominating weights and all other particles will have negligible weights. This decrease in particle representativeness, or *particle degeneracy*, is exacerbated when the SIR is carried over time.

Figure 1 about here.

3.1.2 Adapted and fully adapted BF.

Instead of using the evolution density $g(x_t|x_{t-1})$ to propagate x_{t-1} to x_t, one could use an *unblinded proposal*, $q(x_t|x_{t-1}, y_t)$, i.e. a proposal that incorporates the information about the current observation y_t. These filters are commonly called *adapted filters*. In this case, $q(x_t, x_{t-1}|y_{1:t}) = q(x_t|x_{t-1}, y_t)p(x_{t-1}|y_{1:t-1})$ is the SIR proposal density, while the SIR weights are

$$\omega_t \propto \frac{f(y_t|x_t)g(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})}{q(x_t, x_{t-1}|y_{1:t})} = \frac{f(y_t|x_t)g(x_t|x_{t-1})}{q(x_t|x_{t-1}, y_t)}. \tag{30}$$

Fully adaptation occurs when one is able to sample from $p(x_t|x_{t-1}, y_t)$, in which case the SIR weights are proportional to the predictive density

$$\omega_t \propto p(y_t|x_{t-1}). \tag{31}$$

Even thought fully adaptation is rare, it can be used to guide the researcher in the selection of proposal densities $q(x_t|x_{t-1}, y_t)$. The closer $q(x_t|x_{t-1}, y_t)$ is to $p(x_t|x_{t-1}, y_t)$ the better. However, as Pitt and Shephard (1999) say, "even fully adapted particle filters do not produce iid samples from $p(x_t|y_{1:t})$, due to their approximation of $p(x_t|y_{1:t-1})$ by a finite mixture distribution." The AR(1) plus noise model of Section 2.2 and SV-AR(1) model of Section 2.3 can be implemented by fully adapted and adapted versions of the above BF.

3.2 Auxiliary particle filter

Pitt and Shephard (1999) noticed that writing Bayes' theorem from Equation (28) as

$$p(x_t, x_{t-1}|y_{1:t}) \quad \propto \quad \underbrace{p(x_t|x_{t-1}, y_{1:t})}_{2.Propagate} \underbrace{p(y_t|x_{t-1})p(x_{t-1}|y_{1:t-1})}_{1.Resample}, \tag{32}$$

would lead to alternative ways of designing the SIR proposal density $q(x_t, x_{t-1}|y_{1:t})$. Since $p(y_t|x_{t-1})$ and $p(x_t|x_{t-1}, y_{1:t})$ are usually, respectively, unavailable for point-wise evaluation and sampling (see the discussion about fully adapted filters at the end of Section 3.1), they suggested a generic proposal

$$q(x_{t-1}, x_t|y_{1:t}) = g(x_t|x_{t-1})f(y_t|h(x_{t-1}))p(x_{t-1}|y_{1:t-1}), \tag{33}$$

where $h(.)$ is usually the expected value, median or mode of $g(x_t|x_{t-1})$. The SIR weights would then be written as

$$w_t \propto \frac{f(y_t|x_t)g(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})}{g(x_t|x_{t-1})f(y_t|h(x_{t-1}))p(x_{t-1}|y_{1:t-1})} = \frac{f(y_t|x_t)}{f(y_t|h(x_{t-1}))}. \tag{34}$$

In words, APF would resample old particles x_{t-1} from $p(x_{t-1}|y_{1:t-1})$ with weights proportional to $f(y_t|h(x_{t-1}))$, which take into account the new observation y_t. These are usually called the *first-stage* weights. This step is labeled "*1.Resample*" in Equation (32). Then, new particles x_t are sampled from $g(x_t|x_{t-1})$, such that the combined particles (x_{t-1}, x_t) are draws from $q(x_{t-1}, x_t|y_{1:t})$. These combined particles are then resampled with weights given by Equation (34). These are usually called the *second-stage* weights. This step is labeled "*1.Propagate*" in Equation (32). The final, resampled combined particles approximate $p(x_{t-1}, x_t|y_{1:t})$ and, in particular, the marginal filtering density $p(x_t|y_{1:t})$. Comparing the above labels and the their order of operation, we call the APF a *resample-sample* filter, while the BF is *sample-resample* filter. Figure 2 shows a diagram representation of the bootstrap filter.

3.2.1 Fully adapted APF.

The above generic APF is a partially adapted filter by construction. However, the degree of adaptation depends on how close the first-stage weights $f(y_t|h(x_{t-1}))$ and the predictive $p(y_t|x_{t-1})$ are. For general adapted first-stage weights $q(x_{t-1}|y_t)$ and adapted resampling proposal $q(x_t|x_{t-1}, y_t)$, the SIR weights of Equation (34) become

$$w_t \propto \frac{f(y_t|x_t)g(x_t|x_{t-1})}{q(x_t|x_{t-1}, y_t)q(x_{t-1}|y_t)}. \tag{35}$$

Similar to the fully adapted BF, the APF is fully adapted when $q(x_{t-1}|y_t) = p(y_t|x_{t-1})$ and $q(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1}, y_t)$. In this case, the second-stage weights (Equation (35)) are proportional to one (no resampling necessary).

3.2.2 Local linearization.

Pitt and Shephard (1999) suggest, for more general settings, proposal density $q(x_t|x_{t-1}, y_t)$ that are based on local linearization of the observation equation via an extended Kalman filter-type approximation in order to better approximate $p(x_t|x_{t-1}, y_t)$. See Doucet, Godsill and Andrieu (2000) and Guo, Wang and Chen (2005), amongst others, for additional particle filters and discussion on proposals based on local linear approximations.

Another class of proposals, usually more efficient when available, is based on the *mixture Kalman filters* (MKF) of Chen and Liu (2000). The MKF takes advantage of possible analytical integration of some components of the state vector by conditioning on some other components. Such filters are commonly refereed to as *Rao-Blackwellized particle filter*. This is also acknowledged in Pitt and Shephard (1999) and many other references. See, for instance, Douc, Moulines and Olsson (2009) and Doucet and Johansen (2008) and Carvalho *et al.* (2010).

Figure 2 about here.

3.3 Marginal likelihood

The above filters can be used to approximate $p(y_{1:t})$, the marginal likelihood up to time t, as

$$\hat{p}(y_{1:t}) = \prod_{j=1}^{t} \hat{p}(y_j|y_{1:j-1}) = \frac{1}{N^t} \prod_{t=1}^{t} \sum_{i=1}^{N} f(y_j|x_j^{(i)}), \tag{36}$$

where x_t are particles from $p(x_t|y_{1:t-1})$. See Chopin (2002) and Del Moral *et al.* (2006) for further details and theoretical discussion.

3.4 Effective sample size

The quality of a particle filter can be measured by its ability to generate a "diverse" particle set by drawing from proposals $q(x_t|x_{t-1}, y_t)$ and reweighting with densities $q(x_{t-1}|y_t)$. Kong, Liu and Wong (1994) suggest using the coefficient of variation $CV_t = (N \sum_{i=1}^{N}(\omega_t^{(i)} - 1/N)^2)^{1/2}$, where $\omega_t^{(i)} = w_t^{(i)} / \sum_{j=1}^{N} w_t^{(j)}$ are normalized weights, as a simple criterion to detect the weight degeneracy phenomenon. CV_t varies between 0 (equal weights) and $\sqrt{N-1}$ (N copies of a single particle). Liu and Chen (1995) and Liu (1996) propose tracking the *effective sample size* $N_{\text{eff}} = N/(1+CV_t^2)$, which varies between 1 (N copies of a single particle) and N (equal weights). Cappé *et al.* (2005) tracks the Shannon entropy $\text{Ent} = -\sum_{i=1}^{N} \omega_t^{(i)} \log_2 \omega_t^{(i)}$, which varies between 0 (N copies of a single particle) and $\log_2 N$ (equal weights).

3.5 Examples

3.5.1 AR(1) plus noise model.

From Section 2.2 we can easily see that, given θ, the filtering densities $p(x_t|y_{1:t})$ are available in closed form and no particle filtering is necessary. However, we implement both BF and APF to this model and use the exact densities to assess their performances. It is easy to see that $p(y_t|x_{t-1})$ is normal with mean $h(x_{t-1}) = \alpha + \beta x_{t-1}$ and variance $\sigma^2 + \tau^2$, while $p(x_t|x_{t-1}, y_t)$ is normal with mean $Ay_t + (1 - A)h(x_{t-1})$ and variance $(1 - A)\tau^2$, where $A = \tau^2/(\tau^2 + \sigma^2)$. These results are used to implement fully adapted BF and APF, labeled here by OBF and OAPF (for optimal).

We simulate $S = 50$ data sets for each value of τ^2 in $\{0.05, 0.75, 1.0\}$ and all with $n = 100$ observations; a total of 150 data sets. The other parameters are $(\alpha, \beta, \sigma^2) = (0.05, 0.95, 1.0)$ and $x_0 = 1$. The prior for x_0 is $N(m_0, C_0)$ where $m_0 = 1$ and $C_0 = 10$. We run the four filters $R = 50$ times, each time based on $N = 500$ particles. A total of $150 \times 50 \times 4 = 30,000$ combined runs. We then compute the logarithm of the mean square error of filter f and time t as $MSE_{ft} = \sum_{s=1}^{S} \sum_{r}^{R} (\hat{q}_{sftr}^{\alpha} - q_{st}^{\alpha})^2/RS$, where q_{st}^{α} and q_{sftr}^{α} are the true and approximated αth percentile of $p(x_t|y_{1:t})$, for data set s, time period t, run r, percentile α in $\{5, 50, 95\}$ and filter f in $\{BF, APF, OBF, OAPF\}$.

Figure 3 summarizes our findings based on log relative MSEs of APF, OBF and OAPF relative to BF. It suggests that the optimal filters are better then their counterpart non-optimal filters. In addition, OAPF is uniformly superior to OBF (increasingly in τ^2), so favoring resampling-sampling filters over sampling-resampling filters. Finally, BF is usually better than APF for small τ^2/σ^2 (small signal-to-noise ratio). Similar results are found when MSEs are replaced by mean absolute errors (not shown here).

Figure 4 compares the performance of BF and OBF based on the three criteria introduced in Section 3.4 to monitor particle degeneracy. OBF is increasingly better (more balanced weights) than BF as τ^2 increases. Recall that BF propagates from $N(\alpha + \beta x_{t-1}, \tau^2)$ and

OBF propagates from $N(Ae_t + \alpha + \beta x_{t-1}, (1-A)\tau^2)$, where $e_t = y_t - (\alpha + \beta x_{t-1})$ and $A = \tau^2/(\tau^2 + \sigma^2)$. OBF approaches BF when A approaches zero, or when the signal-to-noise ratio approaches zero. These findings are even more pronounced for larger values of n or τ^2 or both (not known here).

Figures 3 and 4 about here.

3.5.2 SV-AR(1) model.

In this example we illustrate the performance of both bootstrap filter and the auxiliary particle filter for the SV-AR(1) model of Section 2.3. The parameter vector $\theta = (\alpha, \beta, \tau^2)$ is assumed known (see Section 4.4 for the general case where θ is also learned sequentially). Let $\mu_t = \alpha + \beta x_{t-1}$. On the one hand, the BF propagates new particles x_t from $N(\mu_t, \tau^2)$, which are then resampled with weights proportional to $p_N(y_t; 0, e^{x_t})$. On the other hand, the APF resamples old particles x_{t-1} with weights proportional to $p_N(y_t; 0, e^{\mu_t})$. New particles x_t are then propagated from $N(\mu_t, \tau^2)$ and resampled with weights proportional to $p_N(y_t; 0, e^{x_t})/p_N(y_t; 0, e^{\mu_t})$.

Potentially better proposals can be obtained. One could, for instance, use the (rough) normal approximation $N(-1.27, 4.94)$ to $\log y_t^2$ presented in Section 2.3. This linearization leads to first-stage weights $q(x_{t-1}|y_t) = p_N(z_t; \mu_t, 4.94)$, where $z_t = \log y_t^2 + 1.27$, while the resampling proposal $q(x_t|x_{t-1}, y_t)$ is normal with mean $v(z_t/4.94 + \mu_t/\tau^2)$ and variance $v = 1/(1/4.94 + 1/\tau^2)$. Consequently, it can be shown that the second-stage weights are proportional to $p_N(y_t; 0, \exp\{x_t\})/p_N(z_t; x_t, 4.94)$. We call this APF filter simply APF1 in what follows.

A second example is based on Kim, Shephard and Chib (1998). They used, in a MCMC context, a first order Taylor expansion of e^{-x_t} around μ_t to approximate the likelihood $p(y_t|x_t)$ by $\exp\{-0.5x_t(1 - y_t^2 e^{-\mu_t})\}$ (up to a proportionality constant). In this setting, the resampling proposal $q(x_t|x_{t-1}, y_t)$ is $N(\tilde{\mu}_t, \tau^2)$ with $\tilde{\mu}_t = \mu_t + 0.5\tau^2(y_t^2 e^{-\mu_t} - 1)$. First-stage weights are then $q(x_{t-1}|y_t) \propto \exp\{-0.5\tau^{-2}[(1 + \mu_t)\tau^2 y_t^2 e^{-\mu_t} + \mu_t^2 - \tilde{\mu}_t^2]\}$. We call this APF filter simply APF2 in what follows.

In a third, more involving example, inspired by Kim, Shephard and Chib (1998), who use a seven-component mixture of normals to approximate $\log \chi_1^2$ (see Equations (24) and (25) of Section 2.3), we obtain a fully adapted APF for the SV-AR(1) model. In this case, the first-stage weights are proportional to $\sum_{i=1}^7 \pi_i p_N(\log y_t^2; \mu_i + \alpha + \beta m_{t-1}, v_i + \tau^2 + \beta^2 C_{t-1})$, where m_{t-1} and C_{t-1} are the Kalman moments from Section 2.1. By integrating out both states x_t and x_{t-1}, we expect the above weights to be flatter, more evenly balanced than the respective ones based on the BF, APF, APF1 and APF2. In addition, instead of sampling x_t, we first sample κ_t from $\{1, \ldots, 7\}$ with $Pr(\kappa_t = i) \propto \pi_i N(\log y_t^2; \mu_i + \alpha + \beta m_{t-1}, v_i + \tau^2 + \beta^2 C_{t-1})$, for $i = 1, \ldots, 7$, and then update m_t and C_t via Equations (11) to (13) from Section 2.1. See the discussion in the last paragraph of Section 3.2. We call this APF filter simply FAAPF in what follows.

A total of $n = 200$ data points were simulated from $\alpha = -0.03052473$, $\beta = 0.9702$, $\tau^2 = 0.031684$ and $x_0 = -1.024320$. This is the specification used in one of simulated exercises from Pitt and Shephard (1999) and is chosen to mimic the time series behavior of financial returns. We assume that $x_0 \sim N(m_0, C_0)$ for $m_0 = -1.024320$ and $C_0 = 1$. We run the three filters for $R = 50$ times, each time and each one based on $N = 1000$ particles. We then compute their mean absolute error, $MAE = \sum_{t=1}^{n} |\hat{q}_{t,f}^{\alpha} - q_t^{\alpha}|/n$, where q_t^{α} and $q_{t,f}^{\alpha}$ are the true and approximated αth percentile of $p(x_t|y_{1:t})$, for $\alpha = (5, 50, 95)$ and f one of the filters.

Figure 5 summarizes our simulation exercise. The empirical findings suggest that the filters perform quite similarly, with the FAAPF, followed by the BF, being uniformly better than all other filters for all percentiles. This is probably partially due to the fact that the variability of the system equation ($\tau^2 = 0.02$) is much smaller than that of the observation equation. Recall, from Section 2.3, that the variance of the $\log \chi_1^2$ is around 4.94. In other words, $p_N(y_t; 0, \exp\{\alpha + \beta x_{t-1}\})$ does not seem to be a good SIR proposal for $p(y_t|x_{t-1})$. On one of their simulation exercises, Pitt and Shephard (1999) found similar results. They say that "the auxiliary particle filter is more efficient than the plain particle filter, but the difference is small, reflecting the fact that for the SV model, the conditional likelihood is not very sensitive to the state."

Figure 5 about here.

4 Parameter learning

The particle filters introduced in Section 3, and illustrated in the examples of Section 3.5, assumed that θ, the vector of parameters governing the both evolution and observation equations (see Equations 1 and 2), is known. This was partially for didactical or pedagogical reasons and partially to emphasize the chronological order of appearance of the filters. Sequential estimation of fixed parameters θ is historically and notoriously difficult. Simply including θ in the particle set is a natural but unsuccessful solution as the absence of a state evolution implies that we will be left with an ever-decreasing set of atoms in the particle approximation for $p(\theta|y_{1:t})$.

Important developments in the direction of sequentially updating $p(x_t, \theta|y_{1:t})$, instead of simply $p(x_t|y_{1:t}, \theta)$, have been made over the last decade and now sequential parameter learning is an important sub-area of research within the particle filter branch. Liu and West (2001), Storvik (2002), Fearnhead (2002), Polson, Stroud and Müller (2008) and Carvalho, Johannes, Lopes and Polson (2010) are a good representation of the rapid developments in this area. We revisit several of these contributions here along with illustrations of their implementation in the AR(1) plus noise and SV-AR(1) models.

4.1 Liu and West's filter

Liu and West (2001) adapt the generic APF of Section 3.2 to sequentially resample and propagate particles associated with x_t and θ simultaneously. More specifically, Equation (32) is rewritten as

$$p(x_t, x_{t-1}, \theta|y_{1:t}) \quad \propto \quad \underbrace{p(x_t, \theta|x_{t-1}, y_{1:t})}_{2.Propagate}\underbrace{p(y_t|x_{t-1}, \theta)p(x_{t-1}, \theta|y_{1:t-1})}_{1.Resample}. \tag{37}$$

Similar to the APF's generic proposal (Equation 33), Liu and West resample old particles (x_t, θ) with first-stage weights proportional to $p(y_t|h(x_{t-1}), m(\theta))$, with $h(\cdot)$ as before and $m(\theta) = a\theta + (1-a)\bar{\theta}$. Let $\tilde{\theta}$ and $\tilde{x}_{t-1}$ be the resampled particles. New particles θ are then propagated from the resampled particles via $N(m(\tilde{\theta}), h^2V)$, where $a^2 + h^2 = 1$, and new particles x_t are propagated from $g(x_t|\tilde{x}_{t-1}, \tilde{\theta})$. The second-stage weights are proportional to $p(y_t|x_t, \theta)/p(y_t|h(\tilde{x}_{t-1}), m(\tilde{\theta}))$. The quantities $\bar{\theta}$ and V are, respectively, the particle approximations to $E(\theta|y_{1:t})$ and $V(\theta|y_{1:t})$.

The key idea here is the choice of the proposal $q(x_t, \theta|x_{t-1}, y_{1:t})$ to approximate $p(x_t, \theta|x_{t-1}, y_{1:t})$. The proposal $q(x_t, \theta|x_{t-1}, y_{1:t})$ is decomposed into two parts: $q(x_t|\theta, x_{t-1}, y_{1:t}) = g(x_t|x_{t-1}, \theta)$ (blind propagation) and $q(\theta|x_{t-1}, y_{1:t})$, which is locally approximated by $N(m(\theta), h^2V)$. This *smooth kernel density* approximation (West 1993a,b) literally adds an artificial evolution to θ, as suggested in Gordon *et al.* (1993), but it controls the inherently over-dispersion by locally shrinking the particles θ towards their mean $\bar{\theta}$. Liu and West use standard discount factor ideas from basic dynamic linear models to select the tuning constant a (or h). The constants a and h measure, respectively, the extent of the shrinkage and the degree of over dispersion of the mixture. The rule of thumb is to select a greater than or equal to, say, 0.99. The idea is to use the mixture approximation to generate fresh samples from the current posterior in a attempt to avoid particle degeneracy.

The main attraction of Liu and West's filter is its generality as it can be implemented in any state-space model. It also takes advantage of APF's resample-propagate framework and can be considered a benchmark in the current literature. The steps of the LW algorithm are as follows:

Step 1 (Resample) $(\tilde{x}_{t-1}, \tilde{\theta})$ from (x_{t-1}, θ) with weights $w_t \propto p(y_t|h(x_{t-1}), m(\theta))$;

Step 2 (Propagate)

a) $\tilde{\theta}$ to $\hat{\theta}$ via $N(m(\tilde{\theta}), h^2V)$;

b) $\tilde{x}_{t-1}$ to $\hat{x}$ via $g(x_t|\tilde{x}_{t-1}, \hat{\theta})$;

Step 3 (Resample) (x_t, θ) from $(\hat{x}_t, \hat{\theta})$ with weights $w_{t+1} \propto p(y_t|\hat{x}_t, \hat{\theta})/p(y_t|h(\tilde{x}_{t-1}), m(\tilde{\theta}))$.

4.2 Storvik's Filter

Storvik (2002) (see also Fearnhead, 2002) proposes a particle filter that sequentially updates states and parameters by focusing on the particular case where the posterior distribution of θ given $x_{1:t}$ and $y_{1:t}$ depends on a low-dimensional set of sufficient statistics, i.e. $p(\theta|y_{1:t}, x_{1:t}) = p(\theta|s_t)$, that can be recursively and deterministically updated via $s_t = \mathcal{S}(s_{t-1}, x_{t-1}, x_t, y_t)$ (such as equation 21).

Both models we are using as illustrations in this chapter, i.e. the AR(1) plus noise and the SV-AR(1) models, allow sequential parameter learning via updating a set of sufficient statistics. Other, more general examples are the classe of conditionally Gaussian DLMs and the class discrete-state dynamic models, such as hidden Markov models (HMM), change-point models and generalized DLMs. The steps of the Storvik's algorithm are as follows:

Step 1 (Propagate) x_{t-1} to $\tilde{x}_t$ via $q(x_t|x_{t-1}, \theta, y_t)$;

Step 2 (Resample) (x_{t-1}, x_t, s_{t-1}) from $(x_{t-1}, \tilde{x}_t, s_{t-1})$ with weights $w_t \propto \frac{p(y_t|\tilde{x}_t,\theta)p(\tilde{x}_t|x_{t-1},\theta)}{q(\tilde{x}_t|x_{t-1},\theta,y_t)}$;

Step 3 (Propagate)

a) $s_t = \mathcal{S}(s_{t-1}, x_{t-1}, x_t, y_t)$;

b) θ from $p(\theta|s_t)$.

The resampling proposal density $q(x_t|x_{t-1}, \theta, y_t)$ plays the same role as it did in the BF and the APF.

4.3 Particle learning

Carvalho *et al.* (2010) present methods for sequential filtering, particle learning (PL) and smoothing for a rather general class of state space models. They extend Chen and Liu's (2000) mixture Kalman filter (MKF) methods by allowing parameter learning and utilize a resample-propagate algorithm together with a particle set that includes state sufficient statistics. They also show via several simulation studies that PL outperforms both the LW and Storvik filters and is comparable to MCMC samplers, even when full adaptation is considered. The advantage is even more pronounced for large values of n.

Let s_t^x denote state sufficient statistics satisfying deterministic updating rule $s_t^x = \mathcal{K}(s_{t-1}^x, \theta, y_t)$, for $\mathcal{K}(\cdot)$ mimicking the Kalman filter recursions of Section 2.1. The steps of a generic PL algorithm are as follows:

Step 1 (Resample) $(\tilde{\theta}, \tilde{s}_{t-1}^x, \tilde{s}_{t-1})$ from $(\theta, s_{t-1}^x, s_{t-1})$ with weights $w_t \propto p(y_t|s_{t-1}^x, \theta)$;

Step 2 (Propagate)

a) (x_{t-1}, x_t) from $p(x_{t-1}, x_t | s^x_{t-1}, \theta, y_t)$

b) $s_t = \mathcal{S}(\tilde{s}_{t-1}, x_{t-1}, x_t, y_t)$;

c) θ from $p(\theta | s_t)$;

d) $s^x_t = \mathcal{K}(\tilde{s}^x_{t-1}, \theta, y_t)$.

The reason for propagating x_{t-1} in step (2a) above, is that in the great majority of the dynamic models used in practice, $\mathcal{S}$ is a function of x_{t-1}, and possibly several other lags x_t. The AR(1) plus noise model of Section 2.2 and the SV-AR(1) model of Section 2.3 fall in this category. In addition, it is worth mentioning that (x_{t-1}, x_t) is discarded after s_t is propagate.

4.4 Examples

We illustrate the various particle filters with parameter learning via the AR(1) plus noise model and the SV-AR(1) model as before. Then, the SV-AR(1) model is generalized to accommodate Student's t errors (Section 4.4.3), leverage effects (Section 4.4.4) and Markov switching (Section 4.4.5).

4.4.1 AR(1) plus noise model.

We revisit the AR(1) plus noise model Equations (18) and (19) from Section 2.2, but now assuming that $(\sigma^2, \tau^2) = (1, 0.05)$ and that the goal is to sequentially approximate $p(x_t, \alpha, \beta | y_{1:t})$. The priors of (α, β) and x_0 are, respectively, $N(a_0, \tau^2 A_0)$ and $N(m_0, C_0)$ (see Section 2.2.2), while parameter sufficient statistics s_t are defined by the set of Equations (21). One data set with $n = 100$ observations is simulated from $(\alpha, \beta, x_0) = (0.05, 0.95, 1.0)$. The prior hyperparameters are $(m_0, C_0) = (1.0, 10)$, $a_0 = (0, 1)$ and $A_0 = 2I_2$.

Figure 6 shows the true contours of $p(\alpha, \beta | y_{1:n}) \propto p(\alpha, \beta) p(y_{1:n} | \alpha, \beta)$ on a grid for the pair (α, β) along with approximate contours ($N = 1000$ particles) based on a OAPF approximation to $p(y_{1:n} | \alpha, \beta)$ (Section 3.2 and Equation (20)). In practice, when (α, β) is replaced by larger parameter vectors, the use of grids could be replaced by a MCMC, SIR or rejection step. One can argue that approximating $p(y_{1:n} | \theta)$ by particle filters should be done with caution (see Pitt, 2002, and, more recently, Malik and Pitt, 2011, for further discussion).

Figure 7 compares the performance of the LW filter (with $a = 0.995$) and PL to sequential (brute force) MCMC. The MCMC for this model is outlined in Section 2.2.1 and is run for 2000 iterations with the second half used for posterior summaries. The LW filter starts to show particle degeneracy around the 50th observation and moves away from the true percentiles. Finally, Figure 8 compares the performance of the LW filter, Storvik's filter and PL for one data set. Notice that here LW also takes advantage of fully adaptation, so the only different between LW and PL is the handling of fixed parameters. Similarly, the main difference between Storvik's filter and PL is that Storvik's filter propagates first and then resamples, while PL resamples first and then propagates. As expected, both Storvik's filter and PL are significantly better than the "improved" LW filter and PL is slightly better than Storvik's, particularly when dealing with the latent state x_t and the parameter β and more so when approximating the tails of the filtering distributions.

Figures 6 to 8 about here.

4.4.2 SV-AR(1) model.

We revisit the SV-AR(1) plus noise model Equations (22) and (23) from Section 2.3, but now assuming that $(\alpha, \beta|\tau^2)$, τ^2 and x_0 are, respectively, $N(a_0, \tau^2 A_0)$, $IG(\nu_0/2, \nu_0\tau_0^2/2)$ and $N(m_0, C_0)$. As in the illustration of Section 3.5.2, a total of $n = 200$ data points were simulated from $\alpha = -0.03$, $\beta = 0.97$, $\tau^2 = 0.03$ and $x_0 = -0.1$. We assume, as before, that $(m_0, C_0) = (-0.1, 1)$. The other hyperparameters are $a_0 = (-0.03, 0.97)$, $A_0 = 1.6I_2$ and $(\nu_0, \tau_0^2) = (10, 0.04)$.

The LW filter is based on 500000 particles, while PL is based on 50000. MCMC for the model (see Sections (2.3.1) and (2.3.2)) is implemented over time for comparison with both LW filter and PL. MCMC, which starts at the true values, is based on 10000 draws after the same number of draws is discarded as burn-in. Figures 9 and 10 summarize the results. PL and MCMC produce fairly similar results, with LW slightly worse. Notice that LW is based on 10 times more particles than PL. We compared LW, PL and MCMC runs to a fine grid approximation of $p(\alpha, \beta, \tau^2|y_{1:n})$, with a 100-point grid for the log-volatilities x_t in $(-5, 2)$ and 50-point grids in the intervals $(-0.15, 0.1)$, $(0.85, 1.05)$ and $(0.01, 0.15)$, for α, β and τ^2, respectively. In this case, both LW and PL are based on 20000 particles and MCMC is based on 20000 draws after the same number of draws is discarded as burn-in. Figure 11 shows that PL and MCMC both approximate the true distributions quite well. LW underestimates all three parameters.

Figures 9 and 11 about here.

Figure 12 summarizes the $R = 10$ replications of LW and PL, both based on $N = 10000$ particles. LW has larger Monte Carlo error when approximation the filtering distributions for all quantities, with particular emphasis on the volatility of the log-volatility τ^2 and, consequently, on the latent state x_t. Based on this simple exercise and running our code in R, it takes about 7 and 15 minutes to run the LW filter and PL, respectively. It takes about 8 minutes to run MCMC based on the whole time series of $n = 200$ observations. It takes about 13 hours to run MCMC based on $y_{1:t}$ for all $t \in \{1, \ldots, 200\}$, i.e. 50 times slower than PL and 100 times slower than the LW filter.

Figure 12 about here.

4.4.3 SV-AR(1) model with t errors.

In order to illustrate particle filters' ability to approximate the predictive density $p(y_t|y_{1:t-1})$ via Equation 36 from Section 3.3, we implement PL for the SV-AR(1) model and the SV-AR(1) model with Student's t error as in Lopes and Polson (2011). Figure 13 presents data simulated from the SV-AR(1) model with errors following t_ν for $\nu \in \{1, 2, 4, 30\}$. Notice that the number of potential outliers decrease as ν increases with $\nu = 30$ approaching normality. Figure 14 compares the Bayes factors (in the log scale) of the t_ν models against normality. For instance, when the data is t_1 or t_2, each additional outlier makes Bayes factors support t models more significantly. For additional discussion on sequential model comparison and model checking via particle methods see, for instance, Carvalho *et al.* (2010) and Lopes *et al.* (2011).

Figures 13 and 14 about here.

4.4.4 SV-AR(1) model with leverage.

Omori, Chib, Shephard and Nakajima (2009) introduce MCMC for posterior inference in the SV-AR(1) model with leverage. More precisely, log-volatility dynamics (Equation (23)) is now $x_t|x_{t-1}, \theta \sim N(\alpha + \beta x_{t-1} + \tau\rho y_{t-1} \exp\{-x_{t-1}/2\}, \tau^2(1 - \rho^2))$. Negative ρ captures the increase in (log-)volatility x_t that follows a drop in y_{t-1}. One of their examples, where $(\alpha, \beta, \tau^2, \rho) = (-0.026, 0.97, 0.0225, -0.3)$, is revisited here based on $n = 10000$ observations (they use only $n = 1000$) in order to illustrate how a simple, generic LW filter performs relatively well even when the sample size is fairly large. We use their prior specification, $(\beta + 1)/2 \sim Beta(20, 1.5)$, $\alpha|\beta \sim N(0, (1 - \beta)^2)$, $\rho \sim U(-1, 1)$, and $\tau^2 \sim IG(5/2, 0.05/2)$, and run the LW filter based on $N = 500000$ particles and tuning parameter $a = 0.995$. Figure 15 summarizes the results. This LW filter could be easily extended to fit the other SV models they considered, such as the SV-t model (see Section 4.4.3) and the superposition models.

Figure 15 about here.

4.4.5 SV-AR(1) model with regime switching.

Carvalho and Lopes (2007) implements the LW filter for SV-AR(1) models with regime switching, where Equation (23) becomes $x_t|x_{t-1}, s_t, \theta \sim N(\alpha + \beta x_{t-1} + \gamma s_t, \tau^2)$, for $\gamma > 0$ and latent regime switching variable $s_t \in \{0, 1\}$. We assume, for simplicity, that s_t obeys a two-regime homogeneous Markov model with $Pr(s_t = 0|s_{t-1} = 0) = p$ and $Pr(s_t = 1|s_{t-1} = 1) = q$. The vector of fixed parameters is $\theta = (\alpha, \beta, \tau^2, p, q)$ and the vector of latent states is

(x_t, s_t). We revisit their analysis of the IBOVESPA stock index (São Paulo Stock Exchange) but with a larger data set spanning from 01/02/1997 to 08/08/2011 ($n = 3612$ observations). The prior hyperparameters (Section 2.2.2) are $d_0 = (-0.25, 0.95, 0.05)$, $D_0 = 6I_3$, $\nu_0 = 10$ and $\tau_0^2 = 0.05$, with $p \sim Beta(50, 1)$, $q \sim Beta(1, 50)$, $x_0 \sim N(0, 1)$ and $s_0 \sim Ber(0.1)$. Figures 16 and 17 summarize our findings. The model with regime switching captured the major 1997-1999 crisis listed in Carvalho and Lopes (2007) as well as the more recent credit crunch crisis of 2008. It also captured the sharp drop on Monday, August 8th 2011, when the IBOVESPA (and most financial markets worldwide) suffered a 8% fall following worries about the weak U.S. economy and the high levels of public debt in Europe. See Lopes and Polson (2010a) and Rios and Lopes (2011) for further discussion and illustrations of particle methods in SV-AR(1) models with regime switching.

Figures 16 and 17 about here.

4.4.6 SV-AR(1) model with realized volatility.

In this final illustration, we revisit Takahashi *et al.* (2009) who estimate SV models using daily returns and realized volatility simultaneously. Their most general model assumes that returns $y_{1t} \sim N(0, \exp\{x_t/2\})$ and that the log-volatility dynamics is $x_t|x_{t-1}, \theta \sim N(\alpha + \beta x_{t-1} + \tau\rho y_{1,t-1}\exp\{-x_{t-1}/2\}, \tau^2(1 - \rho^2))$ (as in Section 4.4.4). The model is completed with realized volatility $y_{2t} \sim N(\xi + x_t, \sigma^2)$, where ξ is the bias-correction term. We use high frequency data of Tokyo price index (TOPIX) that what was kindly share with us the authors for this illustration. In what follows y_{2t} is the logarithm of the scaled realized volatility based on one-minute intraday returns when the market is open during the 10-year period from April 1st, 1996 to March 31st, 2005 ($n = 2216$ trading days). Therefore, the vector of static parameters of the model is $\theta = (\alpha, \beta, \tau^2, \rho, \xi, \sigma^2)$. The implementation of the LW filter is fairly simple and we fit four models to the data: RV model, SV-AR(1) model, SV-AR(1) model with leverage and the current model. The RV model is basically an AR(1) plus noise model, in which case $\xi = 0$ for identification reasons. We label these four models RV, SV, ASV and ASV-RVC in what follows. The number of particles is $N = 100000$ and LW's tuning parameter is $a = 0.995$. Figure 18 shows posterior medians for time-varying standard deviations and their logarithms. The ASV model seems to be less sensitive to extremes when compared to the SV model. One can argue that the RV model is too adaptive when compared to the SV model. Similarly, the ASV-RVC is less sensitive to extremes when compared to the ASV model, while being less adaptive than the RV model. These results are corroborated by the marginal posterior densities for the models' parameters in Figures 19 and 20. The persistence parameter β and the leverage parameter ρ are smaller in the ASV-RVC model. In addition, both parameters ξ and σ^2 are away from zero, suggesting that the biased-corrected realized volatility helps estimating daily log-volatilities x_t.

Figures 18 to 20 about here.

5 Discussion

This chapter reviews many of the important advances in the particle filter literature over the last two decades. Two relatively simple but fairly general models are used to guide the review: the AR(1) plus noise model and the SV-AR(1) model. We aim at a broad audience of researchers and practitioners and illustrate the benefits and the limitations of particle filters when estimating with dynamic models where sequentially learning of latent states and fixed parameters is the primary interest.

The applications of Section 4.4 based on the several (important) stochastic volatility models was intended to illustrate to the reader how relatively complex (despite univariate) models can be sequentially estimated via particle filters at relatively low computational cost. They are comparable in performance to the standard MCMC proposed in the references listed in each one of the examples. It is important to emphasize that this cost increases with the dimension of both latent state and static parameter vectors and that this is one of the leading sub areas of current theoretical and empirical research.

There are currently several review papers, chapter and books the reader should read after becoming fluent with the tools we introduce here. Amongst those are the earlier papers by Doucet, Godsill and Andrieu (2000), Arulampalam, Maskell, Gordon and Clapp (2002) and Crisan and Doucet (2002), books by Liu (2001), Doucet, De Freitas and Gordon (2001) and Ristic, Arulampalam and Gordon (2004) and the 2002 special issue of IEEE Transactions on Signal Processing on sequential Monte Carlo methods. See also the review by Chen (2003).

More recent reviews are Cappé, Godsill and Moulines (2007), Doucet and Johansen (2009), Prado and West (2010, chapter 6) and Lopes and Tsay (2011). They carefully organize and highlight the fast development of the field over the last decade, such as parameter learning, more efficient particle smoothers, particle filters for highly dimensional dynamic systems and, perhaps the most recent one, the interconnections between MCMC and SMC methods.

Many important topics and issues were left out. Particle smoothers, for instance, are becoming a realistic alternative to MCMC in dynamic systems when the smoothed $p(x_{1:t}|y_{1:t})$, or simply $p(x_t|y_{1:t})$, is the distribution of interest. See Godsill, Doucet and West (2004), Fearnhead, Wyncoll and Tawn (2010), Douc, Garivier, Moulines and Olsson (2009) and Briers, Doucet and Maskell (2010), amongst others.

The interface between PF and MCMC methods is illustrated in our examples (see, for example, Section 3.3 and Figure 20). Hybrid schemes that combine particle methods and MCMC methods are abundant. Gilks and Berzuini (2001) and Polson, Stroud and Müller (2008), for instance, use MCMC steps to sample and replenish static parameters in dynamic systems. Andrieu, Doucet and Holenstein (2010) introduce particle MCMC methods to efficiently construct proposal distributions in high dimension via SMC methods. See also Pitt *et al.* (2011).

Finally, particle filters have recently received a lot of attention in estimating non-dynamic models such as mixtures, gaussian processes, tree models, etc. Important references are

Lopes *et al.* (2011) and Carvalho, Lopes, Polson and Taddy (2010).

Reference

1. Andrieu, C., Doucet, A. and Holenstein, R. (2010). Particle Markov chain Monte Carlo (with discussion) *Journal of the Royal Statistical Society, Series B*, 72, 269-342.

2. Andrieu, C., Doucet, A. and Tadić, V. B. (2005). On-line parameter estimation in general state-space models. In *Proceedings of the 44th Conference on Decision and Control*, 332-37. Institute of Electrical and Electronics Engineers.

3. Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T. (2002). A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50, 174-188.

4. Briers, M., Doucet, A. and Maskell, S. (2010). *Annals of the Institute of Statistical Mathematics* Smoothing algorithms for state-space models, 6261-89.

5. Cappé, O., Godsill, S. and Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proceedings in Signal Processing*, 95, 899-924.

6. Cappé, O., Moulines, E. and Rydén, T. (2005) *Inference in Hidden Markov Models.* Springer, New York.

7. Carlin, B. P., Polson, N. G. and Stoffer, D. S. (1992). A Monte Carlo approach to nonnormal and nonlinear state-space modeling. *Journal of the American Statistical Association*, 87, 493-500.

8. Carter, C. K. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, 81, 541-53.

9. Carvalho, C. M., Johannes, M., Lopes, H. F. and Polson, N. (2010). Particle learning and smoothing. *Statistical Science*, 25, 88-106

10. Carvalho, C. M., Lopes, H.F., Polson, N. and Taddy, M. (2010). Particle learning for general mixtures. *Bayesian Analysis*, 5.

11. Carvalho, C. M. and Lopes, H. F. (2007). Simulation-based sequential analysis of Markov switching stochastic volatility models. *Computational Statistics & Data Analysis*, 51, 4526-4542.

12. Chen, R. and Liu, J. S. (2000). Mixture Kalman filter. *Journal of the Royal Statistical Society, Series B*, 62, 493-508.

13. Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89, 539-52.

14. Crisan, D. and Doucet, A. (2002) A survey of convergence results on particle filtering methods for practitioners. *IEEE Transations on Signal Processing*, 50, 736-746.

15. Del Moral, P., Doucet, A. and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B*, 68, 411-436.

16. Douc, R., Garivier, E. Moulines, E. and Olsson, J. (2009). On the forward filtering backward smoothing particle approximations of the smoothing distribution in general state space models. *Annals of Applied Probability (to appear).*

17. Douc, R., E. Moulines, and J. Olsson (2009). Optimality of the auxiliary particle filter. *Probability and Mathematical Statistics*, 29, 1-28.

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

19. Doucet, A. and Johansen, A. (2008). A Note on Auxiliary Particle Filters. *Statistics & Probability Letters*, 78, 1498-1504.

20. Doucet, A. and Johansen, A. (2009). A Tutorial on Particle Filtering and Smoothing: Fifteen years Later. In D. Crisan and B. Rozovsky, editors, *Handbook of Nonlinear Filtering*. Oxford: Oxford University Press.

21. Doucet, A. and Tadić, V. B. (2003). Parameter estimation in general state-space models using particle methods. *Annals of the Institute of Statistical Mathematics*, 55, 409-22.

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

23. Fearnhead, P. (2002). Markov chain Monte Carlo, sufficient statistics and particle filter. *Journal of Computational and Graphical Statistics*, 11, 848-62.

24. Fearnhead, P., Wyncoll, D. and Tawn, J. (2010). A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97, 447-464.

25. Fernández-Villaverde, J. and Rubio-Ramírez, J. F. (2005). Estimating dynamic equilibrium economies: linear versus nonlinear likelihood. *Journal of Applied Econometrics*, 20, 891-910.

26. Fernández-Villaverde, J. and Rubio-Ramírez, J. F. (2007). Estimating macroeconomic models: A likelihood approach. *Review of Economic Studies*, 74, 1059-87.

27. Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15, 183-202.

28. Gamerman, D. (1998) Markov Chain Monte Carlo for dynamic generalized linear models. *Biometrika*, 85, 215-27.

29. Gamerman, D. and Lopes, H. F. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman & Hall/CRC.

30. Ghysels, E., Harvey, A. C. and Renault, E. (1996) Stochastic Volatility. In C. R. Rao and G. S. Maddala (eds) *Handbook of Statistics: Statistical Methods in Finance*, 119-191. (Amsterdam: North-Holland).

31. Gilks W.R. and Berzuini C. (2001). Following a moving target-MonteCarlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society, Series B*, 63, 127-46.

32. Godsill, S. J., Doucet, A. and West, M. (2004). Monte Carlo smoothing for non-linear time series. *Journal of the American Statistical Association*, 50, 438-449.

33. Gordon, N., Salmond, D. and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F. Radar Signal Process*, 140, 107-113.

34. Guo, D., Wang, X. and Chen, R. (2005). New sequential Monte Carlo methods for nonlinear dynamic systems. *Statistics and Computing*, 15, 135-47.

35. Hull, J., and White, A. (1987). The Pricing of Options on Assets with Stochastic Volatilities. *Journal of Finance*, 42, 281-300.

36. Jacquier, E., Polson, N. G. and Rossi. P. E. (1994). Bayesian analysis of stochastic volatility models. *Journal of Business and Economic Statistics*, 20, 69-87.

37. Kim, S., Shephard, N. and Chib, S. (1998). Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models. *Review of Economic Studies*, 65, 361-393.

38. Kong, A., Liu, J. S. and Wong, W. (1994). Sequential imputation and Bayesian missing data problems. *Journal of the American Statistical Association*, 89, 590-99.

39. Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing.* New York: Springer-Verlag.

40. Liu, J. and Chen, R. (1995). Blind Deconvolution via Sequential Imputations. *Journal of the American Statistical Association*, 90, 567-76.

41. Liu, J. and West, M. (2001). Combined parameters and state estimation in simulation-based filtering. In A. Doucet, N. de Freitas and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice.* New York: Springer-Verlag.

42. Lopes, H. F., Carvalho, C. M., Johannes, M. and Polson, N. G. (2011). Particle learning for sequential Bayesian computation (with discussion). In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. West, editors, *Bayesian Statistics 9*. Oxford: Oxford University Press, 317-360.

43. Lopes, H. F. and Polson, N. G. (2010a). Extracting SP500 and NASDAQ volatility: The credit crisis of 2007-2008. In A. OHagan and M. West, editors, *The Oxford Handbook of Applied Bayesian Analysis*. Oxford: Oxford University Press, 319-42.

44. Lopes, H. F. and Polson, N. G. (2010b). Bayesian inference for stochastic volatility modeling. In Bocker, K. (Ed.) *Rethinking Risk Measurement and Reporting: Uncertainty, Bayesian Analysis and Expert Judgement*, 515-551.

45. Lopes, H. F. and Polson, N. G. (2011) Particle Learning for Fat-tailed Distributions. *Working Paper*, The University of Chicago Booth School of Business.

46. Lopes, H. F. and Tsay, R. (2011). Particle filters and Bayesian inference in financial econometrics. *Journal of Forecasting*, 30, 168-209.

47. Malik, S. and Pitt, M. K. (2011) Particle filters for continuous likelihood evaluation and maximisation. *Journal of Econometrics* (in press).

48. Migon, H. S., Gamerman, D., Lopes, H. F. and Ferreira, M. A. R. (2005). Dynamic models. In Dey, D. and Rao, C.R. (Eds.) *Handbook of Statistics, Volume 25: Bayesian Thinking, Modeling and Computation*, 553-588.

49. Omori, Y., Chib, S., Shephard, N. and Nakajima, J. (2009) Stochastic volatility with leverage: Fast and efficient likelihood inference. *Journal of Econometrics*, 140, 425-449.

50. Olsson, J., Cappé, O., Douc, R. and Moulines, E. (2008). Sequential Monte Carlo smoothing with application to parameter estimation in non-linear state space models. *Bernoulli*, 14, 155-79.

51. Pitt, M. K. (2002) Smooth particle filters for likelihood evaluation and maximisation. Technical Report Department of Economics, University of Warwick.

52. Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94, 590-99.

53. Pitt, M. K., Silva, R. S., Giordani, P. and Kohn, R. (2011) On some properties of Markov chain Monte Carlo simulation methods based on the particle filter.

54. Polson, N. G., Stroud, J. R. and Müller, P. (2008). Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society, Series B*, 70, 413-28.

55. Poyiadjis, G., Doucet, A. and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98, 65-80.

56. Prado, R. and Lopes, H. F. (2009). Sequential parameter learning and filtering in structured AR models. *Working Paper*, The University of Chicago Booth School of Business.

57. Prado, R. and West, M. (2010). *Time Series: Modelling, Computation and Inference.* Chapman & Hall/CRC, The Taylor Francis Group.

58. Rios, M. P. and Lopes, H. F. (2011). Sequential parameter estimation in stochastic volatility models. *Working Paper.* The University of Chicago Booth School of Business.

59. Ristic, B., Arulampalam, S. and Gordon, N. (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications.* Artech House Radar Library.

60. Rosenberg, B. (1972). The Behaviour of Random Variables with Nonstationary Variance and the Distribution Of Security Prices. Working Paper No. 11, University of California, Berkeley, Institute of Business and Economic Research, Graduate School of Business Administration, Research Programme in Finance.

61. Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, 50, 281-89.

62. Takahashi, M., Omori, Y. and Watanabe, T. (2009) Estimating stochastic volatility models using daily returns and realized volatility simultaneously. *Computational Statistics and Data Analysis*, 53, 2404-2426.

63. Taylor, S. J. (1986). *Modelling Financial Time Series.* New York: JohnWiley and Sons.

64. West, W. (1993a). Approximating posterior distributions by mixtures. *Journal of the Royal Statistical Society, Series B*, 54, 553-68.

65. West, M. (1993b). Mixture models, Monte Carlo, Bayesian updating and dynamic models. *Computing Science and Statistics*, 24, 325-33.

66. West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models, 2nd Edition.* New York: Springer.

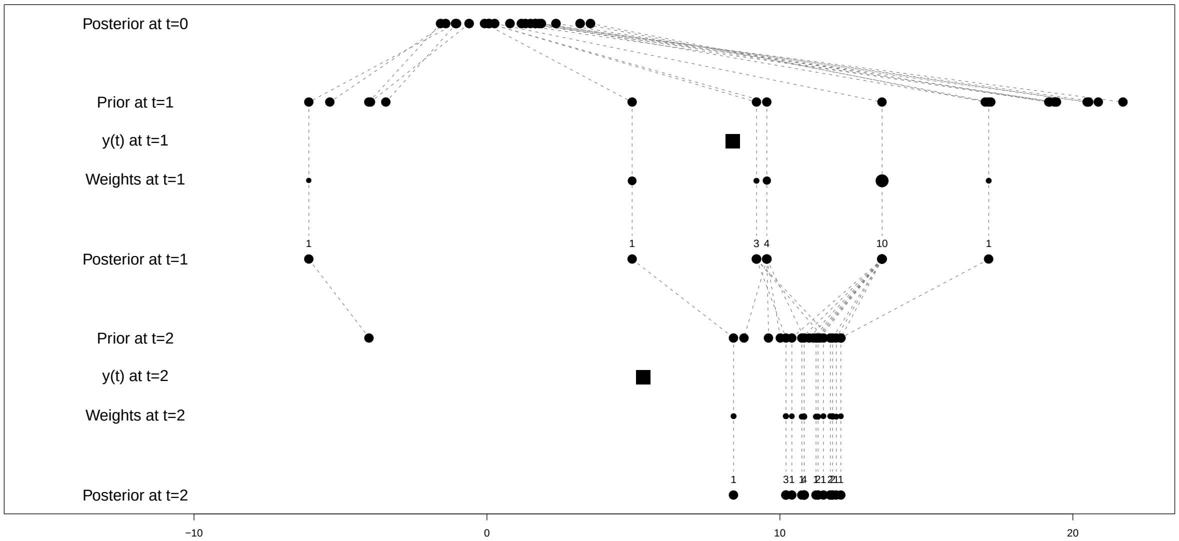

Figure 1: Schematization of the bootstrap filter. Based on $y_1 = 8.4$ and $y_2 = 5.3$ and the following model: Initial distribution: $x_0 \sim N(0, 2)$, evolution equation: $x_t|x_{t-1} \sim N(0.5x_{t-1} + 25x_{t-1}/(1 + x_{t-1}^2) + 8cos(1.2(t - 1)), 10)$, and observation equation: $y_t|x_t \sim N(0.05x_t^2, 1)$.

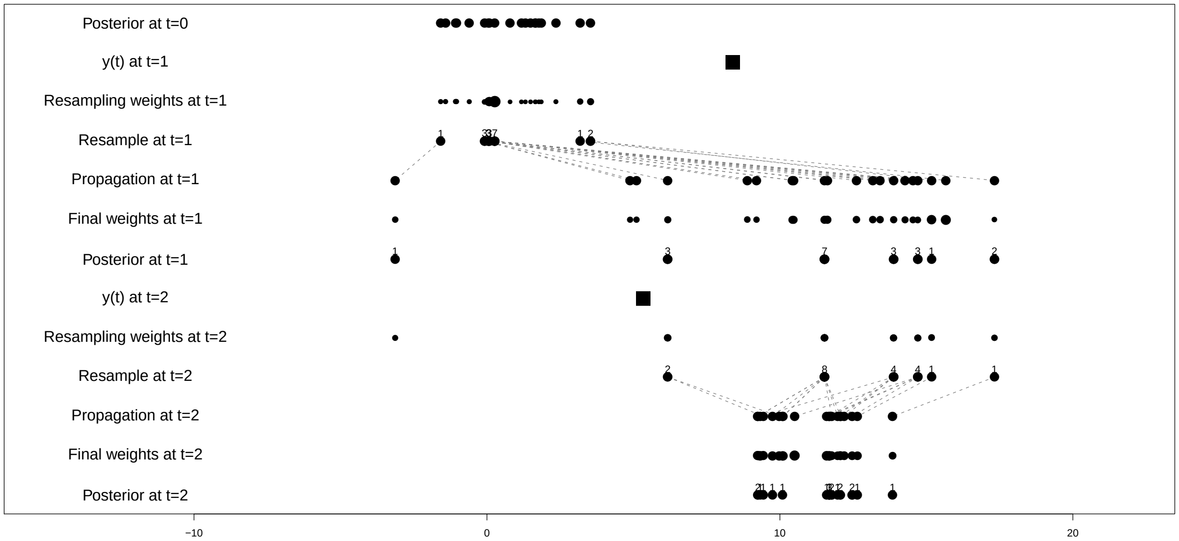

Figure 2: Schematization of the auxiliary particle filter. See description in Figure 1.

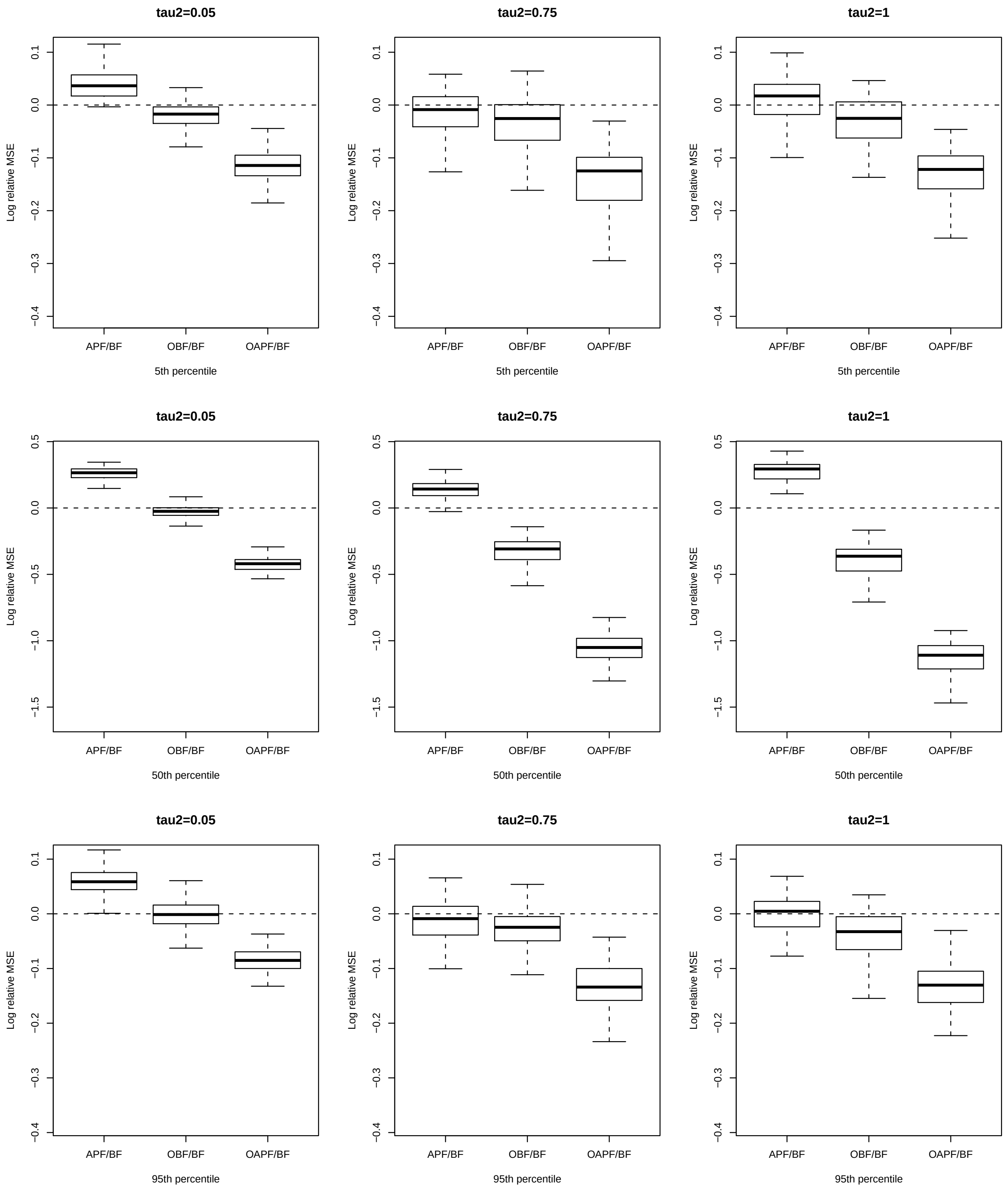

Figure 3: *AR(1) plus noise model (pure filter).* Relative mean square error performance (on the log-scale) of the four filters across $S = 50$ data sets of size $n = 100$ and $R = 50$ runs of each filter. Particle size for all filters is $N = 500$. Numbers below zero indicate a superior performance of the filter relative to the bootstrap filter (BF).

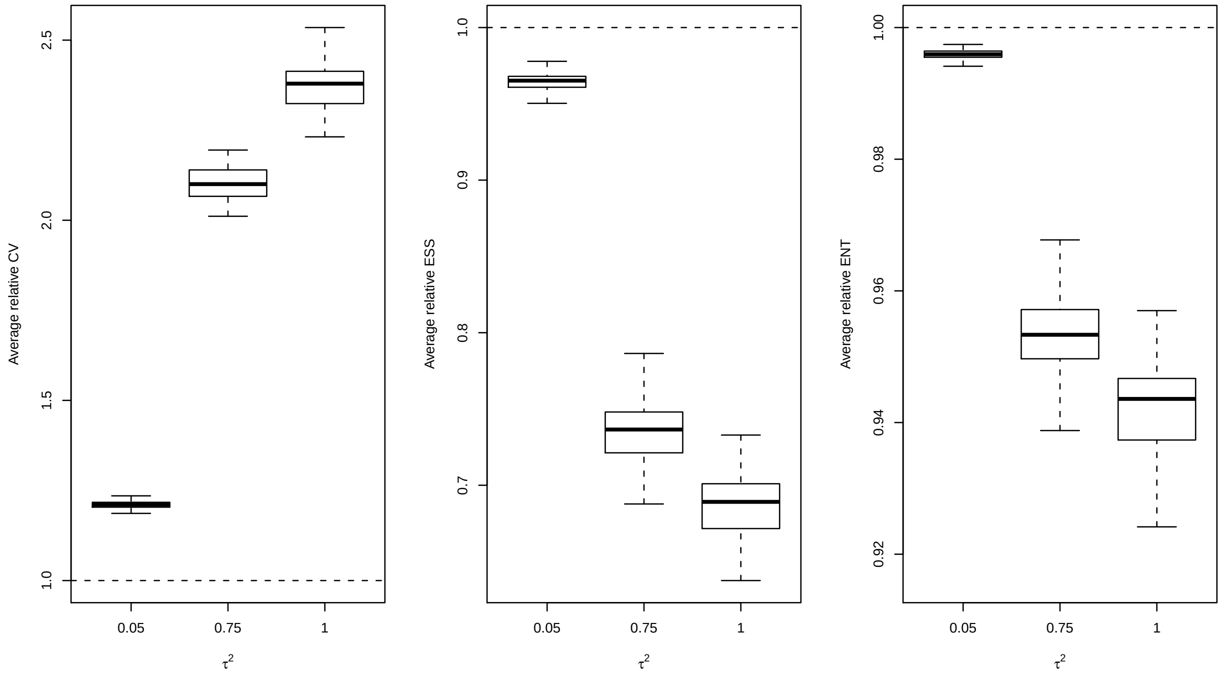

Figure 4: *AR(1) plus noise model (pure filter).* Detecting weight degeneracy via coefficient of covariation (CV), effective sample size (N_{eff}) and entropy (ENT) from Section 3.4. The measures are of the bootstrap filter (BF) relative to its optimal version (OBF) and based on $S = 50$ data sets of size $n = 100$ and $R = 50$ runs of each filter. Particle size for all filters and simulations is $N = 500$. Small CV, large N_{eff} and large ENT implies more balanced weights.

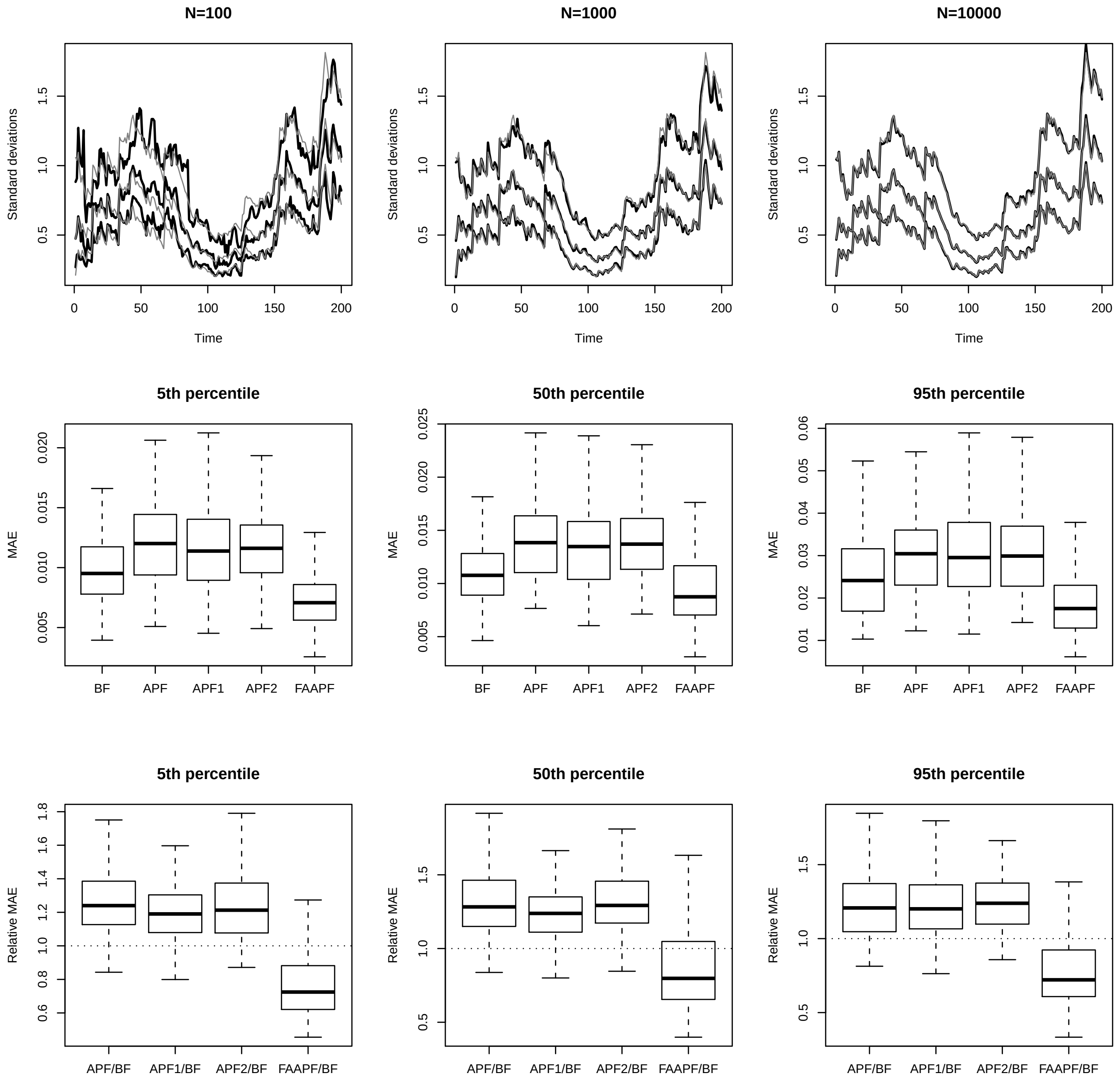

Figure 5: *SV-AR(1) model (pure filter).* Relative mean absolute error performances. The top panels show the trajectories of true (dark lines) and BF-based approximations (grey lines) for the αth percentiles $p(x_t|y_t)$, with α in $\{5, 50, 95\}$ and particle sizes N in $\{100, 1000, 10000\}$. True trajectories are basically BF with $N = 1000000$ (using APF or APF1 produced the same results). The middle panels show MAE based on $R = 100$ runs of each filter based on $N = 1000$ particles. APF1, APF2 and FAAPF are APF with first-stage weights $q(x_{t-1}|y_t)$ and resampling proposal $q(x_t|x_{t-1}, y_t)$ described in Section 3.5. Bottom panels are relative MAE of APF, APF1, APF2 and FAAPF relative to BF.

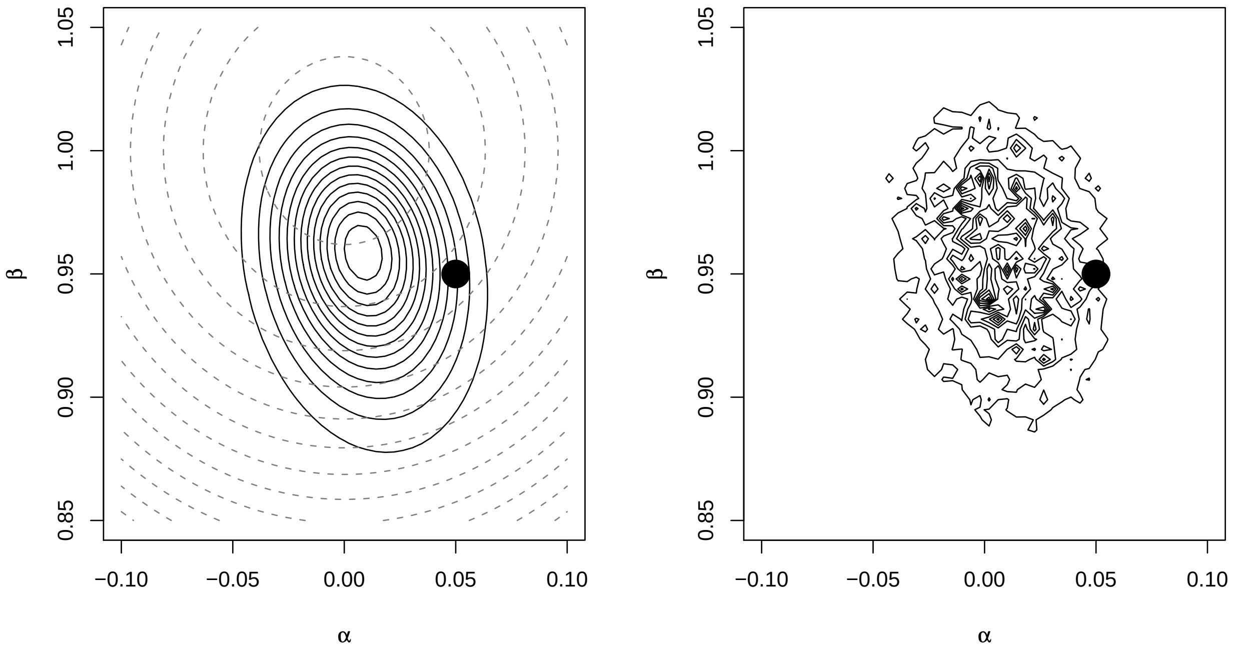

Figure 6: *AR(1) plus noise model (parameter learning).* Left panel: Contours of the prior distribution $p(\alpha, \beta)$ (dashed lines) and exact contours of the posterior distribution $p(\alpha, \beta|y_{1:n})$ (solid lines). Right panel: Contours of $\hat{p}(\alpha, \beta|y_{1:n}) \propto p(\alpha, \beta)\hat{p}(y_{1:n}|\alpha, \beta)$, where approximated integrated likelihood $\hat{p}(y_{1:n}|\alpha, \beta)$ is based on the OAPF of Section 3.2 and Equation (20).

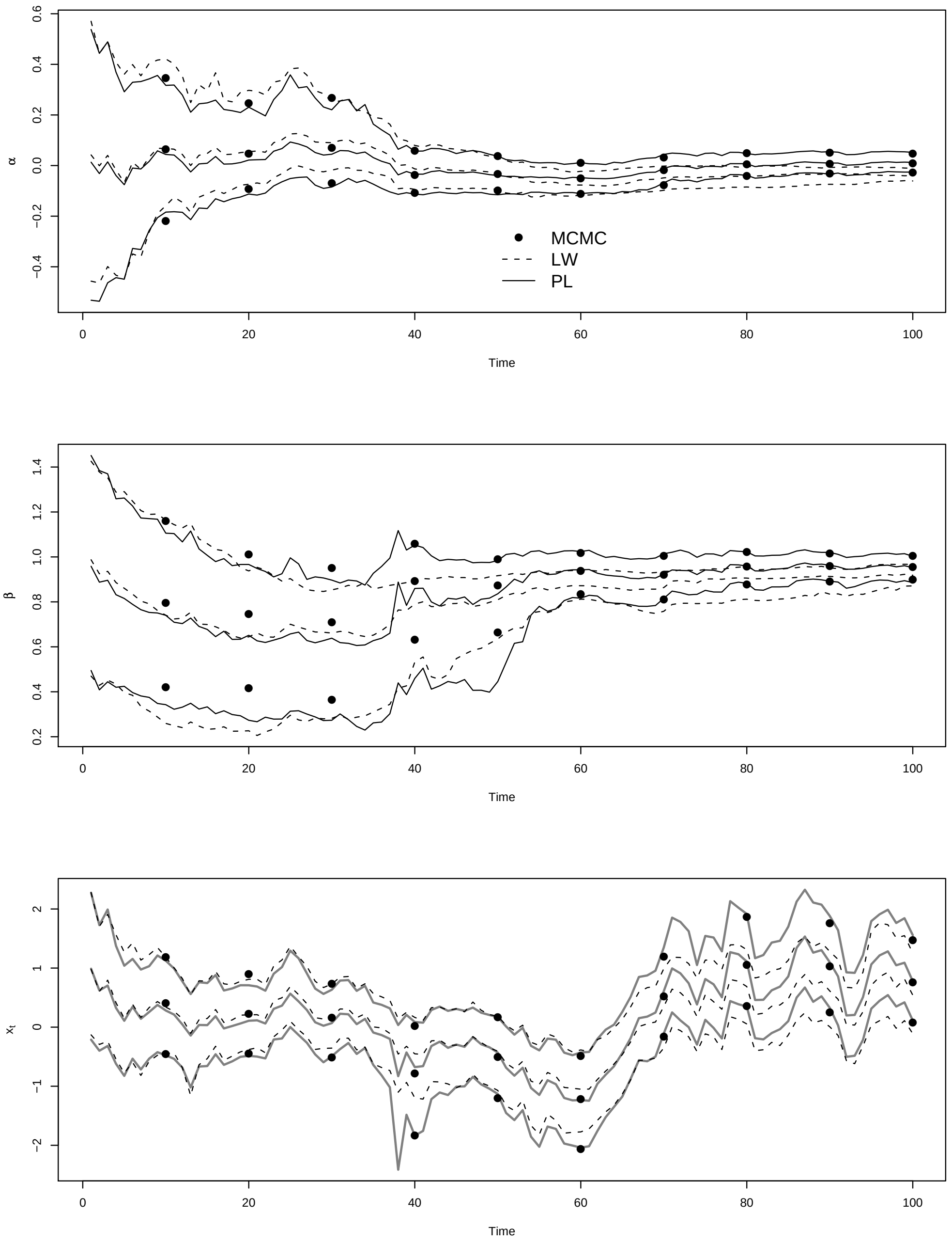

Figure 7: *AR(1) plus noise model (parameter learning).* 5th, 50th and 95th percentiles of $p(\alpha|y_{1:t})$ (top) and $p(\beta|y_{1:t})$ (middle) and $p(x_t|y_{1:t})$ (bottom) based on MCMC, LW filter and PL. MCMC is based on 1000 draws (after discarding the first 1000 draws). LW and PL are based on 1000 particles.

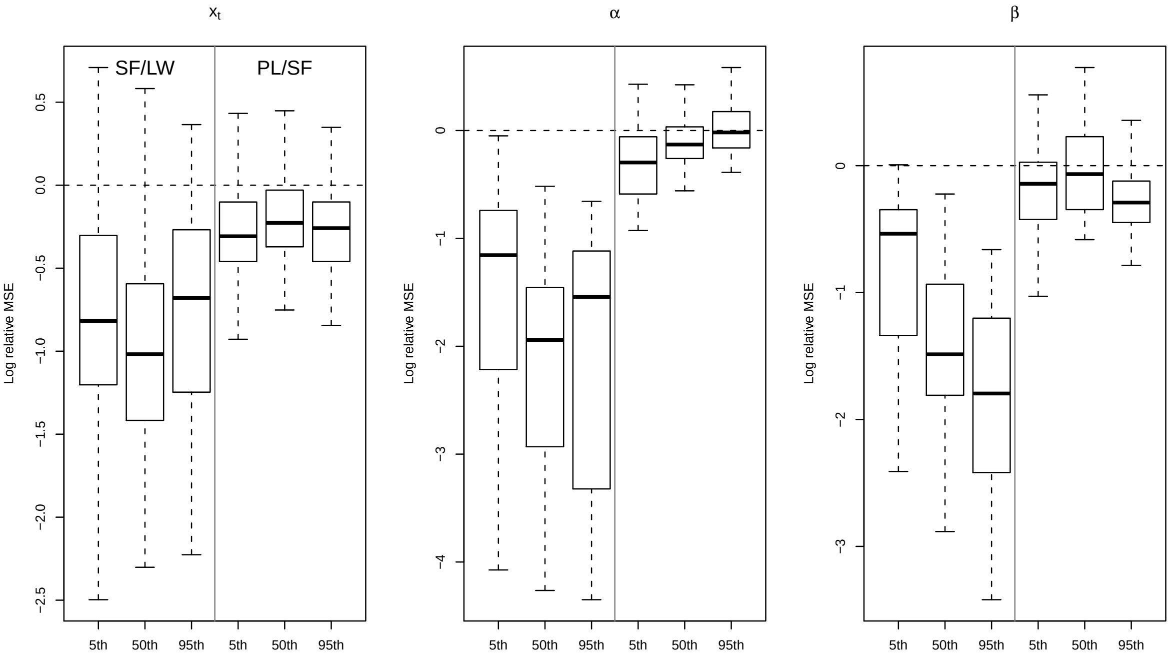

Figure 8: *AR(1) plus noise model (parameter learning).* Relative mean square error performance (on the log-scale) of LW filter, Storvik's filter and PL for one data set of size $n = 100$ and $R = 50$ runs of each filter. The number of particles is $N = 1000$. MSEs are based on comparisons to one PL run with 100000 particles.

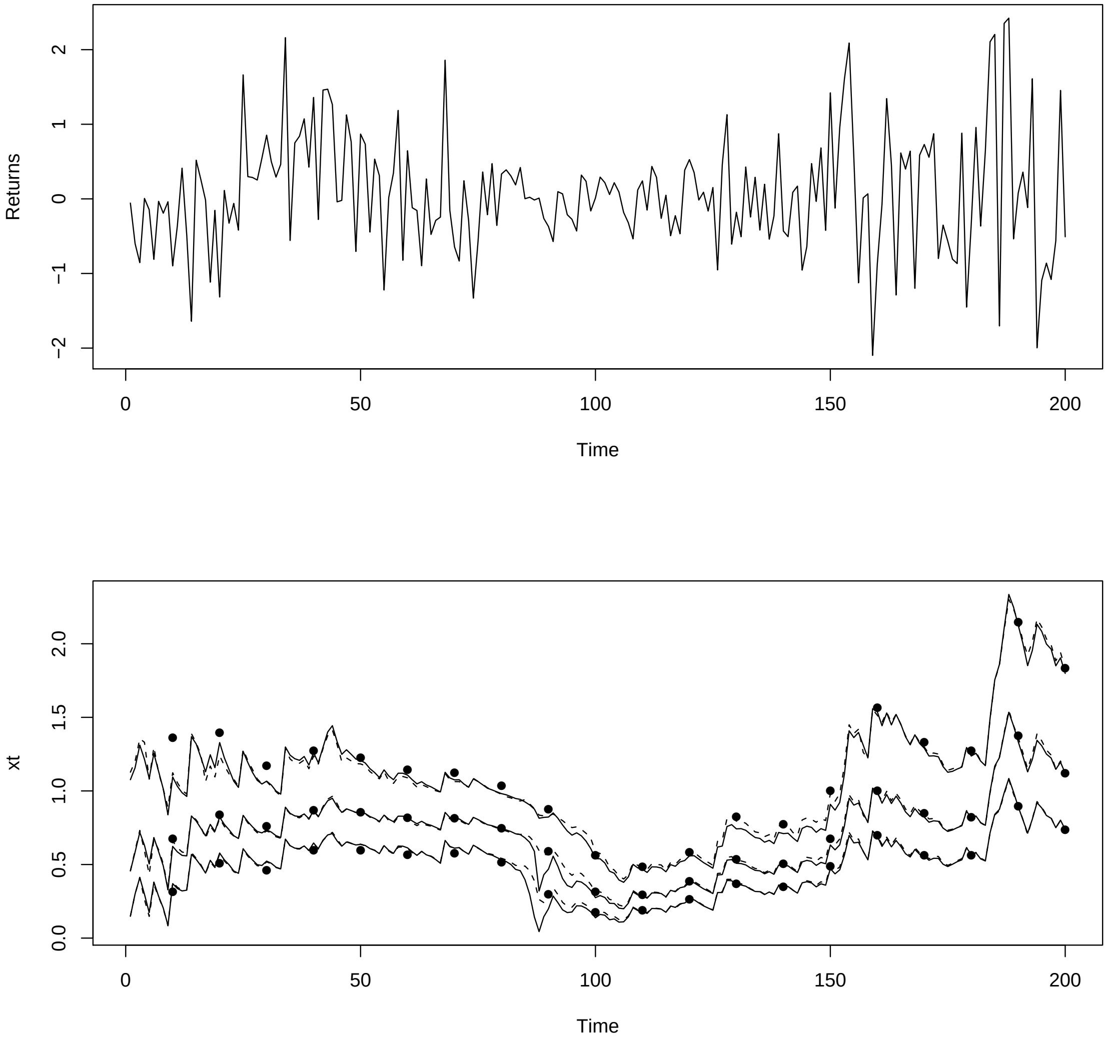

Figure 9: *SV-AR(1) model (parameter learning).* Top: simulated time series. Bottom: Approximate 5th, 50th and 95th percentiles of $p(x_t|y_{1:t})$ based on LW filter (solid lines), PL (dashed lines) and MCMC (dots).

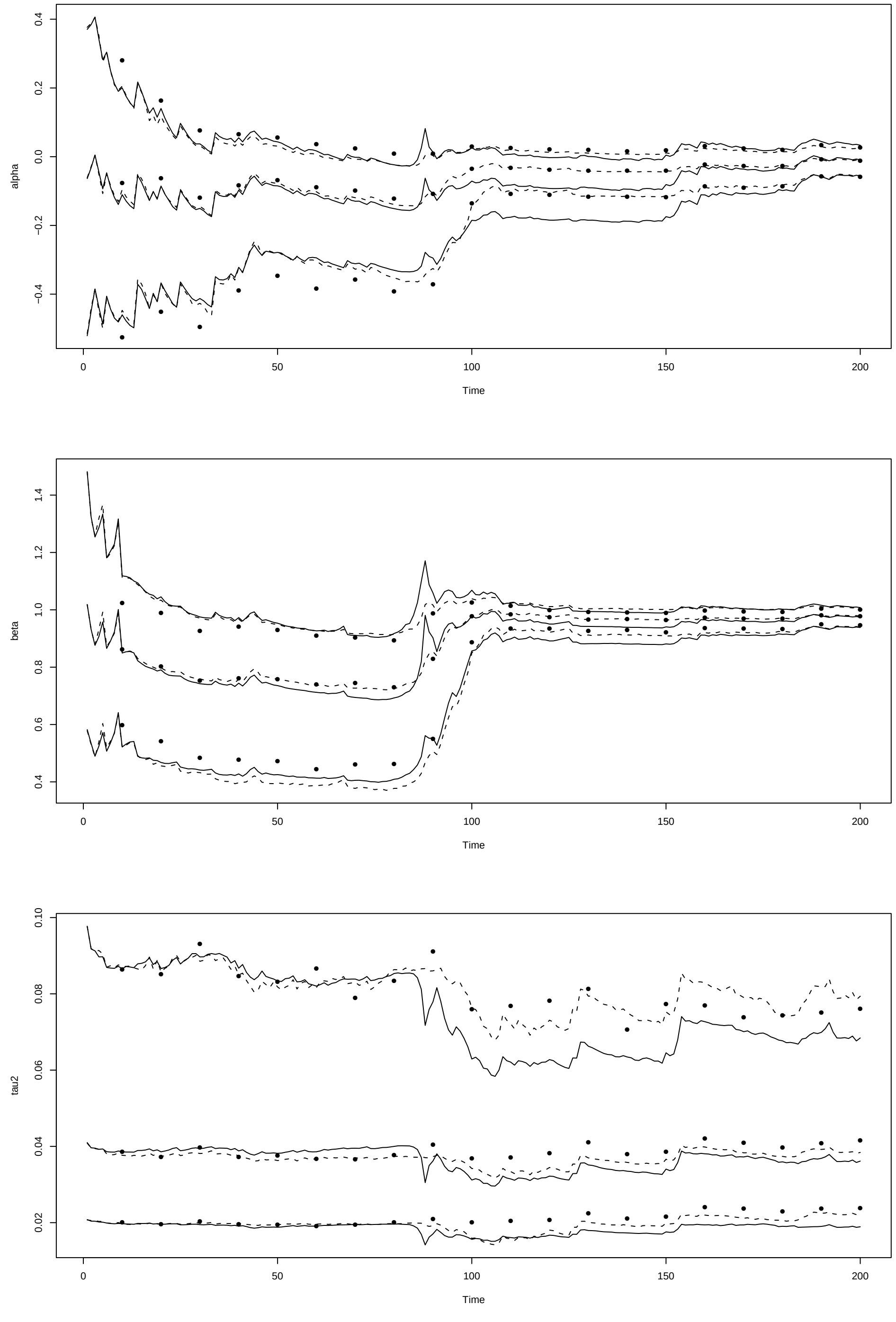

Figure 10: *SV-AR(1) model (parameter learning).* Approximate 5th, 50th and 95th percentiles of $p(\alpha|y_{1:t})$ (top), $p(\beta|y_{1:t})$ (middle) and $p(\tau^2|y_{1:t})$ (bottom) based on LW filter (solid lines), PL (dashed lines) and MCMC (dots).

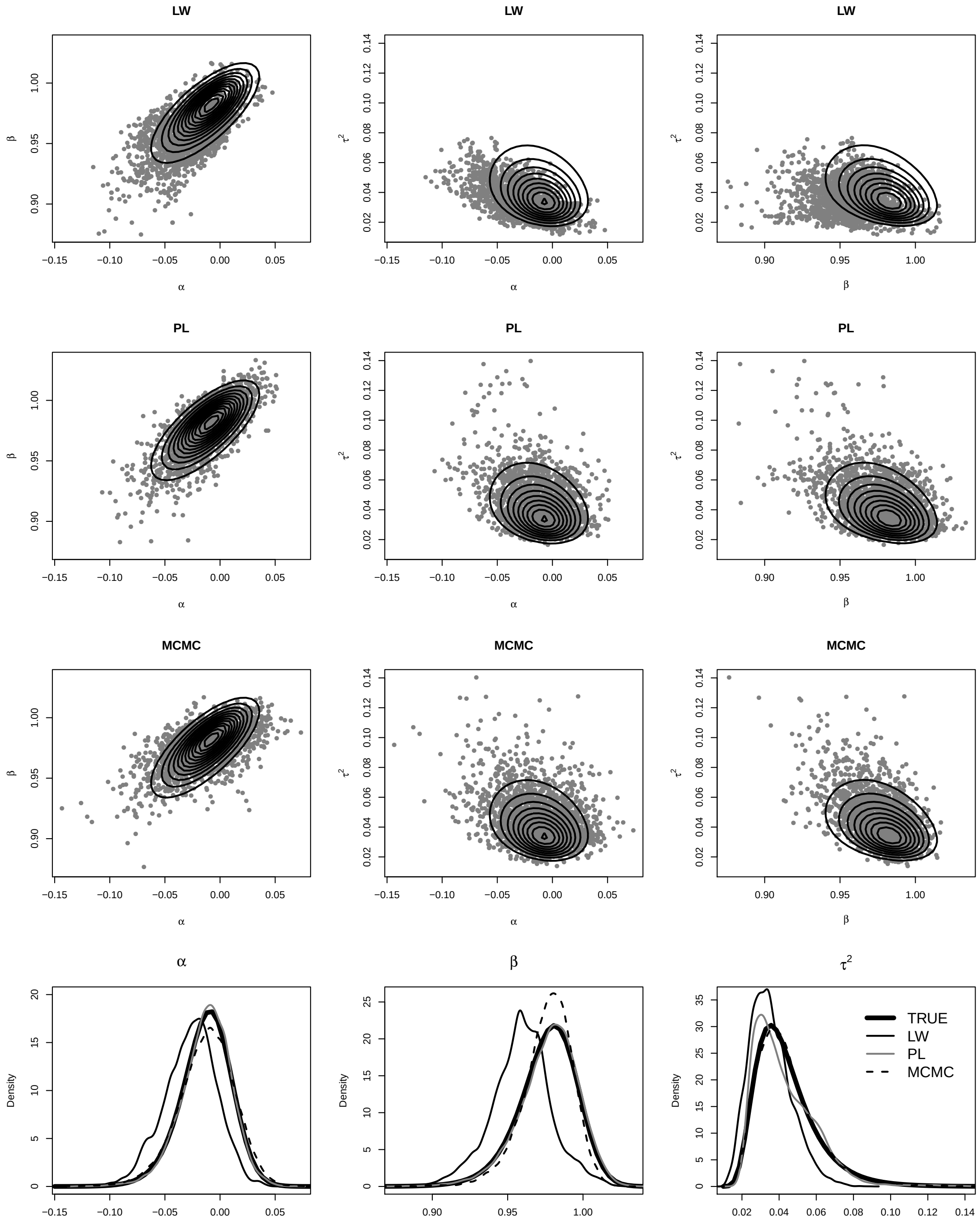

Figure 11: *SV-AR(1) model (parameter learning).* First three rows: True (contours) and approximated (dots) joint posterior distributions: $p(\alpha,\beta|y_{1:n})$ (1st row), $p(\alpha,\tau^2|y_{1:n})$ (2nd row) and $p(\beta,\tau^2|y_{1:n})$ (3rd row). Columns are based on LW filter, PL and MCMC. Forth row: True and approximated marginal posterior distributions $p(\alpha|y_{1:n})$, $p(\beta|y_{1:n})$ and $p(\tau^2|y_{1:n})$.

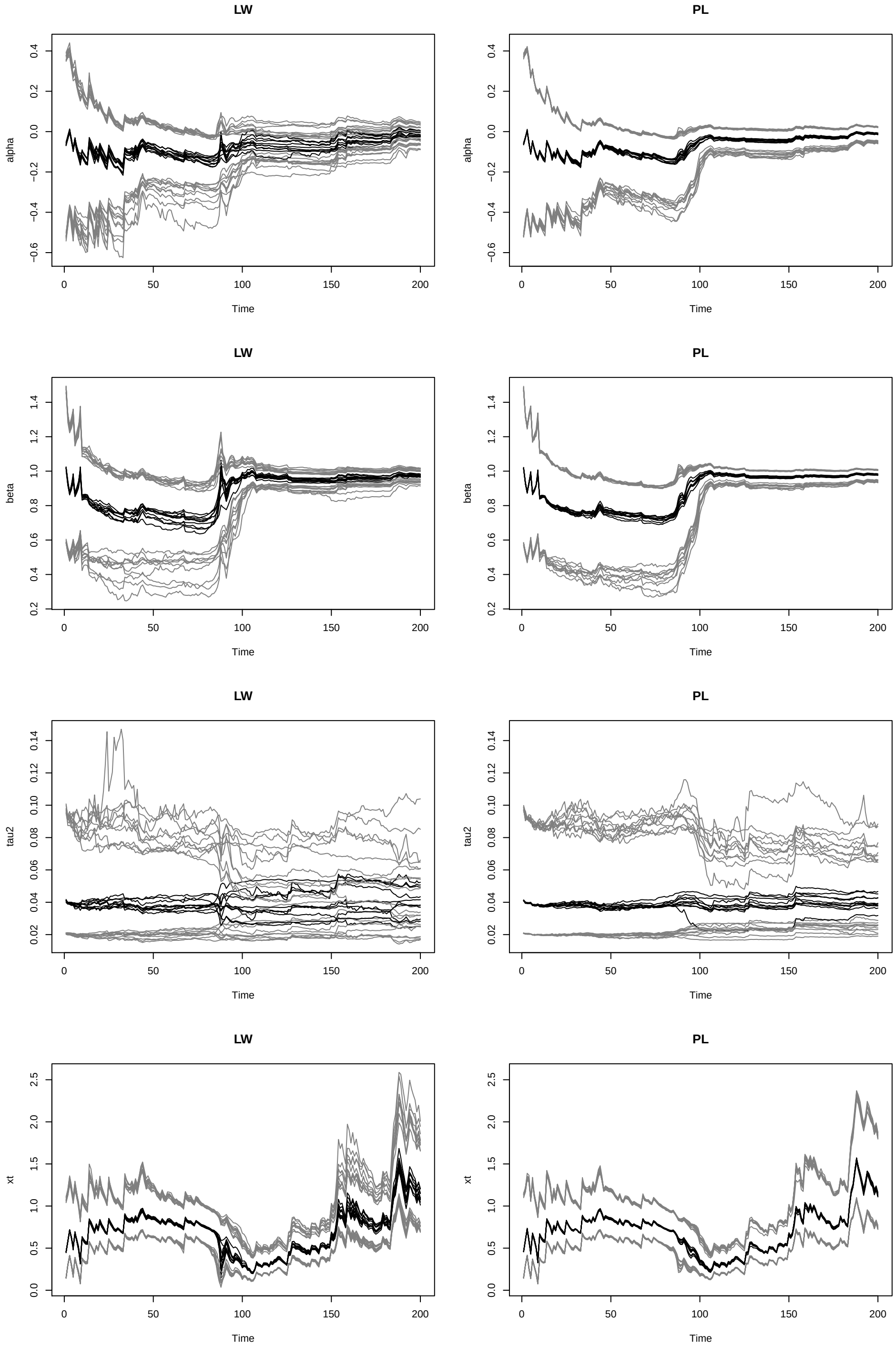

Figure 12: *SV-AR(1) model (parameter learning).* Approximate 5th, 50th and 95th percentiles of $p(\alpha|y_{1:t})$ (1st row), $p(\beta|y_{1:t})$ (2nd row), $p(\tau^2|y_{1:t})$ (3rd row) and $p(x_t|y_{1:t})$ (4th row) based on LW filter (left column) and PL (right column) for $R = 10$ replications of both filters and 10000 particles.

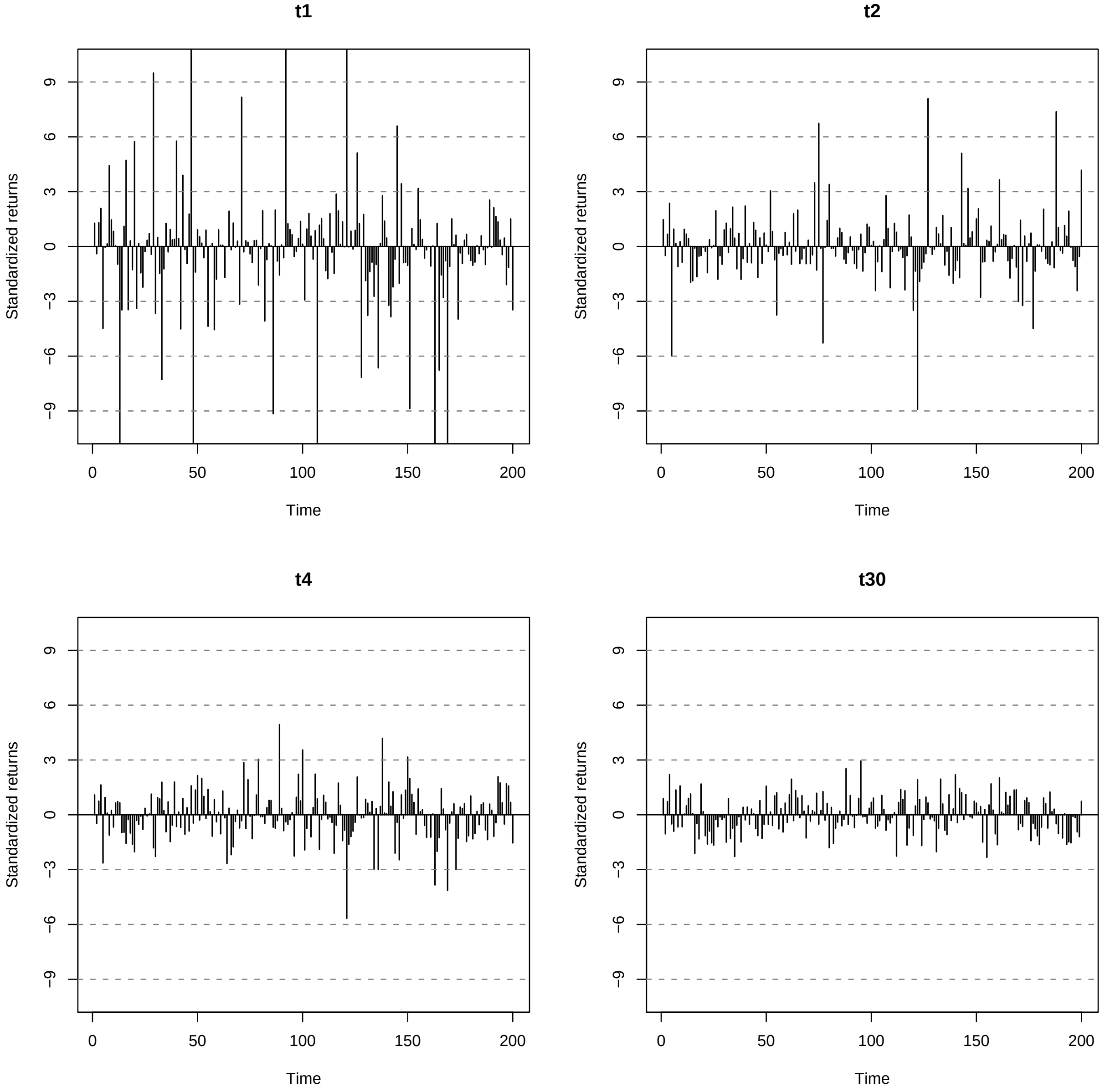

Figure 13: *SV-AR(1) model with t_ν error.* Time series of $n = 200$ data points simulated from the SV-AR(1) model with t_ν errors, $\alpha = -0.03$, $\beta = 0.97$, $\tau^2 = 0.03$, $x_0 = -0.1$ and $\nu \in \{1, 2, 4, 30\}$.

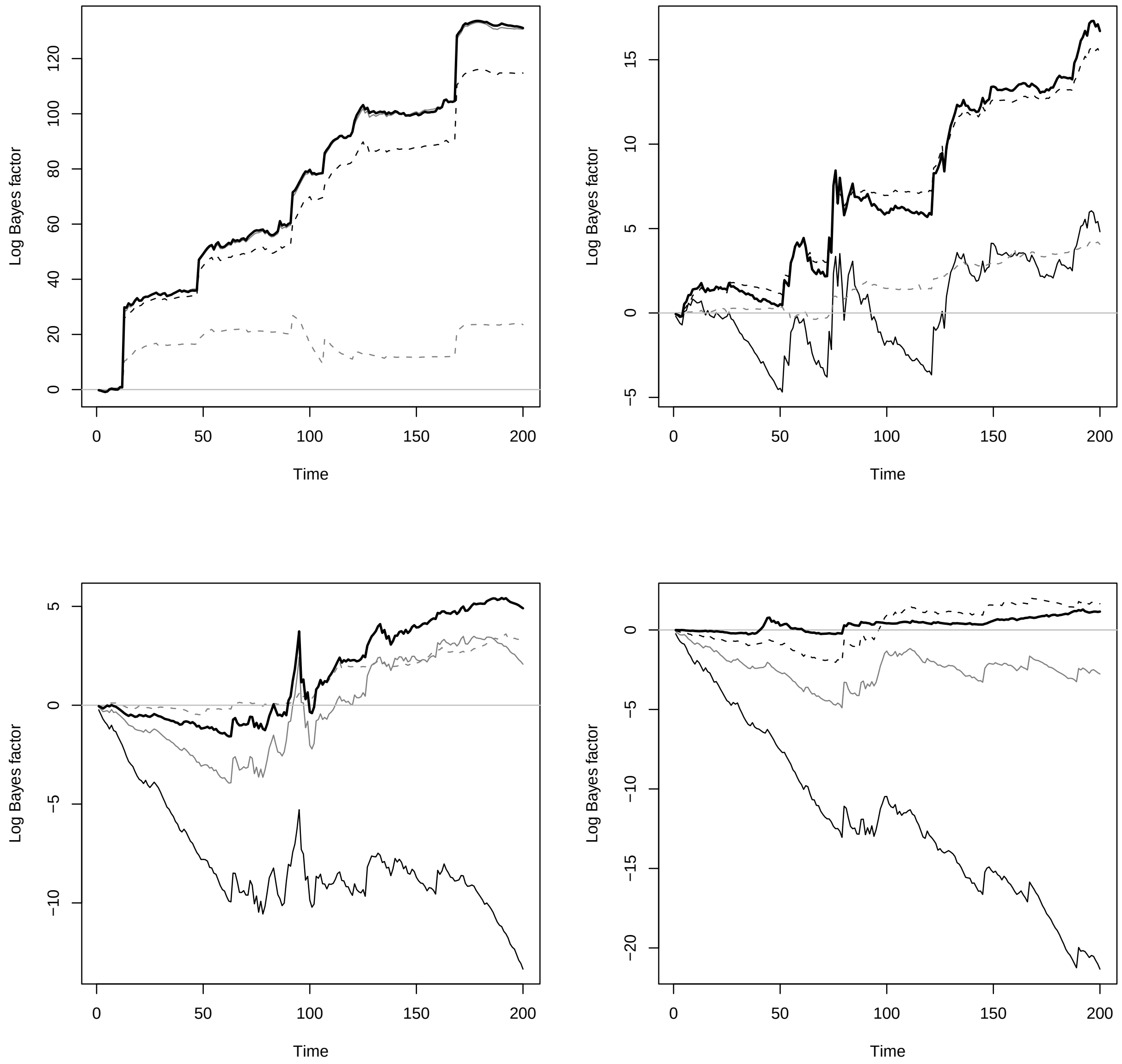

Figure 14: *SV-AR(1) model with t_ν error.* Bayes factors (in the log scale) of fitting t_ν models against normal models. Corresponding simulated data are in Figure 13. The lines are t_1 (solid dark line), t_2 (solid grey line), t_4 (dashed dark line) and t_{30} (dashed grey line). Thicker solid lines correspond to the true data generating models.

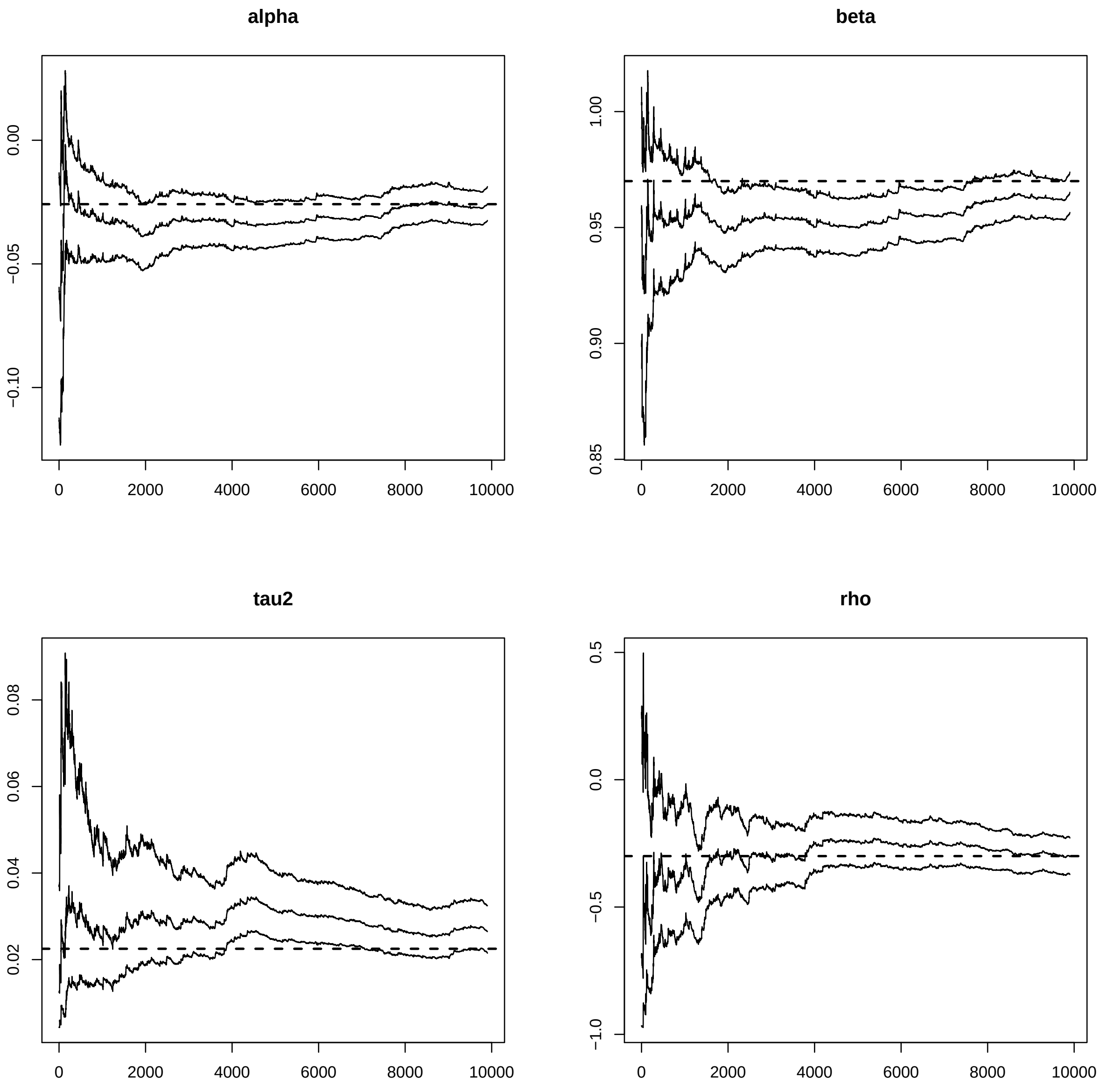

Figure 15: *SV-AR(1) model with leverage.* Sequential parameter learning based on LW filter and $N = 500000$ particles.

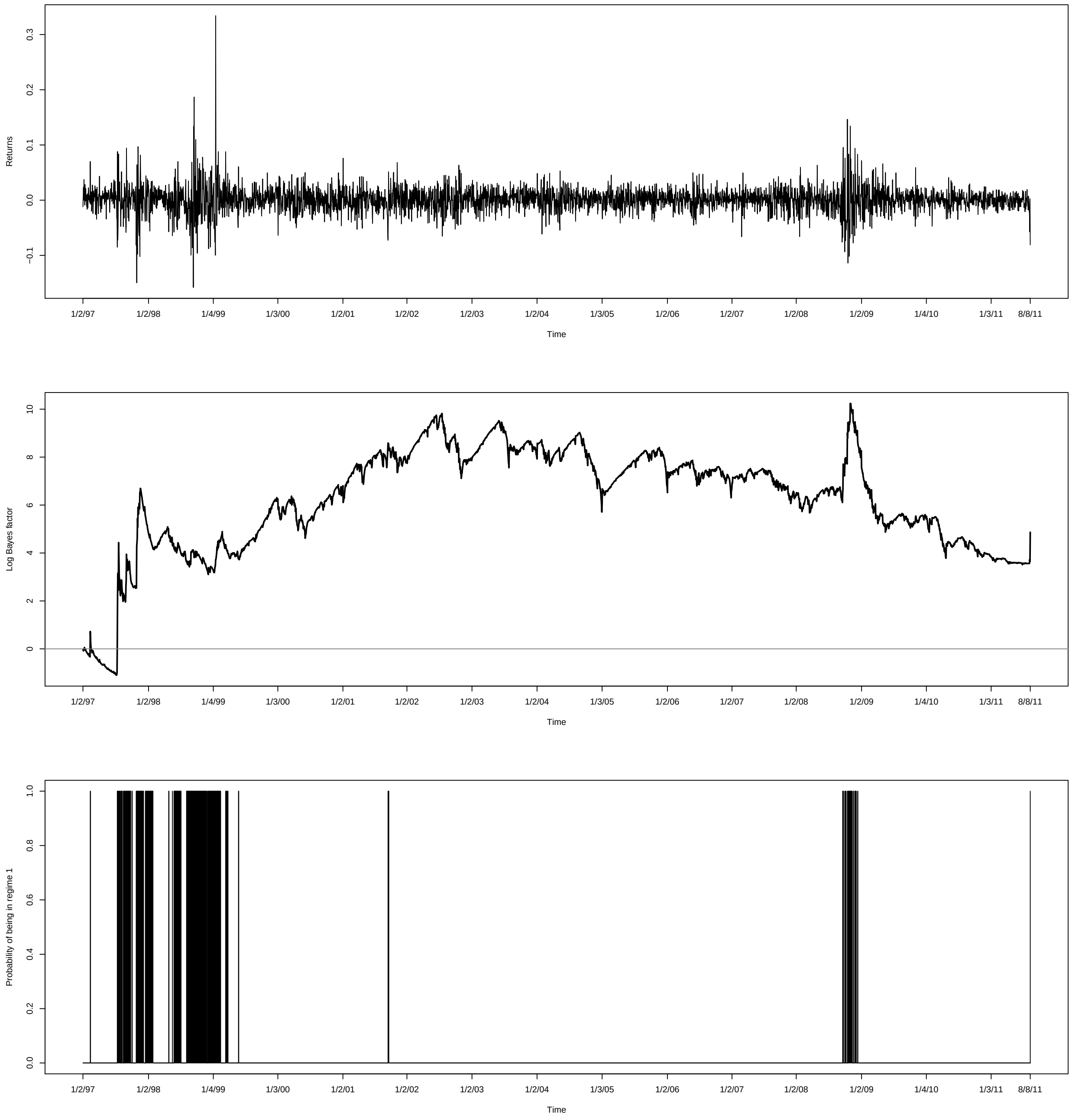

Figure 16: *SV-AR(1) model with regime switching.* IBOVESPA returns (top frame) from 01/02/1997 to 08/08/2011 ($n = 3612$ observations), Log Bayes factor (middle frame) and $Pr(s_t = 1|y_{1:t})$ (bottom frame). The LW filter is based on $N = 200000$ particles.

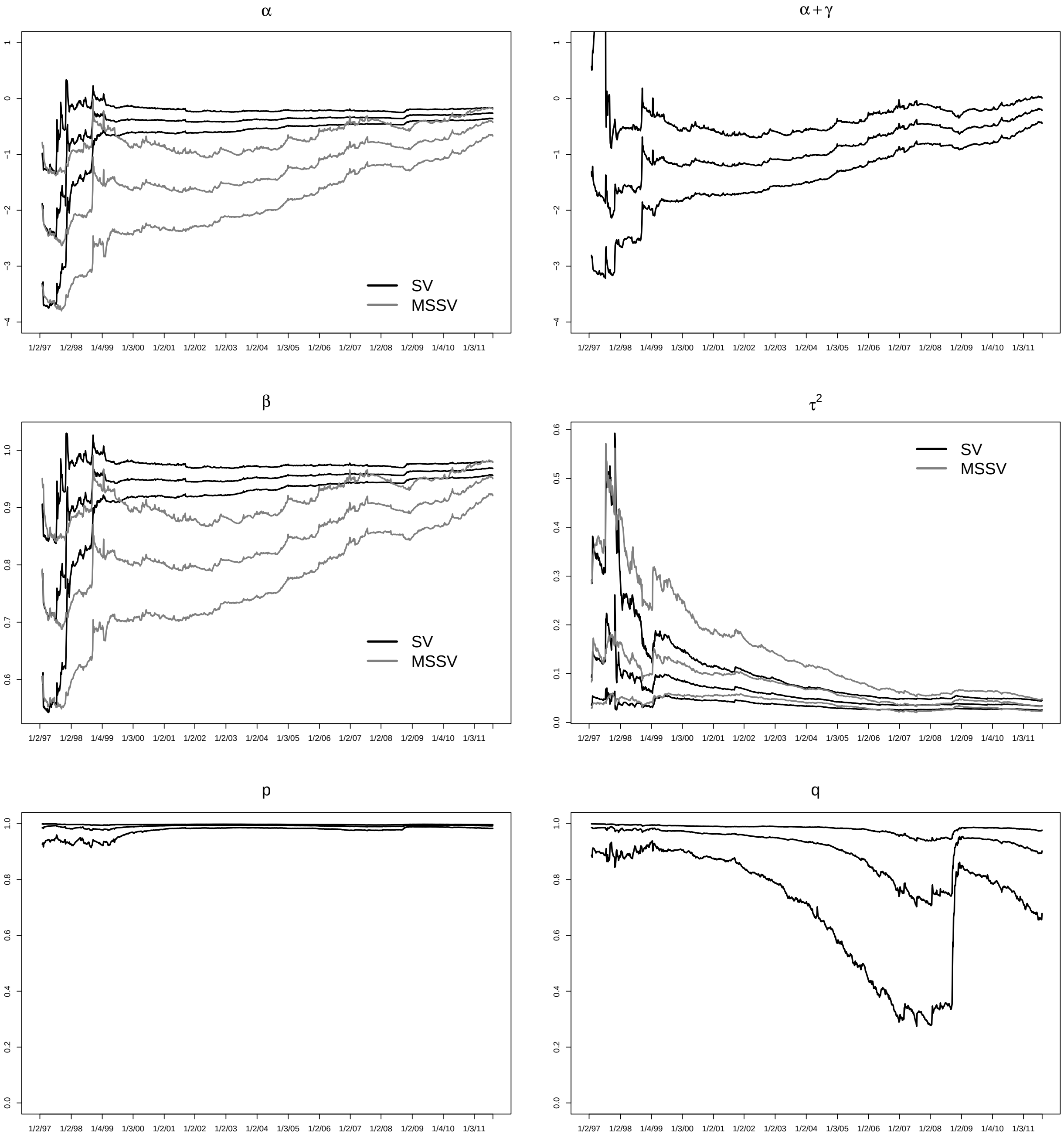

Figure 17: *SV-AR(1) model with regime switching.* Sequential posterior quantiles of α and $\alpha + \gamma$ (top row), β and τ^2 (middle row) and p and q (bottom row).

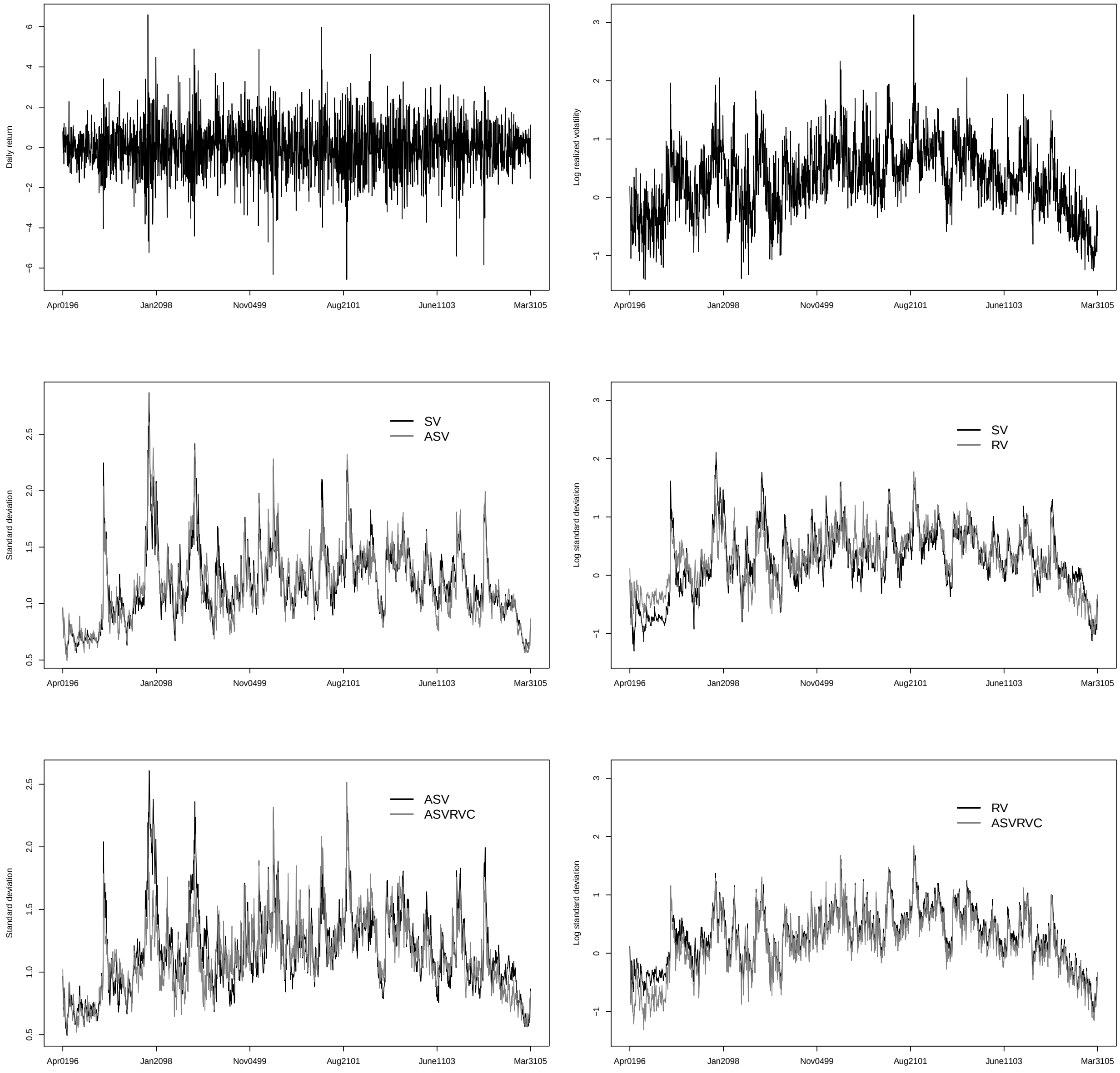

Figure 18: *SV-AR(1) plus realized volatility model.* Top row: Daily returns and logarithm of daily realized volatilities. Middle and bottom rows: Posterior medians of standard deviations and their logarithms based on four models: SV-AR(1) model (SV), SV with leverage (ASV), realized volatility (RV) and ASV-RV combined (ASV-SRVC).

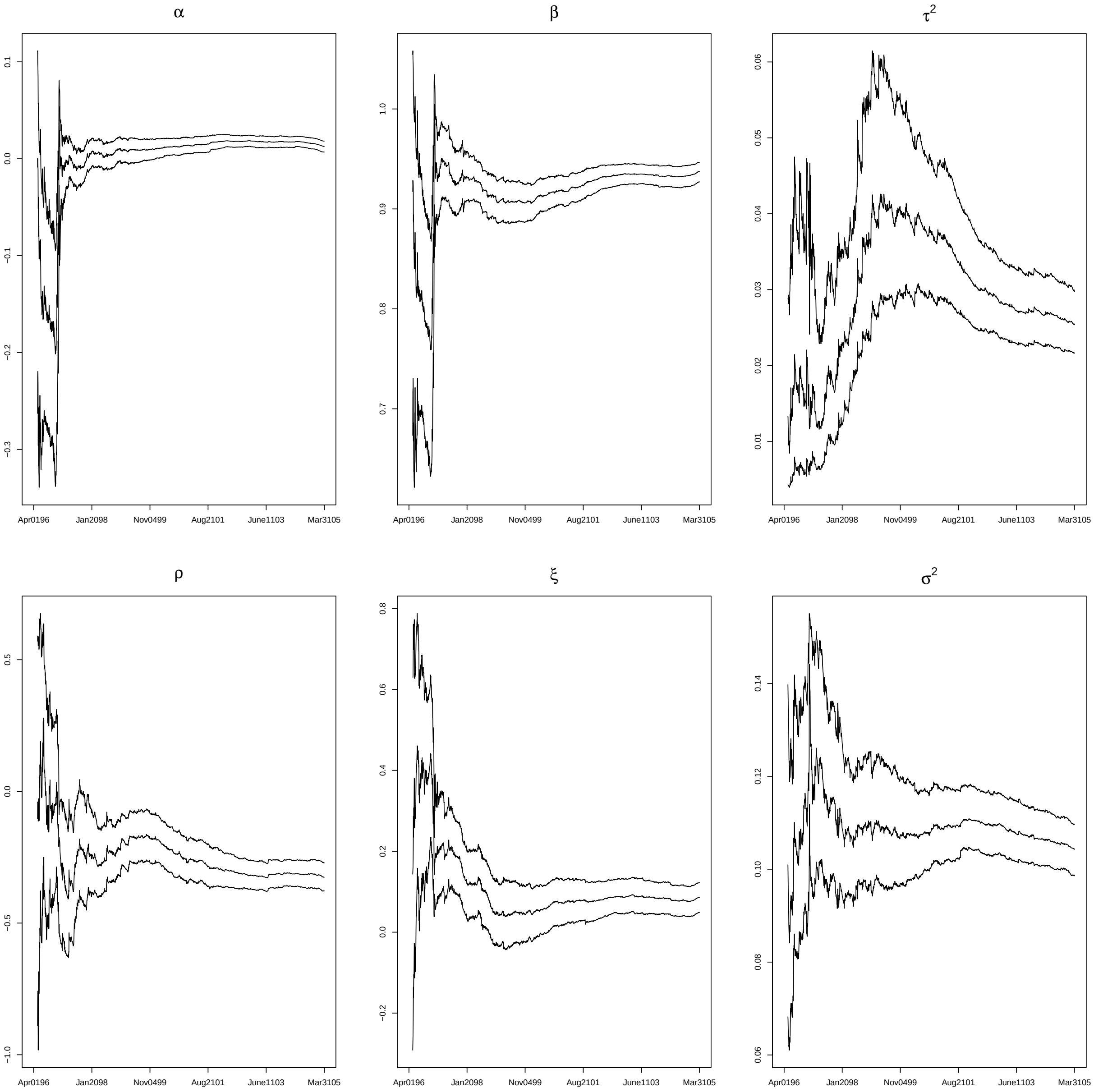

Figure 19: *SV-AR(1) plus realized volatility model.* Percentiles of the sequential posterior distributions of the static parameters. Top row: (α, β, τ^2). Bottom row: (ρ, ξ, σ^2).

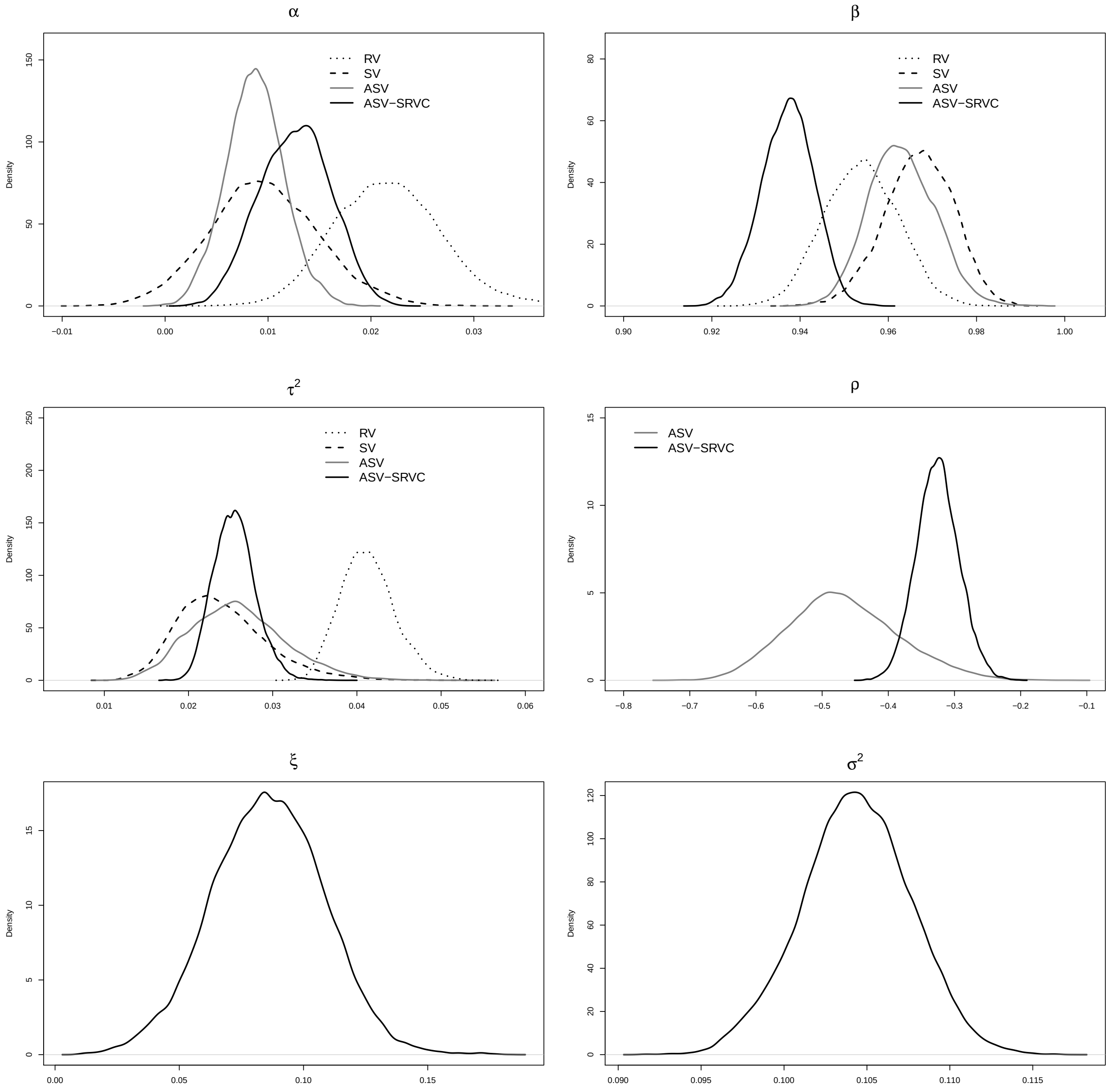

Figure 20: *SV-AR(1) plus realized volatility model.* Posterior distributions for the static parameters of the four competing volatility models (see Figure 18). SV and RV: (α, β, τ^2), ASV: $(\alpha, \beta, \tau^2, \rho)$, ASV-SRVC: $(\alpha, \beta, \tau^2, \rho, \xi, \sigma^2)$.