Particle Filters and Bayesian Inference In Financial Econometrics

Hedibert F. Lopes and **Ruey S. Tsay**
The University of Chicago Booth School of Business
5807 South Woodlawn Avenue, Chicago, Illinois, 60637.
[hlopes,ruey.tsay]@ChicagoBooth.edu

Abstract

In this paper we review *sequential Monte Carlo* (SMC) methods, or particle filters (PF), with special emphasis on its potential applications in financial time-series analysis and econometrics. We start with the well-known normal dynamic linear model, also known as the normal linear state space model, for which sequential state learning is available in closed form via standard Kalman filter and Kalman smoother recursions. Particle filters are then introduced as a set of Monte Carlo schemes that enable Kalman-type recursions when normality or linearity or both are abandoned. The seminal *bootstrap filter* (BF) of Gordon, Salmond and Smith (1993) is used to introduce the SMC jargon, potentials and limitations.

We also review the literature on parameter learning, an area that started to attract much attention from the particle filter community in recent years. We give particular attention to the Liu-West filter (Liu and West, 2001), Storvik filter (Storvik, 2002) and *particle learning* (PL) of Carvalho, Johannes, Lopes and Polson (2010). We argue that the BF and the *auxiliary particle filter* (APF) of Pitt and Shephard (1999) define two fundamentally distinct directions within the particle filter literature. We also show that the distinction is more pronounced with parameter learning and argue that PL, which follows the APF direction, is an attractive extension.

One of our contributions is to sort out the research from BF to APF (during the 90s), from APF to now (the 00s) and from Liu-West filter to Storvik filter to PL. To this end, we provide code in `R` for all the examples of the paper[1]. Readers are invited to find their own way into this dynamic and active research arena.

Key words: Particle learning; Sequential Monte Carlo; Markov chain Monte Carlo; Stochastic Volatility; Realized Volatility; Nelson-Siegel model.

1 Introduction

The Kalman filter (KF) and its many variants and generalizations have played a fundamental role in modern time series analysis by allowing the study and estimation of complex dynamics and by drawing the attention of researchers and practitioners to the rich class of *state-space models*, also known as *dynamic models* (Harvey, 1989, West and Harrison, 1989, 1997). Well-known and widely used variants of KF include (i) the extended KF (Jazwinski, 1970; West,

[1] http://faculty.chicagobooth.edu/hedibert.lopes/research/Jforecasting-PF.

Harrison and Migon, 1985); (ii) the Gaussian sum filter (Alspach and Sorenson, 1972); (iii) the unscented KF (Julier and Uhlmann, 1997; Van der Merwe *et al.*, 2000); and (iv) the Gaussian quadrature KF (Ito and Xiong, 2000).

Despite their wide applicability, approximations provided by these variants become less effective when substantial nonlinearities and/or extreme non-Gaussianity are present in the data. To overcome the difficulty, the last two decades have been exposed to an increasing number of Monte Carlo (MC) based approximations for state-space models. These MC methods are basically divided into two major categories: Markov chain Monte Carlo (MCMC) schemes for offline/batch sampling and sequential Monte Carlo (SMC) schemes for online/sequential sampling. For example, Carlin *et al.* (1992), Carter and Kohn (1994), Frühwirth-Schnatter (1994) and Shephard (1994) propose MCMC methods to estimate general state-space models. However, MCMC-based algorithms are prohibitively costly when performing online estimation of states and parameters; see Gamerman and Lopes (2006).

SMC methods, also known as *particle filters*, are MC schemes that, when used in the state-space context, rebalance draws from the posterior distribution of the states and parameters at a given time (the particles) based on the next observation via its likelihood. In their seminal paper, Gordon, Salmond and Smith (1993) propose one of the most popular filters, the *bootstrap filter* (BF), which is based on a sampling importance resampling (SIR) argument (Smith and Gelfand, 1991). Also influential from the start are the works on sequential Bayesian imputation by Kong, Liu and Wong (1994) and Liu and Chen (1995).

In this paper we review the bootstrap filter and its variants. We also introduce the auxiliary particle filter (APF) of Pitt and Shephard (1999) (see also the discussion in Liu and Chen, 1998) and argue that both filters define two directions within the SMC literature, namely *sample-resample* and *resample-sample* methods. This is done in Section 3, which ends with a list of additional review papers and books on SMC. Section 4 starts by showing how both BF and APF can be used to approximate the likelihood function of fixed parameters. We then introduce the Liu and West filter (Liu and West, 2001) that generalizes APF to sequentially update the posterior distributions of parameters. The section also introduces the *particle learning* (PL) of Carvalho *et al.* (2010). Some illustrative examples appear in Section 5. Final remarks and current research directions are presented in Section 6.

2 Normal dynamic linear model

To introduce the ideas of the particle filter, let us start with the well-known *normal dynamic linear model* (NDLM),

$$y_t = F_t' x_t + v_t, \tag{1}$$
$$x_t = G_t x_{t-1} + w_t, \tag{2}$$

where v_t and w_t are temporally and mutually independent Gaussian sequences with zero mean and variances σ_t^2 and τ_t^2, respectively. Eq. (1) is referred to as the observation equation that relates the observed series y_t to the state vector x_t. Eq. (2) is the state transition equation that governs the time evolution of the state, which might be latent. The local level and the local

linear trend models are special cases of the NDLM. In the local level model, $y_t = x_t + v_t$ and $x_t = x_{t-1} + w_t$ so that $F_t = G_t = 1$, $\sigma_t^2 = \sigma^2$ and $\tau_t^2 = \tau^2$ for all t. In the local linear trend model, $y_t = x_{1t} + v_t$, $x_{1t} = x_{1,t-1} + x_{2,t-1} + w_{1t}$ and $x_{2t} = x_{2,t-1} + w_{2t}$, and we have $x_t = (x_{1t}, x_{2t})'$, $F_t = (1, 0)'$, $G = (g_1, g_2)$, $g_1 = (1, 0)'$, $g_2 = (1, 1)'$, $\sigma_t^2 = \sigma^2$ and $\tau_t^2 = \tau$ for all t, where τ is a 2×2 positive definite matrix.

Conditionally on the quadruple $\{F_t, G_t, \sigma_t^2, \tau_t^2\}$, for $t = 1, \ldots, T$, and on the initial distribution $(x_0|y^0) \sim N(m_0, C_0)$, it is straightforward to show that

$$\begin{aligned} x_t|y^{t-1} &\sim N(a_t, R_t), && (3)\\ y_t|y^{t-1} &\sim N(f_t, Q_t), && (4)\\ x_t|y^t &\sim N(m_t, C_t), && (5) \end{aligned}$$

for $t = 1, \ldots, T$, where $y^t = (y_1, \ldots, y_t)'$ and $N(a, b)$ denotes the normal distribution with mean a and variance b. The three densities in Eqs. (3)-(5) 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^{t-1} and the posterior density of x_t given y^t. The means and variances of the three densities are provided by the *Kalman recursions*:

$$\begin{aligned} a_t &= G_t m_{t-1} \quad \text{and} \quad R_t = G_t C_{t-1} G_t' + \tau_t^2, && (6)\\ f_t &= F_t' a_t \quad \text{and} \quad Q_t = F_t' R_t F_t + \sigma_t^2, && (7)\\ m_t &= a_t + A_t e_t \quad \text{and} \quad C_t = R_t - A_t Q_t A_t', && (8) \end{aligned}$$

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*

$$\begin{aligned} x_t|x_{t+1}, y_t &\sim N(h_t, H_t), && (9)\\ x_t|y^T &\sim N(m_t^T, C_t^T), && (10) \end{aligned}$$

where

$$\begin{aligned} h_t &= m_t + B_t(x_{t+1} - a_{t+1}) \quad \text{and} \quad H_t = C_t - B_t R_{t+1} B_t', && (11)\\ m_t^T &= m_t + B_t(m_{t+1}^T - a_{t+1}) \quad \text{and} \quad C_t^T = C_t - B_t^2(R_{t+1} - C_{t+1}^T), && (12) \end{aligned}$$

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

An important and rich subclass of the NDLM assumes that F_t and G_t are both known, while $\sigma_t^2 = \sigma^2$ and $\tau_t^2 = \tau^2$ are both unknown variances. In this case, the above Kalman recursions can be used to marginalize out the states based on Eq. (4), i.e.,

$$p(y^T|\sigma^2, \tau^2) = \prod_{t=1}^{T} f(y_t; f_t, Q_t), \qquad (13)$$

where $f(x; \mu, \sigma^2)$ 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 (σ^2, τ^2). In other words, should the main objective be sampling from $p(x^T, \sigma^2, \tau^2|y^T)$, then draws can be obtained in two steps:

1. Draw (σ^2, τ^2) from $p(\sigma^2, \tau^2|y^T)$, which is proportional to the prior $p(\sigma^2, \tau^2)$ times the likelihood from Eq. (13);

2. Draw x^T from $p(x^T|\sigma^2, \tau^2)$ by first computing forward moments via Eqs. (6)-(8) and (11), and then sampling backward x_t conditional on x_{t+1} and y^t via Eq. (9).

Sampling (σ^2, τ^2) from step 1 can be performed by SIR, acceptance-rejection or Metropolis-Hastings-type algorithms or replaced by a Gibbs step that draws (σ^2, τ^2) conditional on (y^T, x^T). Reis, Salazar and Gamerman (2006) compare the performance of these sampling and other MCMC schemes in the context of the local level model. Step 2 is known as the *forward filtering, backward sampling* (FFBS) algorithm (Carter and Kohn, 1994; Frühwirth-Schnatter, 1994).

For the NLDM, all densities needed for making inference are well-known and they can easily be carried out in applications. On the other hand, difficulties arise when the model is nonlinear or non-Gaussian, become no closed-form densities are available. As we discuss below, particle filters provide an effective approach to overcoming the difficulty.

3 Basic particle filters

Let us consider a more general dynamic model where the assumptions of normality and/or linearity are relaxed. The observation and state transition equations become

$$\begin{aligned} y_t|x_t &\sim p(y_t|x_t), \\ x_t|x_{t-1} &\sim p(x_t|x_{t-1}), \quad t = 1, 2, \ldots. \end{aligned}$$

Denote the initial probability density of the state by $p(x_0)$. All static parameters, such as σ^2 and τ^2 from the previous section, are assumed to be known throughout this section. Batch and sequential parameter learning are deferred to Section 4. The Kalman recursions from Eqs. (3) and (5) are now replaced, respectively, by

$$p(x_t|y^{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y^{t-1})dx_{t-1} \tag{14}$$

and

$$p\left(x_t|y^t\right) = \frac{p\left(y_t|x_t\right)p\left(x_t|y^{t-1}\right)}{p\left(y_t|y^{t-1}\right)}. \tag{15}$$

In most real-world applications, outside the realm of NDLM, the integration with respect to x_{t-1} in (14) and the implementation of Bayes' theorem in (15) are both analytically intractable and/or computationally costly. As mentioned in the Introduction, there exist approximations for sequential state estimation and filtering based on Kalman-like filters, such as the extended Kalman filter and the unscented Kalman filter. Also, as discussed in the Introduction, there exist several MCMC-type samplers for batch estimation of the whole state vector and parameters similar to the FFBS introduced in Section 2.

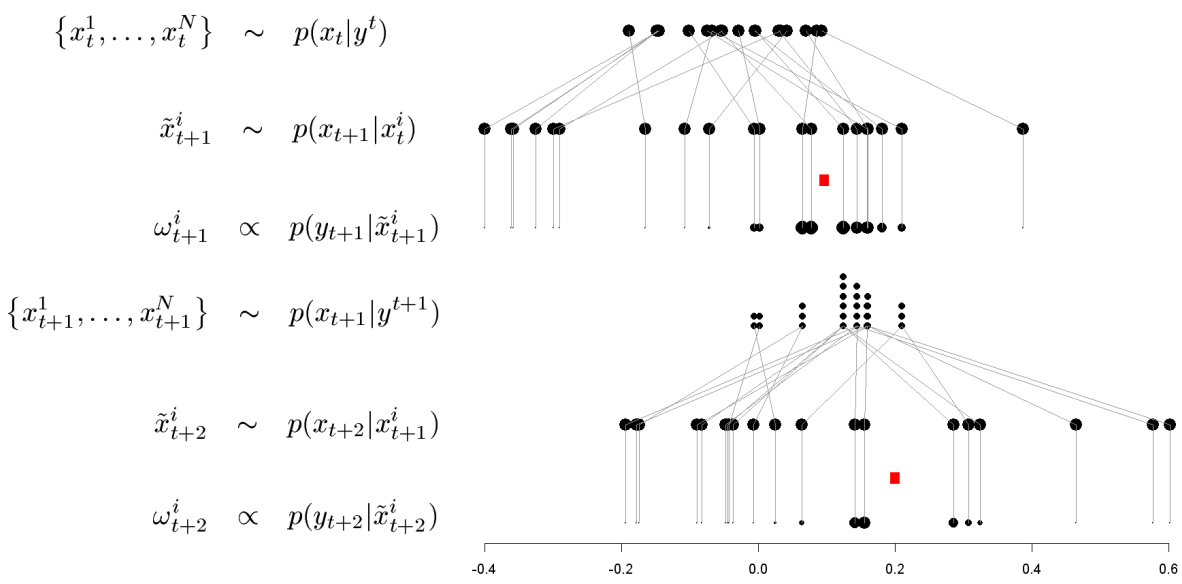

Figure 1: *Bootstrap filter.* A schematic representation of the bootstrap filter over two time periods. The squares are y_{t+1} and y_{t+2}. From top to bottom, the first, second, fourth and fifth set of dots represent particles, while the third and sixth set of dots represent particle weights.

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 the 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.

The goal of most particle filters is to draw a set of i.i.d. particles $\{x_t^{(i)}\}_{i=1}^N$ that approximates $p(x_t|y^t)$ by starting with a set of i.i.d. particles $\{x_{t-1}^{(i)}\}_{i=1}^N$ that approximates $p(x_{t-1}|y^{t-1})$.

The most popular filters are the bootstrap filter (BF), also known as the sequential importance sampling with resampling (SISR) filter, proposed by Gordon, Salmond and Smith (1993), and the auxiliary particle filter (APF), also known as the auxiliary SIR (ASIR) filter, proposed by Pitt and Shephard (1999). However, it is worth mentioning that one of the earliest sequential Monte Carlo algorithms was proposed by West (1992). For recent discussion regarding the similarities and differences between BF and APF see, for instance, Carvalho, Johannes, Lopes and Polson (2010), Doucet and Johansen (2008) and Douc, Moulines and Olsson (2009).

Propagate-resample filters. The BF of Gordon *et al.* (1993) is based on sequential SIR steps over time (Smith and Gelfand, 1992). The Kalman recursions from (14) and (15) are combined

in

$$p(x_t, x_{t-1}|y_t, y^{t-1}) \quad \propto \quad \underbrace{p(y_t|x_t)}_{2.Resample} \underbrace{p(x_t|x_{t-1})p(x_{t-1}|y^{t-1})}_{1.Propagate} \tag{16}$$

In words, the BF first propagates particles from the posterior at time $t-1$ in order to generate particles from the prior at time t. Then it resamples the propagated particles with weights proportional to their likelihoods. This is Algorithm 1 below, whose recursions are illustrated in Figure 1.

Resample-propagate filters. Similarly, the APF first resamples particles from the posterior at time $t-1$ with weights taking into account the next observed data point, y_t. Then it propagates the resampled particles. The identity from equation (16) is rewritten as

$$p(x_t, x_{t-1}|y_t, y^{t-1}) \quad \propto \quad \underbrace{p(x_t|x_{t-1}, y^t)}_{2.Propagate} \underbrace{p(y_t|x_{t-1})p(x_{t-1}|y^{t-1})}_{1.Resample}. \tag{17}$$

The main difficulty with the APF is that in most applications, neither $p(y_t|x_{t-1})$ is available for pointwise evaluation (resampling) nor $p(x_t|x_{t-1}, y^t)$ is available for sampling (propagation). The APF is *fully adapted* when these conditions are satisfied. The main suggestion in Pitt and Shephard (1999) for general state space models is:

a) Use $p(y_t|g(x_{t-1}))$, i.e. the data density $p(y_t|x_t)$ evaluated at $g(x_{t-1})$ (usually the expected value, median or mode of the state transition density $p(x_t|x_{t-1})$) as the proposal weight to resample the old particle x_{t-1}; and

b) Use $q(x_t|x_{t-1}, y_t) \equiv p(x_t|x_{t-1})$ as the proposal density to propagate resampled particles to the new set of particles $\{x_t^{(i)}\}_{i=1}^N$. Notice that here $q(\cdot)$ is blind since it does not incorporate the current observations y_t. See below for more details on better ways of choosing $q(\cdot)$.

Since both resampled and propagated particles come from proposal densities, it follows directly from a simple SIR argument that these particles have weights given by

$$\begin{aligned} w_t \quad &\propto \quad \frac{p(y_t|x_t)p(x_t|x_{t-1})p(x_{t-1}|y^{t-1})}{p(y_t|g(x_{t-1}))p(x_t|x_{t-1})p(x_{t-1}|y^{t-1})} \\ &= \quad \frac{p(y_t|x_t)}{p(y_t|g(x_{t-1}))}. \end{aligned} \tag{18}$$

This leads to Algorithm 2 below.

Algorithm 1: Bootstrap filter (BF)

1. Propagate $\{x_{t-1}^{(i)}\}_{i=1}^N$ to $\{\tilde{x}_t^{(i)}\}_{i=1}^N$ via $p(x_t|x_{t-1})$;
2. Resample $\{x_t^{(i)}\}_{i=1}^N$ from $\{\tilde{x}_t^{(i)}\}_{i=1}^N$ with weights $w_t^{(i)} \propto p(y_t|\tilde{x}_t^{(i)})$.

Algorithm 2: Auxiliary particle filter (APF)

1. Resample $\{\tilde{x}_{t-1}^{(i)}\}_{i=1}^N$ from $\{x_{t-1}^{(i)}\}_{i=1}^N$ with weights $w_t^{(i)} \propto p(y_t|g(x_{t-1}^{(i)}))$.
2. Propagate $\{\tilde{x}_{t-1}^{(i)}\}_{i=1}^N$ to $\{\tilde{x}_t^{(i)}\}_{i=1}^N$ via $p(x_t|\tilde{x}_{t-1})$;
3. Resample $\{x_t^{(i)}\}_{i=1}^N$ from $\{\tilde{x}_t^{(i)}\}_{i=1}^N$ with weights $w_t^{(i)} \propto p(y_t|\tilde{x}_t^{(i)})/p(y_t|g(\tilde{x}_{t-1}^{(i)}))$.

Algorithms 3 and 4 below are the optimal and fully adapted versions of BF and APF when $p(y_t|x_{t-1})$ is analytically tractable and $p(x_t|x_{t-1}, y^t)$ easy to sample from.

Algorithm 3: Optimal bootstrap filter (OBF)

1. Propagate $\{x_{t-1}^{(i)}\}_{i=1}^N$ to $\{\tilde{x}_t^{(i)}\}_{i=1}^N$ via $p(x_t|x_{t-1}, y_t)$;
2. Resample $\{x_t^{(i)}\}_{i=1}^N$ from $\{\tilde{x}_t^{(i)}\}_{i=1}^N$ with weights $w_t^{(i)} \propto p(y_t|x_{t-1}^{(i)})$.

Algorithm 4: Optimal auxiliary particle filter (OAPF)

1. Resample $\{\tilde{x}_{t-1}^{(i)}\}_{i=1}^N$ from $\{x_{t-1}^{(i)}\}_{i=1}^N$ with weights $w_t^{(i)} \propto p(y_t|x_{t-1}^{(i)})$;
2. Propagate $\{\tilde{x}_{t-1}^{(i)}\}_{i=1}^N$ to $\{x_t^{(i)}\}_{i=1}^N$ via $p(x_t|\tilde{x}_{t-1}, y_t)$.

Choosing the proposal. Improvements on the basic particle-filter algorithm include the use of better proposal distributions in the importance sampling stage (Pitt and Shephard, 1999), the use of Markov chain Monte Carlo sampling (Gilks and Berzuini, 2001; Fearnhead, 2002a Fearnhead and Clifford, 2003) and the use of resampling (Liu and Chen, 1995; Carpenter *et al.*, 1999), which can be important to avoid having only a small number of particles with non-negligible weight.

Pitt and Shephard (1999) suggest local linearization of the observation equation (via an extended Kalman filter-type approximation) in order to construct a proposal propagation density, say $q(x_t|x_{t-1}, y^t)$, for the OAPF propagation density $p(x_t|x_{t-1}, y^t)$, that takes into account the current observation y_t and, potentially, outperforms the naïve blind propagation proposal density $p(x_t|x_{t-1})$. See Liu and Chen (1995), Carpenter, Clifford and Fearnhead (1999), Gilks and Berzuini (2001), Doucet, Godsill and Andrieu (2000), Fearnhead (2002) and Guo, Wang and Chen (2005), amongst others, for additional discussion on the choice of $q(x_t|x_{t-1}, y_t)$. See Chen and Lai (2007) for an interesting application of on-line identification and adaptive control of autoregressive models with exogenous inputs (ARX models) with Markov parameter jumps. More efficient proposal densities can be obtained in the presence of conditional linearity

and/or normality. In other words, when the split of the state vector x_t into x_{1t} and x_{2t} leads to, say, $x_{1t}|x_{2t}$ being a NDLM, then part of the sequential learning algorithm can be performed exactly by analytically integrating out x_{1t}. Such filters are commonly refereed to as the *Rao-Blackwellized particle filter* or *mixture Kalman filter* (Chen and Liu, 2000, Andrieu and Doucet, 2002).

Resampling or not? It has been argued that the resampling step in the BF and the second resampling step in the APF should only be performed when *particle degeneracy* is signaled. For instance, Kong, Liu and Wong (1994) introduced the *effective sample size*, $\mathrm{N_{eff}}$, which they estimate by

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N}(w_t^{(i)})^2}. \tag{19}$$

The particle set that approximates $p(x_t|y^t)$ is then represented by $\{(\tilde{x}_t, w_t)^{(i)}\}_{i=1}^N$, using the notation from Algorithms 1 and 2 above.

Reducing MC error. Regardless of whether resampling is performed at each time period or not, when the goal is to produce summary statistics based on the posterior $p(x_t|y^t)$ (for instance, mean, variance, quantiles, etc.), it is more efficient (estimator with lower variance) to perform the computation prior to resampling. For instance, it is more efficient to estimate $E(x_t|y^t)$ by $\sum_{i=1}^N \omega_t^{(i)}\tilde{x}_t^{(i)} / \sum_{j=1}^N w_t^{(j)}$ than by $\sum_{i=1}^N \tilde{x}_t/N$.

Example 1 (Local level model). In this example, we use the local level model to compare the performance of the four particle-filter algorithms discussed above. They are the BF, APF, OBF and OAPF. As mentioned in Section 2, the local level model is $y_t|x_t \sim N(x_t, \sigma^2)$ and $x_t|x_{t-1} \sim N(x_{t-1}, \tau^2)$. For this simple linear model, the traditional Kalman filter is available to produce the "optimal" estimate of the filtered state vector. We use this estimate in evaluating the performance of particle filters.

Based on results of Section 2, it is easy to see that (i) $x_t|y^t \sim N(m_t, A_t\sigma^2)$, where $m_t = (1-A_t)m_{t-1} + A_t y_t$ and $A_t = (A_{t-1}\sigma^2 + \tau^2)/(A_{t-1}\sigma^2 + \tau^2 + \sigma^2)$, (ii) $y_t|x_{t-1} \sim N(x_{t-1}, \sigma^2 + \tau^2)$, and (iii) $x_t|x_{t-1}, y_t \sim N(\omega^2(x_{t-1}/\tau^2 + y_t/\sigma^2), \omega^2)$, where $\omega^{-2} = \sigma^{-2} + \tau^{-2}$. Thus, the four particle-filter algorithms are easy to implement. To compare the filters, we employ the criterion of mean square errors, which are computed using R runs of each particle filter f in $\{\text{BF}, \text{OBF}, \text{APF}, \text{OAPF}\}$, across M time-series of length T. Specifically, the MSE is given by $MSE_f = \sum_{t,m,r}(\hat{x}_{ftmr} - \tilde{x}_{tm})^2/(TMR)$, where $\tilde{x}_{tm}$ is obtained via the standard Kalman filter recursions (Eqs. 3 to 8) for the m-th dataset up to time t, and $\hat{x}_{ftmr} = \sum_{i=1}^N x_{ftmr}^{(i)}/N$ is the particle approximation for $\tilde{x}_{tm}$ based on N particles $\{x_{ftmr}^{(i)}\}_{i=1}^N$.

The relative MSE, relative to the bootstrap filter, is defined as $RMSE_f = MSE_f/MSE_{BF}$ for f in $\{\text{OBF}, \text{APF}, \text{OAPF}\}$. Results are summarized in Figure 2. From the plots, OAPF outperforms OBF for all four values of τ^2 and OBF fares better than BF for all four values of τ^2. Also, it seems that BF performs much better than APF when the signal to noise ratio, τ/σ, is greater than one. □

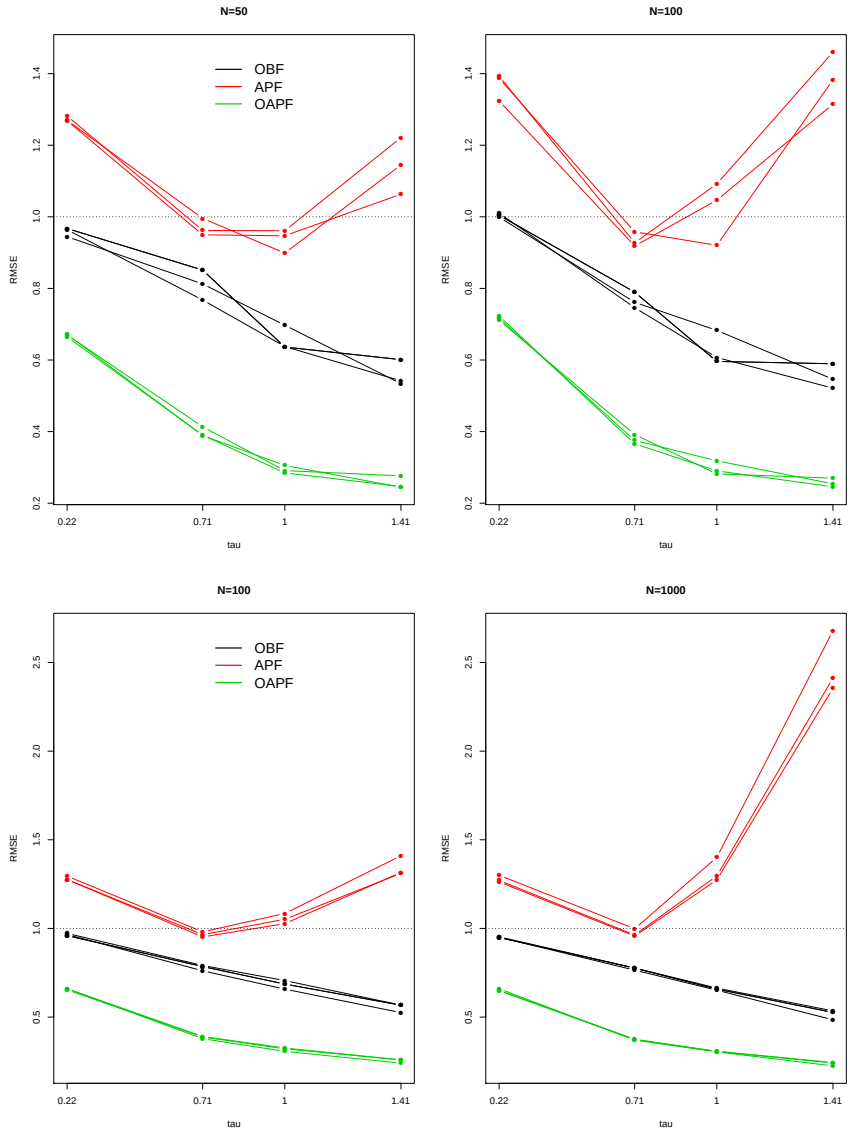

Figure 2: *Comparison between BF, OBF, APF and OAPF via relative mean square error (RMSE).* Local level model is used, where $y_t|x_t \sim N(x_t, \sigma^2)$ and $x_t|x_{t-1} \sim N(x_{t-1}, \tau^2)$, for $t = 1, \ldots, T$, $x_0 \sim N(m_0, C_0)$, $\sigma = 1$, $\tau = 0.22$, 0.71, 1.0 or 1.41, $m_0 = 0$ and $C_0 = 10$. The starting value is $x_0 = 0$ and N denotes the number of particles. RMSE is based on $M = 10$ time series of length T and $R = 10$ runs of each particle filter per time series. Top row: sample size $T = 100$; Bottom row: $T = 1000$.

Review papers. Since Gordon, Salmond and Smith (1993), several review papers have contributed to straightening out the sub-area of sequential Monte Carlo. Here we list a small subset of these papers. The choice is rather subjective and based on our limited and biased views of the field. A few of the early reviews are Doucet, De Freitas and Gordon (2000) and Arulampalam, Maskell, Gordon and Clapp (2002), the 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 ones, along with this paper, are Cappé, Godsill and Moulines (2007), Doucet and Johansen (2009) and Prado and West (2010, ch. 6). 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.

4 Parameter learning

Consider again the general dynamic model. We now address explicitly the unknown vector of static parameters θ of the model:

$$y_t|x_t,\theta \sim p(y_t|x_t,\theta), \tag{20}$$
$$x_t|x_{t-1},\theta \sim p(x_t|x_{t-1},\theta), \tag{21}$$

for $t = 1,\ldots,T$ and initial probability density $p(x_0|\theta)$ and prior $p(\theta)$. There are primarily two ways to tackle the problem of learning θ: batch sampling and online sampling.

Batch sampling. The solution involves obtaining an approximation, say $p^N(y^T|\theta)$, to the joint likelihood $p(y^T|\theta)$. For the NDLM of Section 2, the predictive density was obtained analytically from Eq. (13). The approximation $p^N(y^T|\theta)$ can be obtained by any of the previous filters as

$$p^N(y^T|\theta) = \prod_{t=1}^{T} p^N(y_t|y^{t-1},\theta) = \frac{1}{N^T}\prod_{t=1}^{T}\sum_{i=1}^{N} p(y_t|x_t^{(i)},\theta), \tag{22}$$

where $x_t^{(i)} \sim p(x_t|x_{t-1}^{(i)},\theta)$, for $i = 1,\ldots,N$. Therefore, the components of θ can be sampled iteratively via a standard MCMC sampler, such as a Metropolis-Hastings algorithm, or via a SIR step. Two of the major drawbacks of this solution are: 1) SMC loses its appealing sequential nature and 2) the overall MCMC or SIR scheme can be highly sensitive to the approximation $p^N(y^T|\theta)$. See, for instance, Chopin (2002) and Del Moral *et al.* (2006) for more theoretical justifications and further details, Doucet and Tadic (2003), Andrieu, Doucet and Singh (2004), Poyiadjis, Doucet and Singh (2005), Andrieu, Doucet and Tadic (2005) and Olsson, Cappé, Douc and Moulines (2006) for expectation-maximization-like schemes, and Fernández-Villaverde and Rubio-Ramírez (2005, 2007) and DeJong *et al.* (2009) for applications in dynamic stochastic general equilibrium macroeconomic models.

Particle filters and MCMC. Before introducing particle filters that learn about parameters in a sequential manner, we should mention that 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.

Online sampling. The solution here is to produce sequential MC approximations to $p(x_t, \theta|y^t)$, sometimes $p(x_{t-1}, x_t, \theta|y^t)$ and/or other small dimensional functions (small compared to t) of (x^t, θ) conditional on y^t. Simply resampling θ over time is bound to fail since, in general, after a few time steps the particle set will contain only one particle. Gordon, Salmond and Smith (1993) suggest incorporating artificial evolution noise for θ when tackling the problem of sequentially learning the static parameters of a state space model. Since parameters are not states, adding noise artificially will eventually distort and compromise the validity of the approximated posterior distributions. Their approach imposes a loss of information in time as artificial uncertainties added to the parameters eventually result in a very diffuse posterior density for θ. In what follows, we introduce three well established filters for sequentially learning both x_t and θ: (i) The Liu and West filter; (ii) The Storvik filter; and (iii) The *particle learning* (PL) filter of Carvalho *et al.* (2010) and Lopes *et al.* (2010).

4.1 Liu and West filter

Liu and West (2001) combine (i) the APF of Pitt and Shephard (1999), (ii) a kernel smoothing approximation to $p(\theta|y^{t-1})$ via a mixture of multivariate normals, and (iii) a neat shrinkage idea to incorporate artificial evolution for θ without the associated loss of information; see West (1993a,b). More specifically, let the set of i.i.d. particles $\{x_{t-1}^{(i)}, \theta_{t-1}^{(i)}\}_{i=1}^N$ approximate $p(x_{t-1}, \theta|y^{t-1})$ such that $p(\theta|y^{t-1})$ can be approximated by

$$p^N(\theta|y^{t-1}) = \frac{1}{N}\sum_{j=1}^{N} f(\theta; m^{(j)}, V) \tag{23}$$

where $m^{(j)} = a\theta_{t-1}^{(j)} + (1-a)\bar{\theta}$, $\bar{\theta} = \sum_{j=1}^N \theta_{t-1}^{(j)}/N$, $V = h^2 \sum_{j=1}^N (\theta_{t-1}^{(j)} - \bar{\theta})(\theta_{t-1}^{(j)} - \bar{\theta})'/N$ and $h^2 = 1 - a^2$. The subscript t of θ_t is used only to indicate that samples are from $p(\theta|y^t)$. The APF of Pitt and Shephard (1999) of Eq.(17) can now be written for the state vector (x_t, θ_t) as

$$\begin{aligned} p(x_t, x_{t-1}, \theta_t, \theta_{t-1}|y_t, y^{t-1}) &= \underbrace{p(y_t|x_{t-1}, \theta_{t-1})p(x_{t-1}|\theta_{t-1}, y^{t-1})p(\theta_{t-1}|y^{t-1})}_{1.Resample} \\ &\times \underbrace{p(x_t|x_{t-1}, \theta_t, y^t)p(\theta_t|\theta_{t-1}, y^t)}_{2.Propagate}. \end{aligned} \tag{24}$$

In general and similar to the APF filter of Section 3, $p(y_t|x_{t-1}, \theta)$ is not available for pointwise evaluation and/or $p(x_t|x_{t-1}, \theta_t, y^t)$ is not easy to sample from. Liu and West resample

old particles with weights proportional to $p(y_t|g(x_{t-1}), m(\theta_{t-1}))$, where $g(\cdot)$ and $m(\cdot)$ are described above. Then, they propagate θ_t from the proposal propagation density $p(\theta_t|\theta_{t-1})$ and propagate x_t conditional on θ_t from the evolution density $q(x_t|x_{t-1}, \theta, y^t) \equiv p(x_t|x_{t-1}, \theta_t)$. The propagated particles (x_t, θ_t) have associated weights

$$\tilde{\omega}_t \propto \frac{p(y_t|x_t, \theta_t)}{p(y_t|g(x_{t-1}), m(\theta_{t-1}))},$$

which leads to Algorithm 5 below.

The performance of the LW filter depends on the choice of the tuning parameter a, which drives both the shrinkage and the smoothness of the normal approximation. It is common practice to set a around 0.98 or higher. The components of θ can be either transformed in order to accommodate the approximate local normality or the multivariate normal approximation could be replaced by a composition of, say, conditionally normal densities for location parameters and inverse-gamma densities for scale/variance parameters. See, for example, Petris *et al.* (2009, pp. 222 - 228) for an example based on the local level model and Carvalho and Lopes (2007) for an application to Markov switching stochastic volatility models.

Example 2: Stochastic volatility model. In its simplest form, asset returns y_t are modeled as conditionally independent normal random variables with log-variance x_t following a first-order autoregressive model, i.e. $y_t|x_t \sim N(0, \exp\{x_t\})$ and $x_t|x_{t-1} \sim N(\alpha + \beta x_{t-1}, \tau^2)$; see Jacquier, Polson and Rossi (1994) and Kim, Shephard and Chib (1998). One possible version of the LW filter assumes, for example, that $\theta = (\alpha, \beta, \log \tau^2)$ and $g(x_{t-1}) = \alpha + \beta x_{t-1}$. □

Algorithm 5: Liu and West filter (LWF)

1. Resample $\{(\tilde{x}_{t-1}, \tilde{\theta}_{t-1})^{(i)}\}_{i=1}^N$ from $\{(x_{t-1}, \theta_{t-1})^{(i)}\}_{i=1}^N$ with weights $w_t^{(i)} \propto p(y_t|g(x_{t-1}^{(i)}), m^{(i)})$, where $m^{(i)}$ is defined in Eq. (23).
2. Propagate
   (a) $\{\tilde{\theta}_{t-1}^{(i)}\}_{i=1}^N$ to $\{\hat{\theta}_t^{(i)}\}_{i=1}^N$ via $N(\tilde{m}^{(i)}, V)$, then
   (b) $\{\tilde{x}_{t-1}^{(i)}\}_{i=1}^N$ to $\{\hat{x}_t^{(i)}\}_{i=1}^N$ via $p(x_t|\tilde{x}_{t-1}^{(i)}, \hat{\theta}_t^{(i)})$.
3. Resample $\{(x_t, \theta_t)^{(i)}\}_{i=1}^N$ from $\{(\hat{x}_t, \hat{\theta})^{(i)}\}_{i=1}^N$ with weights $w_t^{(i)} \propto \frac{p(y_t|\hat{x}_t^{(i)}, \hat{\theta}_t^{(i)})}{p(y_t|g(\tilde{x}_{t-1}^{(i)}), \tilde{m}^{(i)})}$.

4.2 Storvik filter

For the class of state-space models where $p(\theta|x^t, y^t)$ can be rewritten as $p(\theta|s_t)$, where s_t is a low-dimensional set of conditionally sufficient statistics and can be recursively computed via $s_t = \mathcal{S}(s_{t-1}, x_t, y_t)$, Storvik (2002) (see also Fearnhead, 2002) proposed Algorithm 6 below. This algorithm can be thought of as an extension of the bootstrap filter with the additional steps of sequentially updating the sufficient statistics and sampling θ.

Algorithm 6: Storvik Filter (SF)

1. Propagate $\{x_{t-1}^{(i)}\}_{i=1}^N$ to $\{\tilde{x}_t^{(i)}\}_{i=1}^N$ via $q(x_t|x_{t-1}, \theta, y^t)$.
2. Resample $\{(x_t, s_{t-1})^{(i)}\}_{i=1}^N$ from $\{(\tilde{x}_t, s_{t-1})^{(i)}\}_{i=1}^N$ with weights
$$w_t^{(i)} \propto \frac{p(y_t|\tilde{x}_t^{(i)}, \theta)p(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, \theta)}{q(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, \theta, y^t)}.$$
3. Compute sufficient statistics: $s_t = \mathcal{S}(s_{t-1}, x_t, y_t)$.
4. Sample θ from $p(\theta|s_t)$.

Example 2 (continued). The stochastic volatility model admits recursive sufficient statistics for $\theta = (\alpha, \beta, \tau^2)$ when $p(\theta)$ is conditionally conjugate normal-inverse gamma. More precisely, when $(\alpha, \beta|\tau^2) \sim N(b_0, \tau^2 B_0)$ and $\tau^2 \sim IG(c_0, d_0)$, it easy to see that $(\alpha, \beta|\tau^2, x^t) \sim N(b_t, \tau^2 B_t)$ and $(\tau^2|x^t) \sim IG(c_t, d_t)$, where $B_t^{-1} = B_{t-1}^{-1} + z_t z_t'$, $B_t^{-1} b_t = B_{t-1}^{-1} b_{t-1} + x_t z_t$, $c_t = c_{t-1} + 1/2$, $d_t = d_{t-1} + (x_t - b_t' z_t)x_t/2 + (b_{t-1} - b_t)' B_{t-1}^{-1} b_{t-1}/2$ and $z_t' = (1, x_{t-1})$. The recursive sufficient statistics are functions of x_{t-1}, x_{t-1}^2, $x_{t-1}x_t$ and x_t^2.

All simulated exercises in Storvik (2002) are based on a blind propagation rule, i.e. $q(x_t|x_{t-1}, \theta, y^t)$ above is equal to $p(x_t|x_{t-1}, \theta)$. In this case, resampling is performed with weights $w_t \propto p(y_t|x_t, \theta)$. Like any other PF with blind propagation, such as the bootstrap filter, this filter is bound to suffer from particle degeneracy, which in turn directly compromises sequential parameter estimation.

4.3 Particle learning

Carvalho *et al.* (2010) present methods for sequential filtering, particle learning (PL) and smoothing for rather general 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. Recall the simulated exercise from Section 3 that empirically shows that resample-propagate filters tend to outperform propagate-resample ones. 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 T.

Let s_t and s_t^x denote the parameter and state sufficient statistics satisfying deterministic updating rules $s_t = \mathcal{S}(s_{t-1}, x_t, y_t)$, as in the Storvik filter from the previous subsection, and $s_t^x = \mathcal{K}(s_{t-1}^x, \theta, y_t)$, for $\mathcal{K}(\cdot)$ mimicking the Kalman filter recursions (see Example 3 below). Then PL can be described as follows.

Algorithm 7: Particle learning (PL)

1. Resample $(\tilde{\theta}, \tilde{s}^x_{t-1}, \tilde{s}_{t-1})$ from $(\theta, s^x_{t-1}, s_{t-1})$ with weights $w_t \propto p(y_t|s^x_{t-1}, \theta)$.
2. Sample x_t from $p(x_t|\tilde{s}^x_{t-1}, \tilde{\theta}, y^t)$.
3. Update parameter sufficient statistics: $s_t = \mathcal{S}(\tilde{s}_{t-1}, x_t, y_t)$.
4. Sample θ from $p(\theta|s_t)$.
5. Update state sufficient statistics: $s^x_t = \mathcal{K}(\tilde{s}^x_{t-1}, \theta, y_t)$.

In many cases $\mathcal{S}$ will also be a function of x_{t-1} and possibly other lags of the state variable, such as in the stochastic volatility model of Example 2. In these cases, the above algorithm is slightly changed and particles for such lagged values are also carried over time. Therefore, step 2 is modified to sample (x_{t-1}, x_t) from $p(x_{t-1}, x_t|s^x_{t-1}, \theta, y^t)$, which implies sampling x_{t-1} from $p(x_{t-1}|s^x_{t-1}, \theta, y^t)$ and x_t from $p(x_t|x_{t-1}, \theta, y^t)$.

Example 3 (Conditional NDLM). Carvalho *et al.* (2010) derive the PL scheme for the class of conditional NDLM defined by the observation and evolution equations that assume the form of a linear system (see the NDLM Eqs. (1) and (2)) conditional on an auxiliary state λ_t

$$y_t = F'_{\lambda_t} x_t + v_t, \qquad v_t \sim N(0, \sigma^2_{\lambda_t}), \tag{25}$$

$$x_t = G_{\lambda_t} x_{t-1} + w_t, \qquad w_t \sim N(0, \tau^2_{\lambda_t}), \tag{26}$$

with the quadruple $\{F_{\lambda_t}, G_{\lambda_t}, \sigma^2_{\lambda_t}, \tau^2_{\lambda_t}\}$ being a function of the static parameter vector θ. The marginal distributions of the observation error and state shock distributions are any combination of normal, scale mixture of normals, or discrete mixture of normals depending on the specification of the distribution of the auxiliary state variable $p(\lambda_{t+1}|\theta)$ (Chen and Liu, 2000). Extensions to hidden Markov specifications where λ_{t+1} evolves according to $p(\lambda_{t+1}|\lambda_t, \theta)$ are discussed in Carvalho *et al.* (2010). In the case where the auxiliary state variable λ_t is discrete, such as in stochastic volatility with jumps models (Markovian or not), the state x_{t-1} can be analytically integrated out, in addition to x_t and λ_t, at the initial resampling step, i.e. $p\left(y_t|(\lambda_{t-1}, s^x_{t-1}, \theta)^{(i)}\right) = \sum_{\lambda_t} p(y_t|\lambda_t, (s^x_{t-1}, \theta)^{(i)}) p(\lambda_t|(\lambda_{t-1}, \theta)^{(i)})$, where the conditional sufficient statistics for states (s^x_t) and parameters (s_t) satisfy the deterministic updating rules $s_t = \mathcal{S}\left(s_{t-1}, x_t, \lambda_t, y_t\right)$ and $s^x_t = \mathcal{K}\left(s^x_{t-1}, \theta, \lambda_t, y_t\right)$, where $\mathcal{S}(\cdot)$ denotes, as defined previously, the recursive update of the parameter sufficient statistics and $\mathcal{K}(\cdot)$ denotes the Kalman filter recursions of the conditional NDLM given in Eqs. (3) to (8). The algorithm can be summarized as follows:

1. resample $(\tilde{\lambda}_{t-1}, \tilde{\theta}, \tilde{s}^x_{t-1}, \tilde{s}_{t-1})$ from $(\lambda_{t-1}, \theta, s^x_{t-1}, s_{t-1})$ with weights $w_t \propto p(y_t|\lambda_{t-1}, s^x_{t-1}, \theta)$;
2. sample λ_t from $p(\lambda_t|\tilde{\lambda}_{t-1}, \tilde{\theta}, y^t)$;
3. sample x_t from $p(x_t|\lambda_t, \tilde{s}^x_{t-1}, \tilde{\theta}, y^t)$;
4. compute $s_t = \mathcal{S}(s_{t-1}, x_t, \lambda_t, y_t)$;

5. sample θ from $p(\theta|s_t)$; and

6. compute $s_t^x = \mathcal{K}(\lambda_t, s_{t-1}^x, \theta, y_t)$.

In the case where the auxiliary state variable λ_t is continuous, the authors extend the above scheme by adding to the current particle set a propagated particle $\lambda_{t+1} \sim p(\lambda_{t+1}|(\lambda_t, \theta)^{(i)})$. □

PL in time series models. Successful implementations of PL (and hybrid versions of PL) have appeared over the last couple of years. Rios and Lopes (2009), for example, propose a hybrid LW-Storvik filter for the Markov switching stochastic volatility model that outperforms the Carvalho and Lopes (2007) filter. Lund and Lopes (2009) sequentially estimate a regime-switching macro-finance model for the postwar US term-structure of interest rates, while Prado and Lopes (2009) adapt PL to study state-space autoregressive models with structured priors. Chen, Petralia and Lopes (2009) propose a hybrid PL-LW sequential MC algorithm that fully estimates non-linear, non-normal dynamic to stochastic general equilibrium models, with a particular application in a neoclassical growth model. Additionally, Dukic, Lopes and Polson (2009) use PL to track flu epidemics using Google trends data, while Lopes and Polson (2010) use PL to estimate volatility and examine volatility dynamics for financial time series, such as the S&P500 and the NDX100 indices, during the early part of the credit crisis.

Sequential Bayesian computation via PL. Lopes *et al.* (2010) develop a simulation-based approach to sequential Bayesian computation for both dynamic and non-dynamic systems. They show through various important applications that PL provides a simple yet powerful framework for efficient sequential posterior sampling strategies. For example, Carvalho *et. al.* (2009) adapt PL to a rich and general class of mixture models that include finite mixture models and Dirichlet process mixture models, as well as for the less common settings of latent feature selection through an Indian Buffet process and dependent distribution tracking through a probit stick breaking model. Taddy, Gramacy and Polson (2010) show that PL is the best alternative to perform online posterior filtering of tree-states in dynamic regression tree models, while Gramacy and Polson (2010) use PL for online updating of Gaussian process models for regression and classification. Shi and Dunson (2009) adopt PL for stochastic variable selection and model search in linear regression and probit models, while Mukherjee and West (2009) focus on model comparison for applications in cellular dynamics in systems biology.

Example 4 (Comparison between LW, Storvik and PL). We compare the performance of these particle filters using the local level model of Example 1 with parameter learning, where the random walk system equation is replaced by a first-order autoregression. More precisely, $y_t|x_t, \theta \sim N(x_t, \sigma^2)$ and $x_t|x_{t-1}, \theta \sim N(\alpha + \beta x_{t-1}, \tau^2)$, where $x_0 \sim N(m_0, C_0)$ and $\theta = (\alpha, \beta, \tau^2, \sigma^2)$. The prior distribution of θ is $p(\theta) = p(\sigma^2)p(\tau^2)p(\alpha, \beta|\tau^2)$, where $\sigma^2 \sim IG(n_0/2, n_0\sigma_0^2/2)$, $\tau^2 \sim IG(\nu_0/2, \nu_0\tau_0^2/2)$ and $(\alpha, \beta) \sim N(b_0, \tau^2 B_0)$. The recursive sufficient statistics for θ can easily be derived. It can be shown that $\beta|(\tau^2, x^t) \sim N(b_t, \tau^2 B_t)$ and $\tau^2|x^t \sim IG(\nu_1/2, \nu_1\tau_1^2/2)$, where $\nu_1 = \nu_0 + t$, $B_t^{-1} = B_0^{-1} + Z_t'Z_t$, $B_t^{-1}b_t = B_0^{-1}b_0 + Z_t^T z_t$, and $\nu_1\tau_1^2 = \nu_0\tau_0^2 + (z_t - X_t b_t)'z_t + (b_0 - b_t)'B_0^{-1}b_0$, for $z_t = (x_1, \ldots, x_t)'$, $Z_t = (1^t, Z_{2t})$,

$Z_{2t} = (x_0, \ldots, x_{t-1})'$ and 1^t a t-dimensional vector of ones. The quantities $(n_t, n_t\tau_t^2, b_t, B_t)$ can be rewritten recursively as functions of $(\nu_{t-1}, \nu_{t-1}\tau_{t-1}^2, b_{t-1}, B_{t-1})$, x_{t-1}, x_t and y_t. A time series of length $T = 200$ was simulated using $\theta = (0.0, 0.9, 0.5, 1.0)$ and $x_0 = 0$. The prior hyperparameters are $m_0 = 0$, $C_0 = 10$, $b_0 = (0.0, 0.9)'$, $B_0 = I_2$, $n_0 = \nu_0 = 10$, $\tau_0^2 = 0.5$ and $\sigma_0^2 = 1.0$, leading to relatively uninformative prior information. The performance of the filters is assessed by running each algorithm for $R = 100$ times based on $N = 1000$ particles. A very long PL ($N = 100,000$) is run to serve as a benchmark for comparison. Let $q(\gamma, \alpha, t)$ be the 100αth percentile of $p(\gamma|y^t)$, where γ is an element of θ. We define the root mean squared error as the square root of $MSE(\gamma, \alpha, f, t) = \sum_{t,r}[q(\gamma, \alpha, t) - q_{fr}(\gamma, \alpha, t)]^2/R$ for filter f in {LW,STORVIK,PL} and replication $r = 1, \ldots, R$. Finally, a full adaptation is implemented for the three filters. In other words, LW differs from PL only through the sequential estimation of θ, Storvik differs from PL only to the extent that Storvik propagates first and then resamples, while PL resamples first and then propagates. Results are summarized in Figures 3 and 4. Both the Storvik filter and PL are significantly better than the LW filter, while PL is moderately better than Storvik, particularly when estimating the pair (τ^2, σ^2). $\square$

4.4 Smoothing

In addition to delivering sequential filtering for parameters and states, particle filters can also be implemented effectively when the main goal is smoothing the states conditional on the whole vector of observations y^T. In this sense, *particle smoothers* are alternatives to MCMC in state-space models (Kitagawa, 1996). Godsill, Doucet and West (2004) introduced an algorithm that relies on (i) forward particle sampling and (ii) backward particle resampling. Carvalho *et al.* (2010) extend the algorithm to accommodate sequential learning of the parameter vector θ. In this more general case, it can be shown that

$$p(x^T, \theta|y^T) = \left\{\prod_{t=1}^{T-1} p(x_t|x_{t+1}, \theta, y^t)\right\} p(x_T, \theta|y^T), \tag{27}$$

whose components, by Bayes rule and conditional independence, are

$$p(x_t|x_{t+1}, \theta, y^t) \propto p(x_{t+1}|x_t, \theta)p(x_t|\theta, y^t). \tag{28}$$

This leads to a backward sampling algorithm that resamples forward particles x_t from $p(x_t|\theta, y^t)$ with weights proportional to $p(x_{t+1}|x_t, \theta)$. More precisely, for each particle i, for $i = 1, \ldots, N$, start with $(\tilde{x}_T, \tilde{\theta})^{(i)} = (x_T, \theta)^{(i)}$, i.e. a draw from $p(x_T, \theta|y^T)$. Then, for $t = T-1, \ldots, 1$, sample $\tilde{x}_t^{(i)}$ from $\{x_t^{(j)}\}_{j=1}^N$ with weights $\pi_t^{(j)} \propto p(\tilde{x}_{t+1}|x_t^{(j)}, \tilde{\theta})$. In the end, $(\tilde{x}_1, \ldots, \tilde{x}_T)^{(i)}$ are draws from $p(x^T|y^T)$ for $i = 1, \ldots, N$. Notice that the algorithm is $O(TN^2)$ so the computational time to obtain draws from $p(x^T|y^T)$ is expected to be much larger than the computational time to obtain draws from $p(x_t|y^t)$ via standard SMC filters for $t = 1, \ldots, T$. See Briers, Doucet and Maskell (2010) for an alternative $O(TN^2)$ SMC smoother for the case where θ is known. An $O(TN)$ smoothing algorithm has recently been introduced by Fearnhead, Wyncoll and Tawn (2008) also for the case where θ is known. See also Douc, Garivier, Moulines and Olsson (2009) for additional FFBS particle approximations.

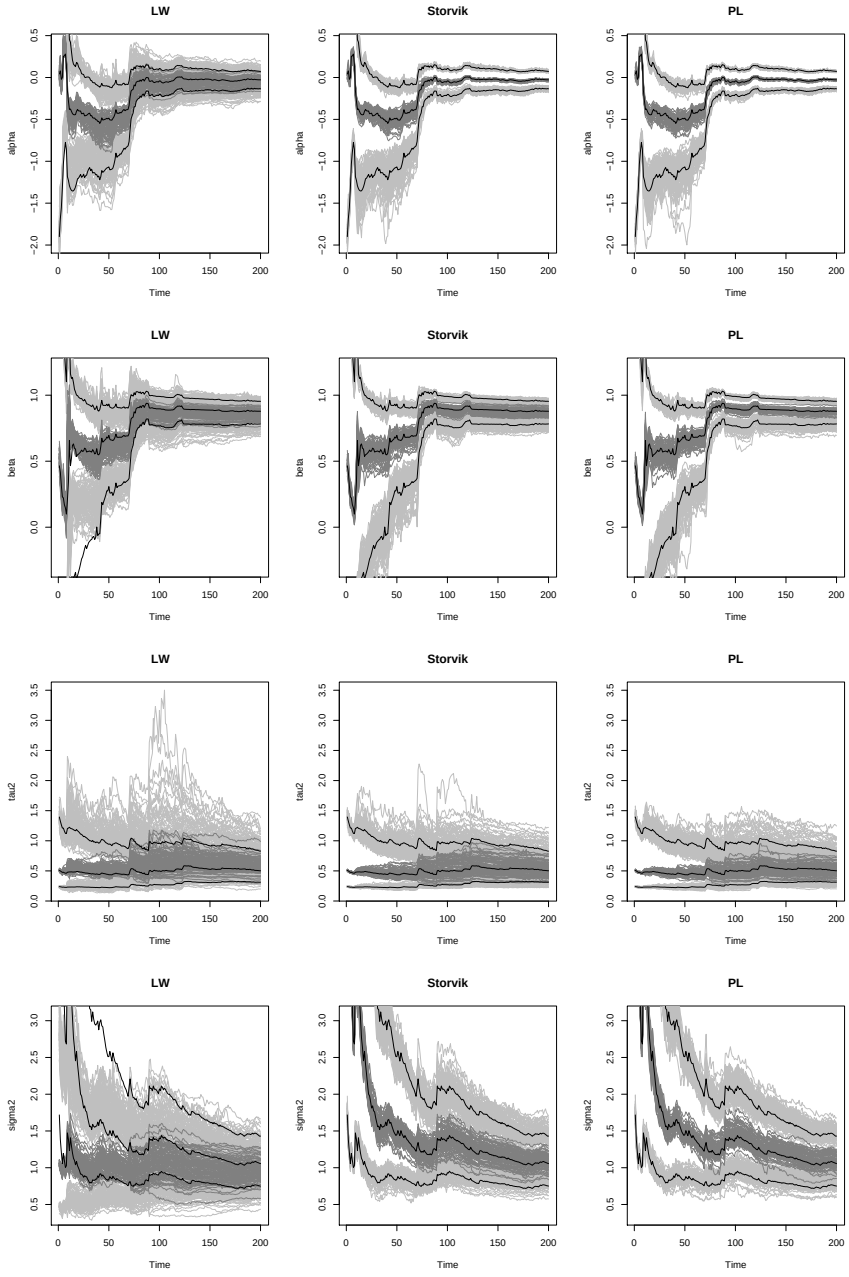

Figure 3: *Comparison between LWF, SF and PL.* Percentiles of $p(\theta|y^t)$ (2.5th, 50th and 97.5th) based on 100 replications of each particle filter with $N = 1000$ particles (grey lines). Black lines are based on PL and $N = 100,000$. Liu and West filter (left column), Storvik's filter (center column) and particle learning (right column). The row (from top to bottom) represents the components of $\theta = (\alpha, \beta, \tau^2, \sigma^2)$.

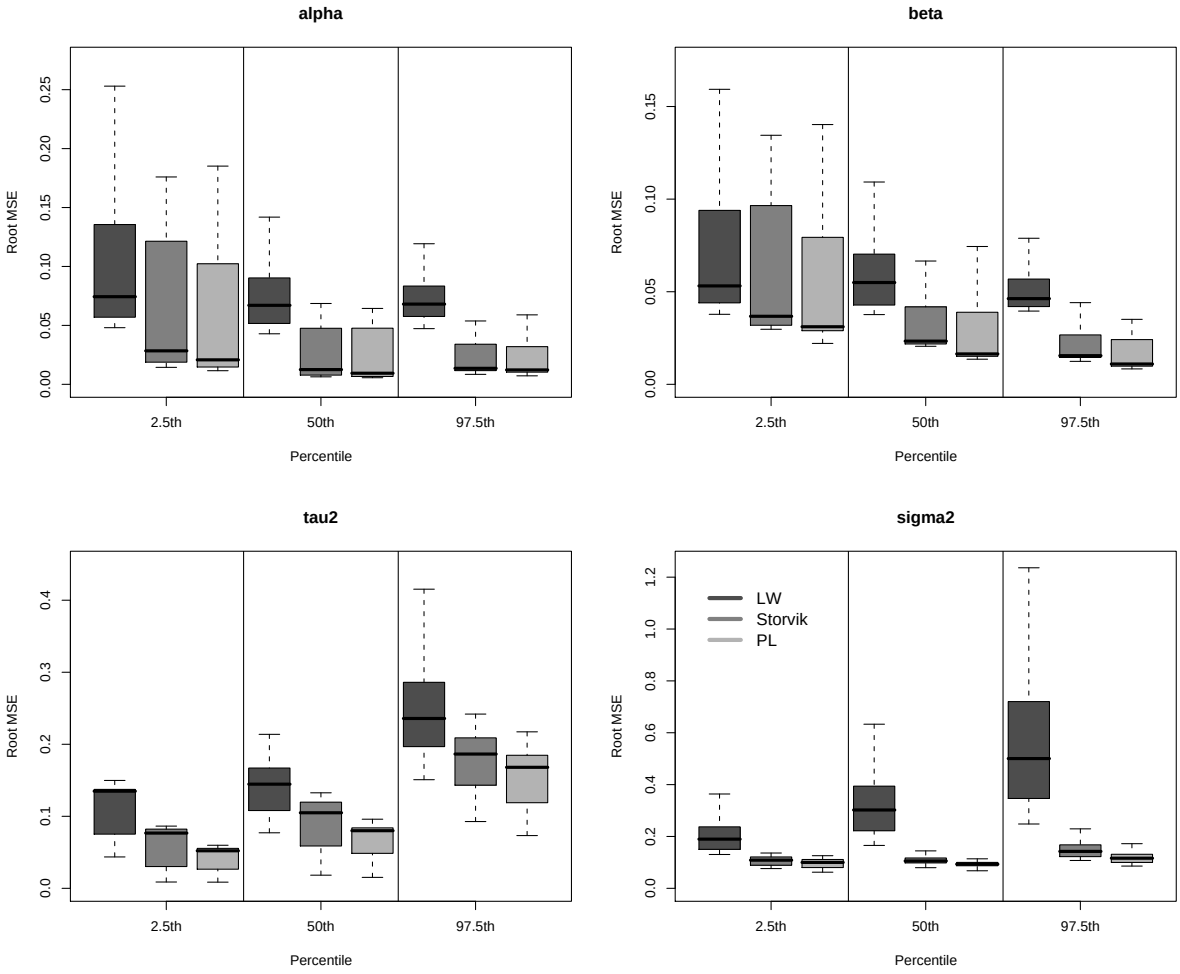

Figure 4: *Comparison between LWF, SF and PL.* Root mean squared error of $R = 100$ replications for each filter. All filters are based on $N = 1000$ particles and the root mean squared error is computed against a long PL run ($N = 100,000$).

Example 4 (cont.) Comparison between PL and MCMC. In the case of pure filtering, i.e. when the parameter vector $\theta = (\alpha, \beta, \tau^2, \sigma^2)$ is known and fixed, it is easy to see that both filtered and smoothed distributions, $p(x_t|y^t, \theta)$ and $p(x_t|y^T, \theta)$, are available in closed form (see Eqs. (5) and (10), respectively). For example, Figure 5 shows that results of particle filtering and a smoothing approximation based on the OAPF ($N = 2000$ particles) virtually match the true distributions. Figure 6 shows that both MCMC approximation ($M = 2000$ draws, after $M_0 = 10000$ as burn-in) and PL approximation ($N = 2000$ particles) to $p(\alpha|y^T)$, $p(\beta|y^T)$, $p(\tau^2|y^T)$ and $p(\sigma^2|y^T)$ are virtually identical. Computational cost is measured here in CPU seconds with $N = 1000$ and $M_0 = M = 1000$ points. FFBS is about one order of magnitude faster than PL for smoothing (34s versus 233s), but PL is approximately three orders of magnitude faster than FFBS for filtering (2s versus 3500s). □

4.5 Sequential model assessment

One of the direct benefits of particle filters is the easy approximation of marginal likelihoods and Bayes factors. These tasks are usually rather involved under MCMC approximations, where computing marginal likelihoods is essentially an independent task in the MCMC paraphernalia. See Kass and Raftery (1995), Carlin and Han (2001), Lopes and West (2004), Gamerman and Lopes (2006, ch. 7) and their references for additional details on several MCMC-based algorithms for Bayesian model assessment.

Even in the simple case of the NDLM of Section 2 and Eq. (13), computing $p(y_1, \ldots, y_T)$ is a nontrivial task. In this case, the simplest Monte Carlo solution is

$$p^N(y^T) = \frac{1}{N}\sum_{i=1}^{N} p(y^T|(\sigma^2, \tau^2)^{(i)}) \approx p(y^T) = \int p(y^T|\sigma^2, \tau^2)p(\sigma^2, \tau^2)d\sigma^2 d\tau^2, \qquad (29)$$

where $\{(\sigma^2, \tau^2)^{(i)}\}_{i=1}^N$ is a random sample from the prior $p(\sigma^2, \tau^2)$. Despite its simplicity, this approximation is very unstable when the prior and the likelihood for (σ^2, τ^2) are moderately separated. Moreover, the MC approximation deteriorates quickly for more general state-space models where the dimension of the parameter space is likely to be greater than two. The sequential Monte Carlo solution to this problem is rather straightforward, with Eq. (13) being approximated by

$$p^N(y^T) = \frac{1}{N^T}\prod_{t=1}^{T}\sum_{i=1}^{N} p(y_t|(x_{t-1}, \sigma^2, \tau^2)^{(i)}), \qquad (30)$$

where $\{(x_{t-1}, \sigma^2, \tau^2)^{(i)}\}_{i=1}^N$ is the particle approximation to $p(x_t, \sigma^2, \tau^2|y^{t-1})$ obtained from the LW filter, Storvik's filter or PL. See the stochastic volatility with Student-t errors from Section 5.2 for the sequential computation of posterior model probabilities.

There are contributions that explicitly deal with parameter, state and model uncertainties all together via SMC methods. Fearnhead (2004), MacEachern, Clyde and Liu (1999 and Carvalho, Lopes, Polson and Taddy (2009) use particle methods for general mixtures.

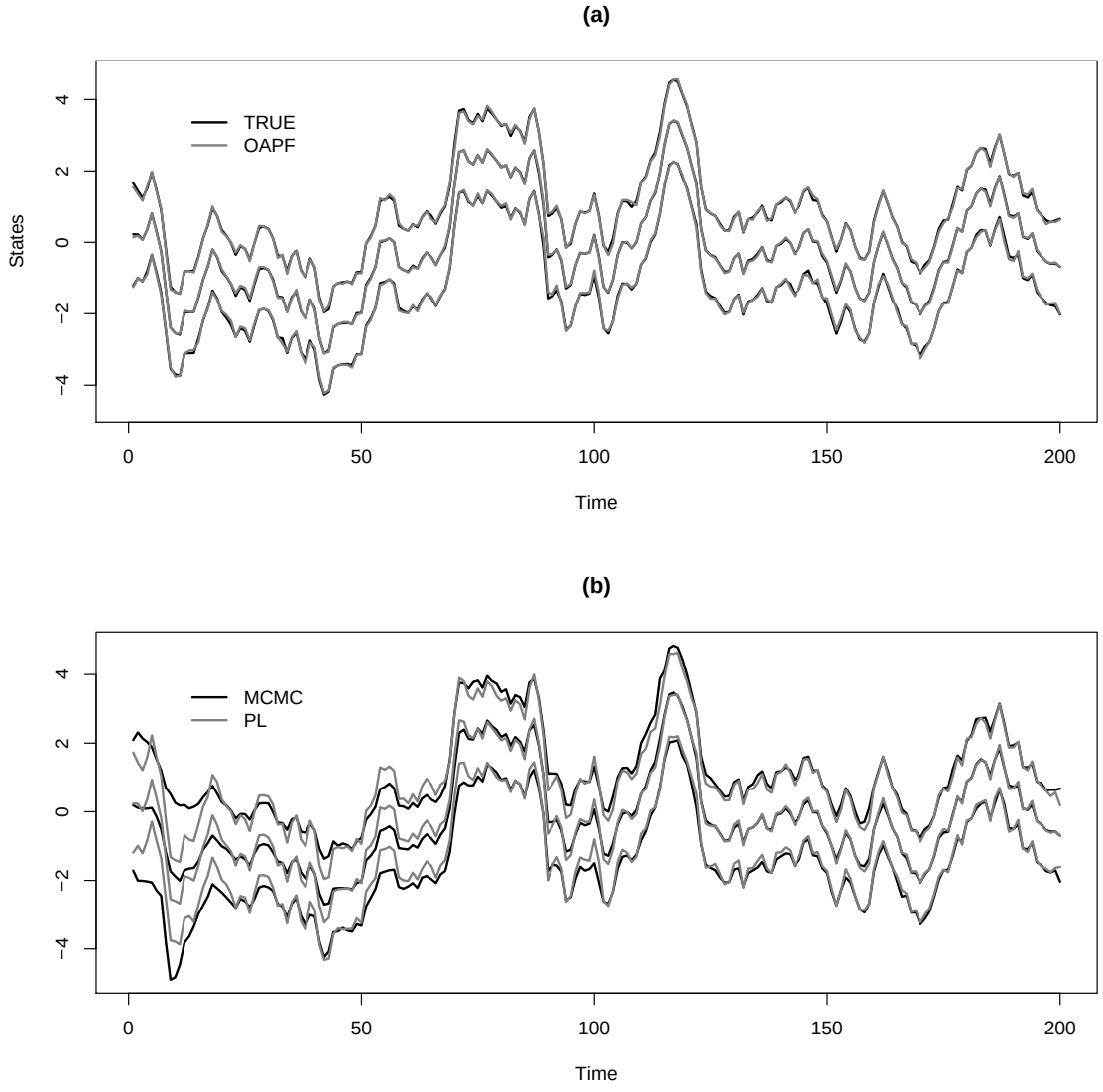

Figure 5: *Particle smoothing.* (a) Comparing the true filtered and smoothed distributions, $p(x_t|y^t)$ and $p(x_t|y^T)$, respectively, with approximations based on $N = 2000$ particles from the OAPF. (b) Comparing the MCMC and PL approximations to the filtered and smoothed distributions, $p(x_t|y^t)$ and $p(x_t|y^T)$. MCMC is based on $M = 2000$ draws, after $M_0 = 10,000$ as burn-in, while PL is based on $N = 2000$ particles.

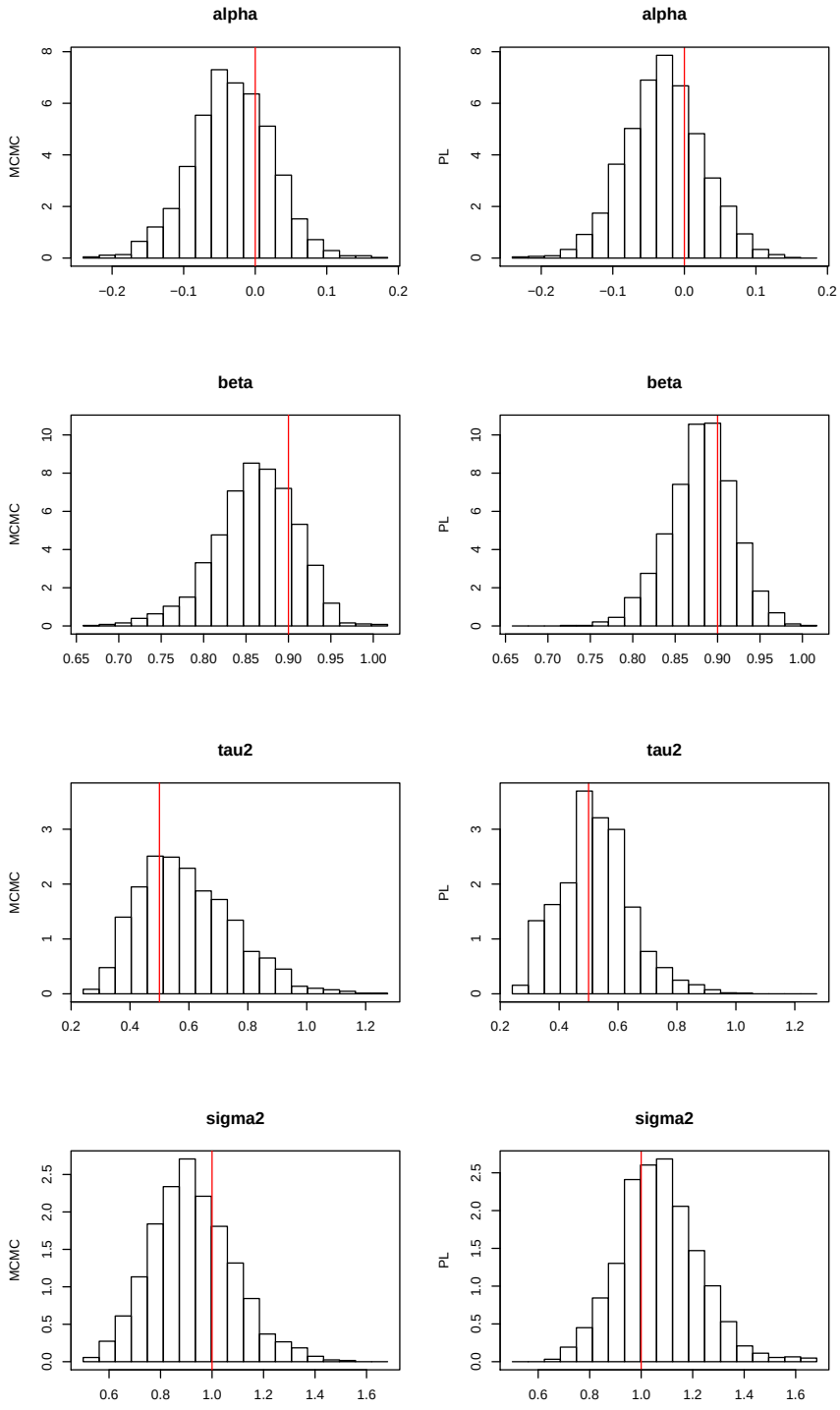

Figure 6: *Parameter learning.* MCMC (left column) and PL (right column) approximations to $p(\theta|y^T)$. Rows are the components of θ, i.e. α, β, τ^2 and σ^2. MCMC is based on $M = 2000$ draws, after $M_0 = 10,000$ as burn-in, while PL is based on $N = 2000$ particles.

5 Applications

In this section we apply particle filters to four time-series problems that are of common interest in many scientific areas. The first application revisits the stochastic volatility model of Example 2. The second application is concerned with the Markov switching stochastic volatility model of Carvalho and Lopes (2007). The last two applications illustrate the use of particle filters in modeling realized volatilities and in estimating unemployment rates via a dynamic generalized linear model.

5.1 Dynamic beta regression

Da Silva, Migon and Correia (2009) use a dynamic beta regression to analyze (via MCMC) the Brazilian monthly unemployment rate from March 2002 to December 2009. More precisely, they model the unemployment rate at time t, namely y_t, by

$$\begin{aligned} y_t|\mu_t,\phi &\sim Beta(\phi\mu_t, \phi(1-\mu_t)) \quad \text{(Beta model)} \\ \mu_t^{-1} &= 1+\exp\{-\beta_t\} \quad \text{(link function)} \\ \beta_t|\beta_{t-1}, W &\sim N(\beta_{t-1}, W), \quad \text{(transition)} \end{aligned}$$

for $t = 1, \ldots, T$, $\beta_0 \sim N(m_0, C_0)$, $\phi \sim IG(a_0, b_0)$ and $W \sim IG(c_0, d_0)$. The dynamic beta regression is a special case of the *dynamic generalized linear model* (DGLM) of West, Harrison and Migon (1985), where the observational distribution belongs to the exponential family. The data was downloaded from The Brazilian Institute for Geography and Statistics (IBGE) site.[2]

We illustrate the computation of sequential Bayes factors via particle filters by comparing the Beta regression model to a simple local level model, i.e. $y_t|\mu_t, \sigma^2 \sim N(\mu_t, \sigma^2)$ and $\mu_t|\mu_{t-1}, \tau^2 \sim N(\mu_{t-1}, \tau^2)$, where $\mu_0 \sim N(m_0, C_0)$, $\sigma^2 \sim IG(a_0, b_0)$ and $\tau^2 \sim IG(c_0, d_0)$ with given hyperparameters. The prior hyperparameters were set at $m_0 = 0.1$, $C_0 = 100$, $a_0 = 2.1$, $b_0 = (a_0+1)0.00001$, $c_0 = 2.1$ and $d_0 = (c_0+1)0.00001$, for the local level model and at $m_0 = \log(y_1/(1-y_1))$, $C_0 = 0.1$, $a_0 = 2.1$, $b_0 = (a_0+1)15000$, $c_0 = 2.1$ and $d_0 = (c_0+1)0.05$, for the dynamic beta model. More general dynamics, such as seasonality, could easily be included in both models with only slight modifications to the models and particle filters. We ignore the seasonality here for simplicity.

Sequential inference for the local level model was performed by PL whereas that for the dynamic beta regression was performed by the LWF. Results appear in Figure 7, while a Monte Carlo study is presented in Figures 8 and 9. The estimations of μ_t under both models are relatively similar, with the sequential Bayes factor slightly favoring the dynamic beta regression model. The Monte Carlo error associated with the estimation of parameters and Bayes factors is relatively small. See Carvalho, Lopes and Polson (2009) for more details about PL for dynamic generalized linear models and dynamic discrete choice models.

[2] `http://www.sidra.ibge.gov.br/bda/pesquisas/pme/default.asp#dead`.

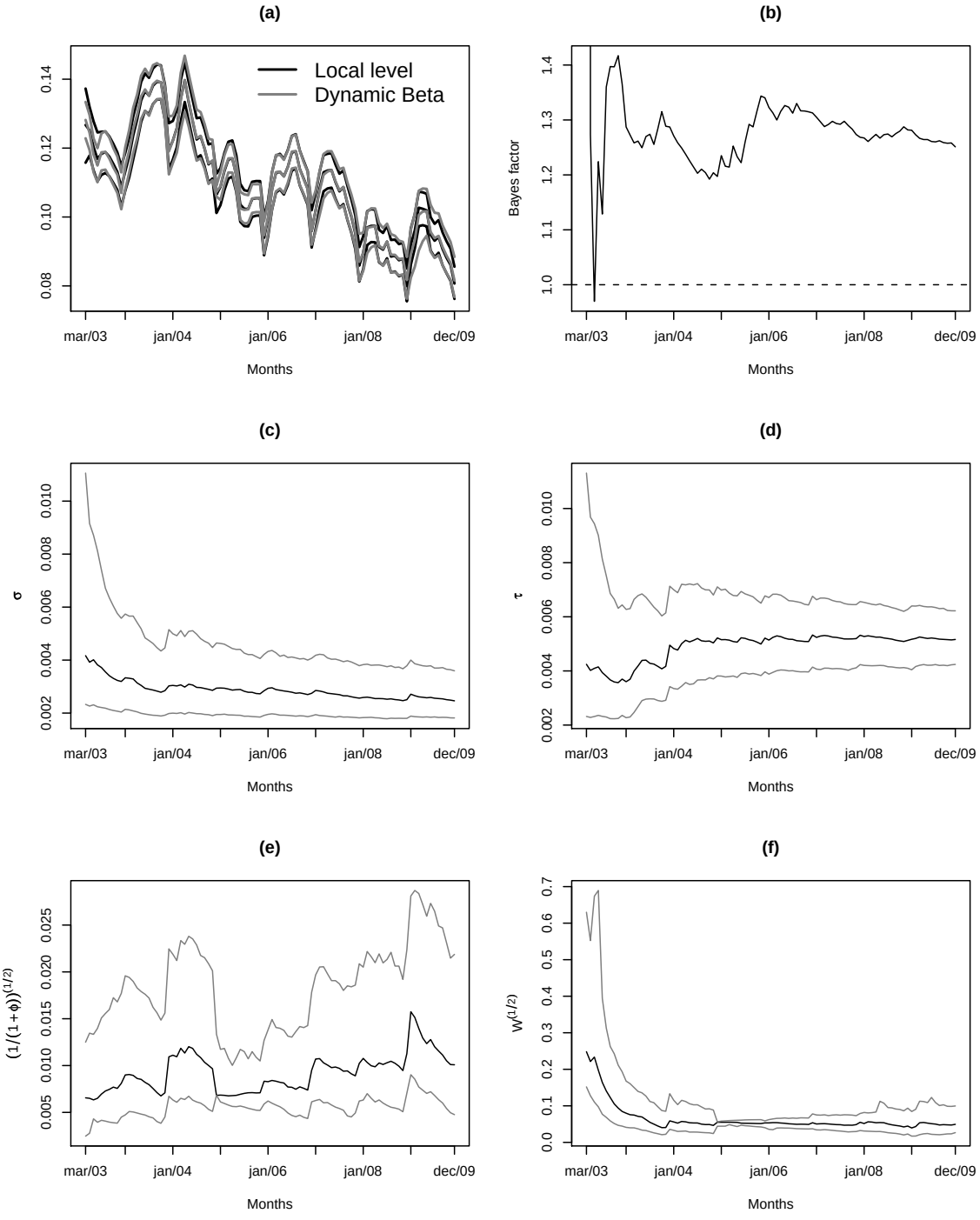

Figure 7: *Dynamic beta regression.* (a) Sequential Monte Carlo (SMC) approximations for the median and 95% credibility interval based on $N = 10,000$ particles for both models. (b) Sequential Bayes factor. (c) and (d) Sequential parameter learning for σ and τ from the local level model. (e) and (f) Sequential parameter learning for $(1 + \phi)^{-1/2}$ and $W^{1/2}$ from the dynamic beta model.

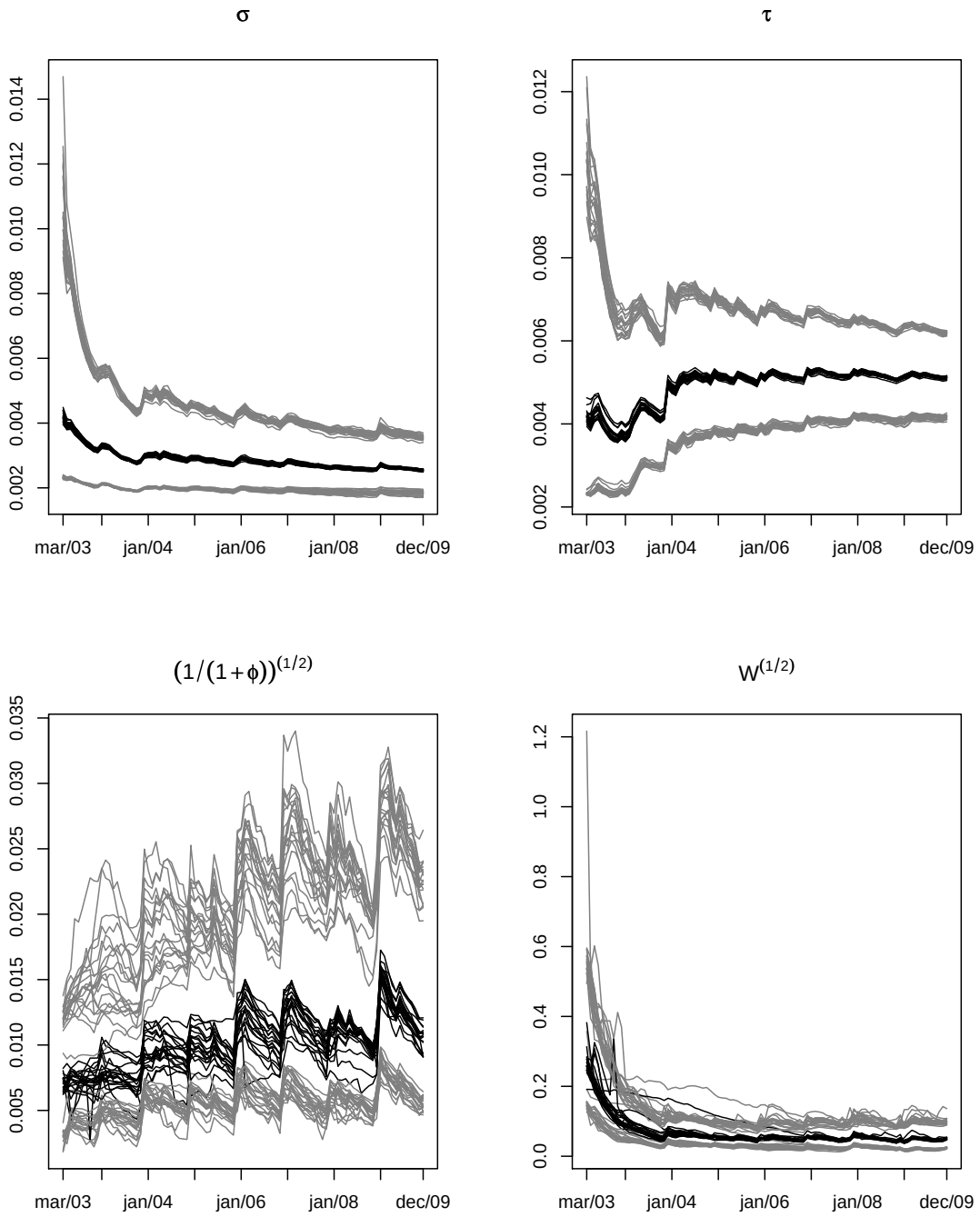

Figure 8: *Dynamic beta regression.* A total of 20 replications of the SMC algorithm, each one based on $N = 10,000$ particles. *Top row:* σ and τ from the local level model. *Bottom row:* $(1+\phi)^{-1/2}$ and $W^{1/2}$ from the dynamic beta model.

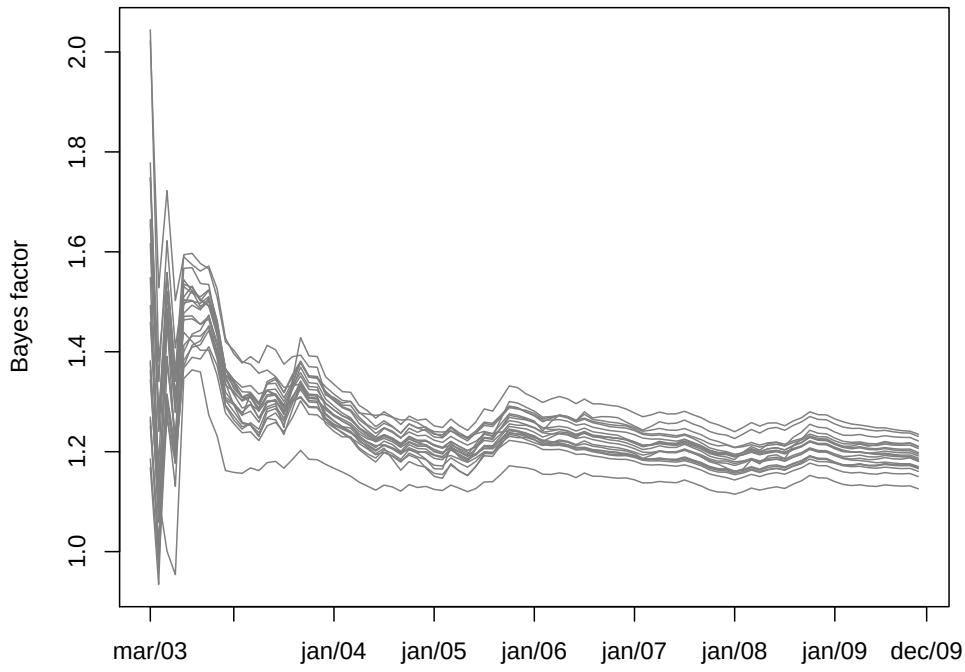

Figure 9: *Dynamic beta regression.* A total of 20 replications of the SMC algorithm, each one based on $N = 10,000$ particles.

5.2 Stochastic volatility model with Student-t innovations

We revisit the simple SV model with normal innovations of Example 2 and compute sequential Bayes factors against the alternative SV model with Student-t innovations. We use monthly log returns of GE stock from January 1926 to December 1999 for 888 observations. This series was analyzed in Example 12.6 of Tsay (2005, ch. 12). [3] The competing models are:

$$\begin{aligned} \text{Observation equation} &: y_t|(x_t,\theta) \quad \sim t_\eta(0,\exp\{x_t\}), \\ \text{System equation} &: (x_t|x_{t-1},\theta) \quad \sim N(\alpha+\beta x_{t-1},\tau^2), \end{aligned}$$

where $t_\eta(\mu,\sigma^2)$ denotes the Student-t distribution with η degrees of freedom, location μ and scale σ^2. The number of degrees of freedom η is treated as known. Sequential posterior inference is based on the Liu and West filter with $N = 100,000$ particles. The shrinkage constant a is set at $a = 0.95$, whereas prior hyperparameters are $m_0 = 0$, $C_0 = 10$, $\nu_0 = 3$, $\tau_0^2 = 0.01$, $b_0 = (0,1)'$ and $B_0 = 10I_2$. Particle approximation to the sequential posterior model probabilities, assuming a uniform prior for η over models $\{t_\infty, t_2, \ldots, t_{20}\}$, appears in Figure 10, where t_∞ denotes the normal distribution. Figure 10(d) shows percentiles of $p(\sigma_t|y^t)$ when integrating out over all competing models in $\{t_\infty, t_2, \ldots, t_{20}\}$. One can argue that the data slowly move over time from a more t-like, heavy tail model towards a more Gaussian, thin tail model. Figures 11 and 12 present posterior summaries for the volatilities and parameters of a few competing models.

5.3 Markov switching stochastic volatility model

Jumps have been intensively studied in financial data analysis; see, for example, Eraker, Johannes and Polson (2003). So *et al.* (1998) suggest a model that allows for occasional discrete shifts in the parameter determining the level of the log-volatility through a Markovian process. They claim that this model not only is a better way to explain volatility persistence but is also a tool to capture changes in economic forces, as well as abrupt changes due to unusual market forces. Carvalho and Lopes (2007) adopt the LWF when sequentially estimating univariate financial time series with a MSSV structure. Let us call this filter the CL filter. For illustration, we consider a MSSV model with two regimes, i.e.

$$\begin{aligned} y_t|x_t,\theta &\sim N(0,\exp\{x_t\}), \\ \lambda_t|x_{t-1},s_t,\theta &\sim N(\alpha_{s_t}+\beta x_{t-1},\tau^2), \end{aligned}$$

where $Pr(s_t = j|s_{t-1} = i) = p_{ij}$, for $i,j = 1,2$, and parameters $\theta = (\alpha_1,\alpha_2,\beta,\tau^2,p_{11},p_{22})$. Rios and Lopes (2009) propose an extension of the CL filter, which they named the extended LW (ELW) filter, that combines features of the LW filter and PL. The simulation exercise from figure 13 shows that the CL filter degenerates after 500 observations whereas the ELW filter never collapses.

[3] The data are available at `http://faculty.chicagobooth.edu/ruey.tsay/teaching/fts2/m-geln.txt`

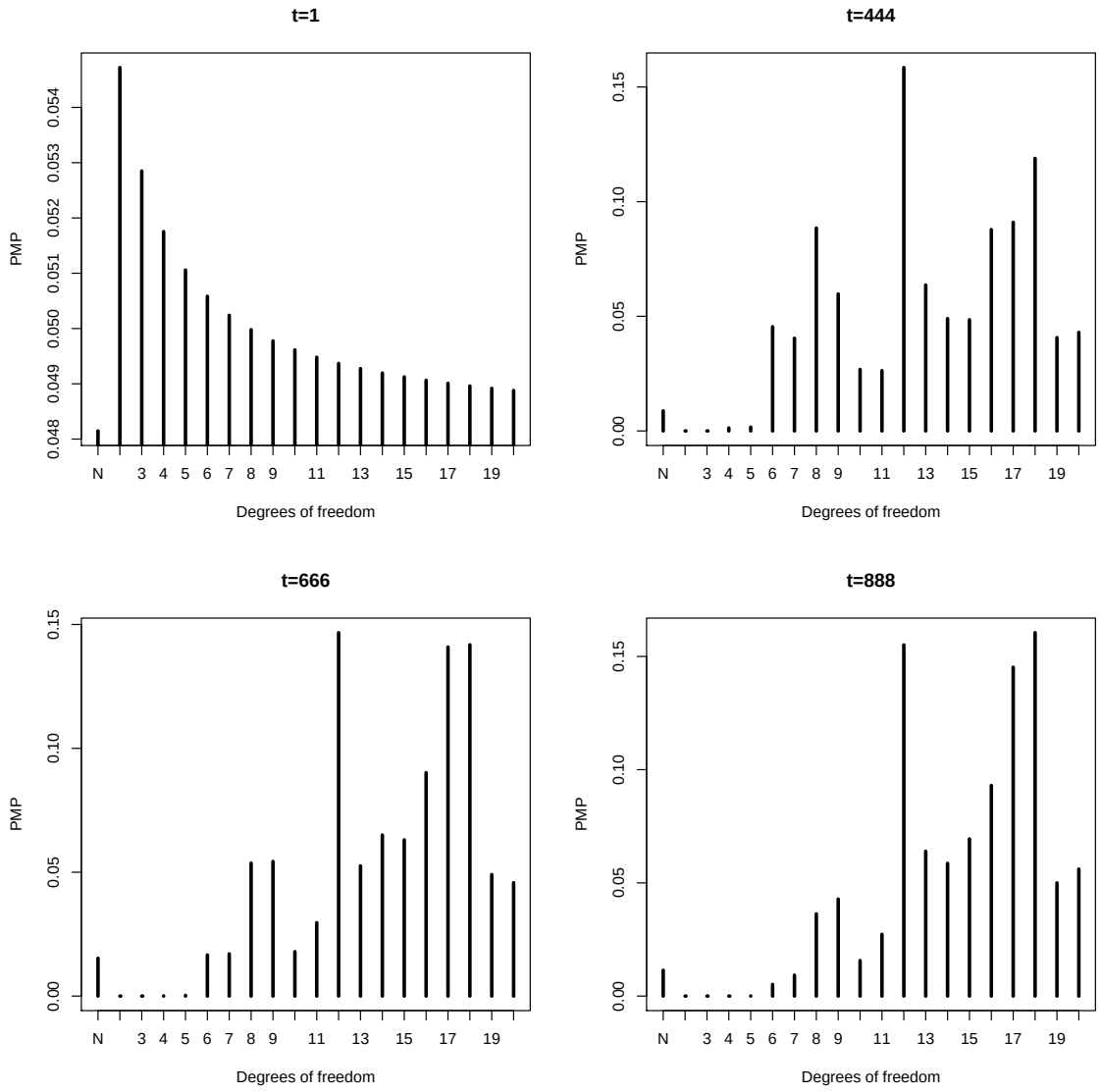

Figure 10: *Stochastic volatility model.* Sequential posterior model probability for the number of degrees of freedom η.

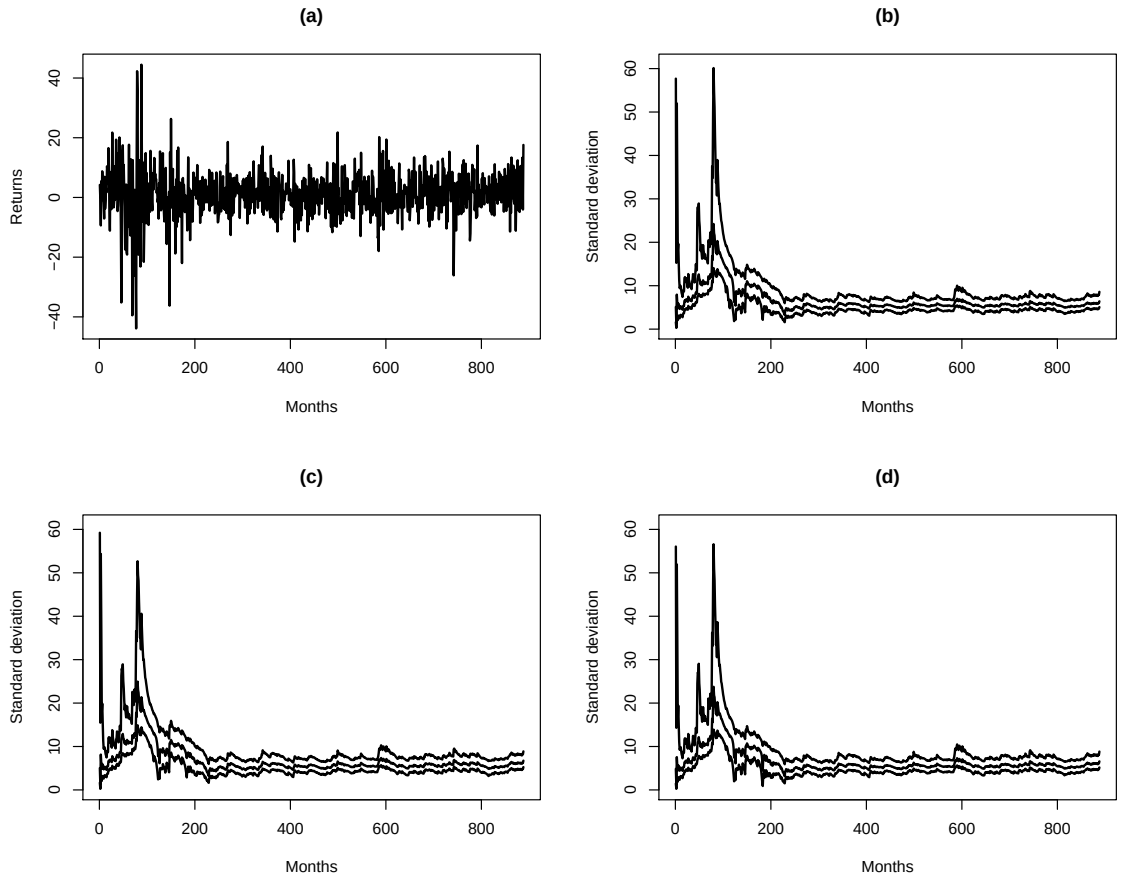

Figure 11: *Stochastic volatility model.* (a) GE returns; (b) and (c) 2.5th, 50th and 97.5th percentiles of $p(\sigma_t|y^t, M)$, where $\sigma_t^2 = \exp\{x_t\}$, for $M = t_{12}$ and $M = t_{18}$, respectively. (d) 2.5th, 50th and 97.5th percentiles of $p(\sigma_t|y^t)$ by integrating out over all competing models in $\{\text{Normal}, t_2, \ldots, t_{20}\}$.

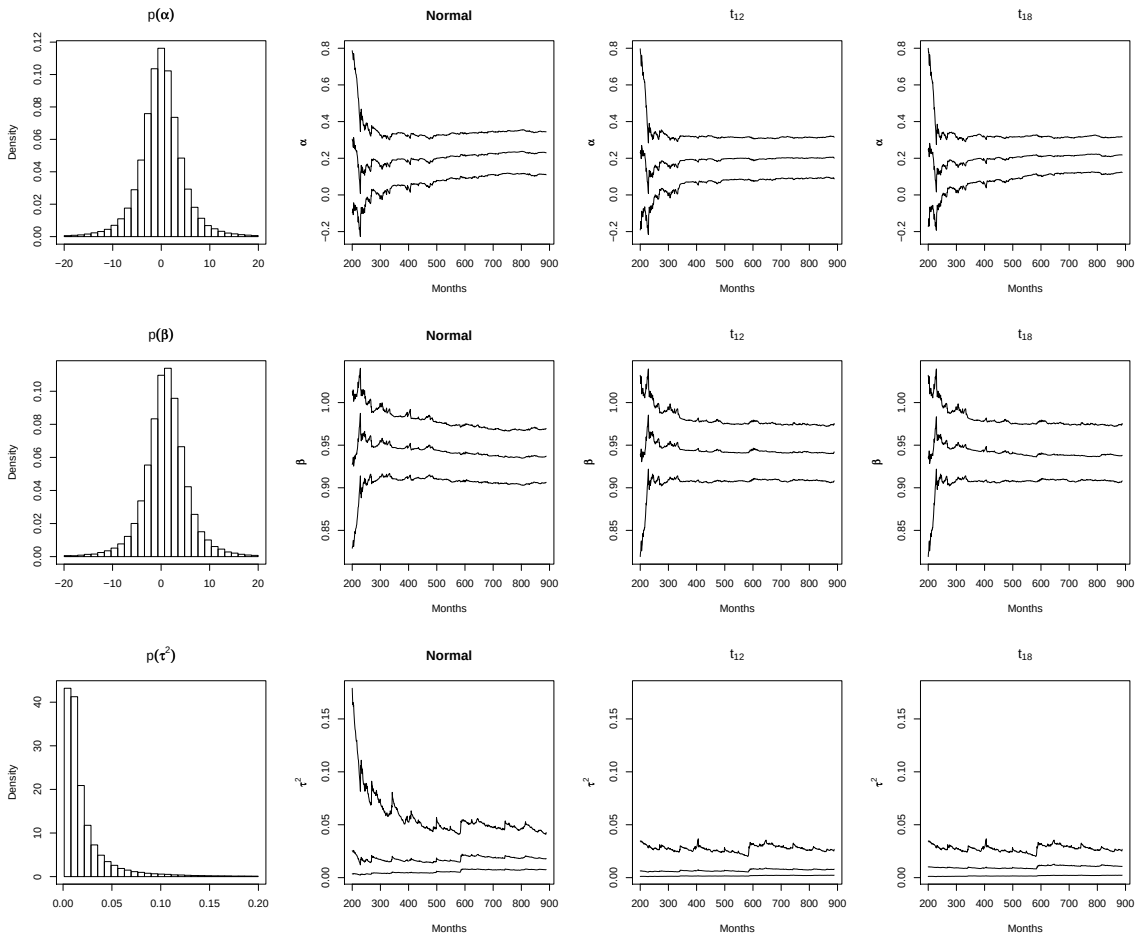

Figure 12: *Stochastic volatility model.* Column 1: Marginal prior distributions for α, β and τ^2. Columns 2 to 4: Sequential 2.5th, 50th and 97.5th percentiles of $p(\gamma|y^t, M_1)$, for γ in $(\alpha, \beta, \tau^2, M)$ and model $M \in \{\text{Normal}, t_{12}, t_{18}\}$.

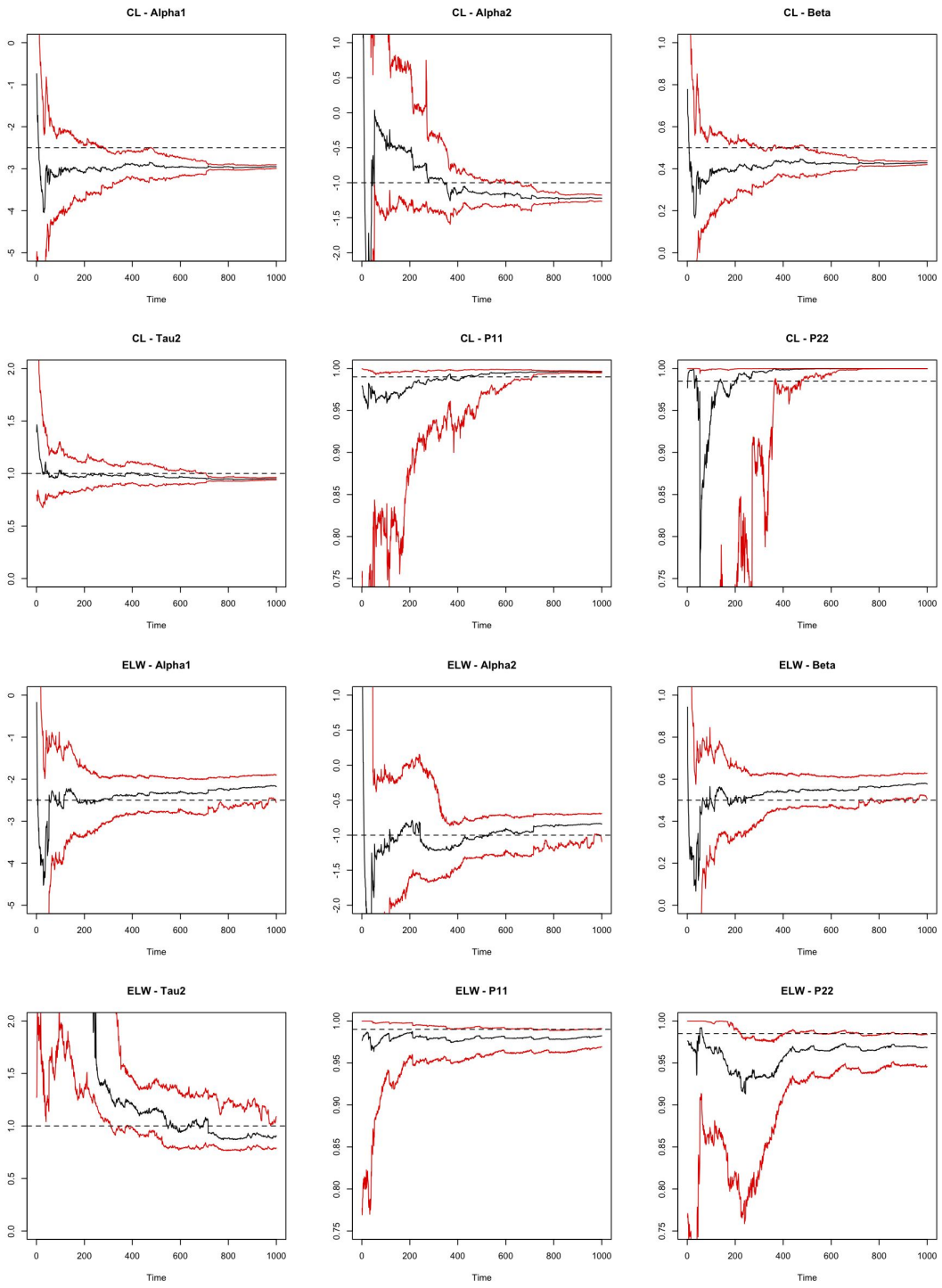

Figure 13: *Markov switching stochastic volatility.* Carvalho and Lopes' (2007) filter (last two rows) and Rios and Lopes' (2009) ELW filter (first two rows). 2.5th, 50th and 97.5th percentiles of the marginal distribution of each parameter based on $N = 5000$ particles. The dotted lines are the true values are $\alpha_1 = -2.5$, $\alpha_2 = -1.0$, $\beta = 0.5$, $\tau^2 = 1.0$, $p_{11} = 0.99$ and $p_{22} = 0.985$.

5.4 Realized volatility

We consider the intradaily realized volatilities of Alcoa stock from January 2, 2003 to May 7, 2004 for 340 observations. The daily realized volatilities used are the sums of squares of intraday 5-minute, 10-minute and 20-minute log returns measured in percentages; see Tsay (2005, Ch. 11). In what follows, we use the logarithms of the daily realized volatilities. Figure 14 presents the time series of log realized volatilities. As expected, all three series behave similarly over time and are highly positively correlated, with the 5 & 10 and 10 & 20 minute realized volatilities more correlated than the 5 & 20 minute ones. Table 1 shows summary statistics of the time series.

Table 1: *Summary statistics.* Correlations (*below main diagonal*) and covariances (*main diagonal and above*).

					Correlations/covariances		
RV	Mean	Median	Skewness	Kurtosis	5-minute	10-minute	20-minute
5-minute	0.992	0.977	1.091	5.479	*0.314*	*0.270*	*0.258*
10-minute	0.913	0.871	0.153	0.769	0.857	*0.317*	*0.307*
20-minute	0.850	0.847	0.034	0.843	0.732	0.865	*0.396*

Two competing models. We entertain two models: i) the three RV time-series are modeled by independent univariate local level models; and ii) the trivariate vector of RV time-series is modeled by a multivariate local level model. In the first model, say model $\mathcal{M}_1$, the i-minute log realized volatility y_{it}, for $i = 5, 10, 20$, is initially modeled by a local level model: $(y_{it}|x_{it}, \sigma_i^2) \sim N(x_{it}, \sigma_i^2)$ and $(x_{it}|x_{i,t-1}, \tau_i^2) \sim N(x_{i,t-1}, \tau_i^2)$. In the second model, say model $\mathcal{M}_2$, the univariate local level model is extended to jointly model the $p = 3$ time series of realized volatilities: $(y_t|x_t, \Sigma) \sim N(1_p x_t, \Sigma)$ and $(x_t|x_{t-1}, \tau^2) \sim N(x_{t-1}, \tau^2)$ and 1_p is a p-dimensional unity vector. The diagonal elements of the covariance matrix Σ are σ_i^2, for $i = 1, \ldots, p$, and the non-diagonal elements are σ_{ij}, for $i < j = 1, \ldots, p$.

Parameter learning. The variances σ_i^2 and τ_i^2 under $\mathcal{M}_1$ are, a priori, independent with $\sigma_i^2 \sim IG(a_0, b_0)$, $\tau_i^2 \sim IG(c_0, d_0)$ and hyperparameters $a_0 = c_0 = 10$, $b_0 = 1.1$ and $d_0 = 0.55$ common across $i = 5, 10, 20$. In this case the prior mean and mode of σ_i^2 are 0.122 and 0.1, respectively, while its prior 95% credibility interval is $(0.064, 0.229)$. Similarly, the prior mean and mode of τ_i^2 are 0.061 and 0.05, respectively, while its 95% credibility interval is $(0.032, 0.115)$. Under model $\mathcal{M}_2$, $\tau^2 \sim IG(c_0, d_0)$ and $\Sigma \sim IW(\nu_0, S_0)$[4]. When $p = 1$, $\sigma^2 \sim IG(\nu_0/2, S_0/2)$, so we set $\nu_0 = 2a_0 = 20$ and $S_0 = 2b_0 I_p = 2.2 I_p$ for comparison to the

[4] Σ is inverse-Wishart with parameters ν_0 and S_0 and density $p(\Sigma; \nu_0, S_0) \propto |\Sigma|^{-\frac{\nu_0+p+1}{2}} \exp\{-0.5\text{tr}(S_0\Sigma^{-1})\}$, for $\nu_0 > p-1$, $S_0 > 0$ (positive definite) and $\text{tr}(\Sigma) = \sigma_1^2 + \cdots + \sigma_p^2$. The mean and the mode of Σ are $S_0/(\nu_0 - p - 1)$ and $S_0/(\nu_0 + p + 1)$, respectively. Its inverse Σ^{-1} is Wishart with the same parameters and denoted by $\Sigma^{-1} \sim W(\nu_0, S_0)$. The mean and the mode of Σ^{-1} are $\nu_0 S_0^{-1}$ and $(\nu_0 - p - 1)S_0^{-1}$ $(\nu_0 \geq p + 1)$, respectively.

univariate models. The prior mean and mode of Σ are $0.138I_p$ and $0.092I_p$, respectively. The parameter $\theta = (\tau^2, \Sigma)$ can be sampled from $p(\theta|s_t) = p_{IG}(\tau^2; c_t, d_t)p_{IW}(\Sigma; \nu_t, S_t)$, where the recursive sufficient statistics are $c_t = c_{t-1} + 1/2$, $d_t = d_{t-1} + (x_t - x_{t-1})^2/2$, $\nu_t = \nu_{t-1} + 1$ and $S_t = S_{t-1} + (y_t - 1_p x_t)(y_t - 1_p x_t)'$.

State learning. Let us start by assuming that $(x_{t-1}|y^{t-1}, \theta) \sim N(m_{t-1}, C_{t-1})$, with $x_0 \sim N(m_0, C_0)$, $\theta = (\Sigma, \tau^2)$ and $s^x_{t-1} = (m_{t-1}, C_{t-1})$. PL starts by resampling the particles $\{(\theta, s_{t-1}, s^x_{t-1})^{(i)}\}_{i=1}^N$ with weights $p(y_t|s^x_{t-1}, \theta) = p_N(y_t; 1_p m_{t-1}, Q_t)$, where $Q_t = R_t D_p + \Sigma$, $D_p = 1_p 1_p'$ and $R_t = C_{t-1} + \tau^2$. The state sufficient statistics s^x_{t-1} are then propagated to $s_t = (m_t, C_t)$, where $m_t = (1 - A_t 1_p)m_{t-1} + A_t y_t$, $C_t = R_t - A_t Q_t A_t'$ and $A_t = R_t 1_p' Q_t^{-1}$. Since both x_{t-1} and x_t are used in d_t when sampling τ^2 from $IG(c_t, d_t)$, then (x_{t-1}, x_t) need to be sampled from $p(x_{t-1}, x_t|y^t, \theta) = p_N(x_{t-1}; g(y_t), V_t)p_N(x_t; h(y_t, x_{t-1}), \bar{C}_t)$, where $V_t^{-1} = C_{t-1}^{-1} + 1_p' W^{-1} 1_p$, $g(y_t) = V_t(C_{t-1}^{-1} m_{t-1} + 1_p' W^{-1} y_t)$, $\bar{C}_t^{-1} = \tau^{-2} + 1_p' \Sigma^{-1} 1_p$ and $h(y_t, x_{t-1}) = \bar{C}_t(\tau^{-2} x_{t-1} + 1_p' \Sigma^{-1} y_t)$, for $W = \tau^2 D_p + \Sigma$. Finally, marginal posterior inference for x_t given (y^t, θ) is more efficient if drawn from $p(x_t|y_t, s^x_{t-1}, \theta) = p_N(x_t; \tilde{m}_t, \tilde{C}_t)$, where $\tilde{C}_t^{-1} = R_t^{-1} + 1_p' \Sigma^{-1} 1_p$ and $\tilde{m}_t = \tilde{C}_t(R_t^{-1} m_{t-1} + 1_p' \Sigma^{-1} y_t)$.

Results. Figures 15 to 17 summarize the sequential learning of parameters and states for the univariate local level model based on $N = 10,000$ particles, which is fairly large considering the sample size $T = 340$. Figures 18 to 21 summarize the results for the multivariate local model, also based on $N = 10,000$ particles. Figure 22 compares the sequential posterior medians for the latent states from the three individual fits of model $\mathcal{M}_1$ against the multivariate fit of model $\mathcal{M}_2$. Notice the shrinkage effect of model $\mathcal{M}_2$, which provides a smoother point estimate for the latent state.

6 Concluding Remark

In this paper we review particle filters, which are also known as sequential Monte Carlo (SMC) methods. We argue that, after almost two decades since the seminal paper of Gordon, Salmond and Smith (1993), SMC methods now belong in the toolbox of researchers and practitioners in many areas of modern science, ranging from signal processing and target tracking to robotics, bioinformatics and financial econometrics. This paper focuses on the latter with five demonstrations.

Besides the references of PF in financial econometrics cited in Section 5, some additional ones are Johannes, Korteweg and Polson (2008) on predictive regressions and optimal portfolio allocation, Raggi and Bordignon (2006), Jasra *et al.* (2008), Li *et al.* (2008), Creal (2008) and Li (2009) on Lévy-type SV models, and Johannes, Polson and Stroud (2009) on extracting latent jump diffusions from asset prices. See also Fearnhead and Meligkotsidou (2004) and Fearnhead *et al.* (2008) for particle filters in partially-observed continuous-time models and diffusions. PF for jump Markov systems are studied by Doucet, Gordon and Krishnamurthy (2001) and Andrieu, Davy and Doucet (2003).

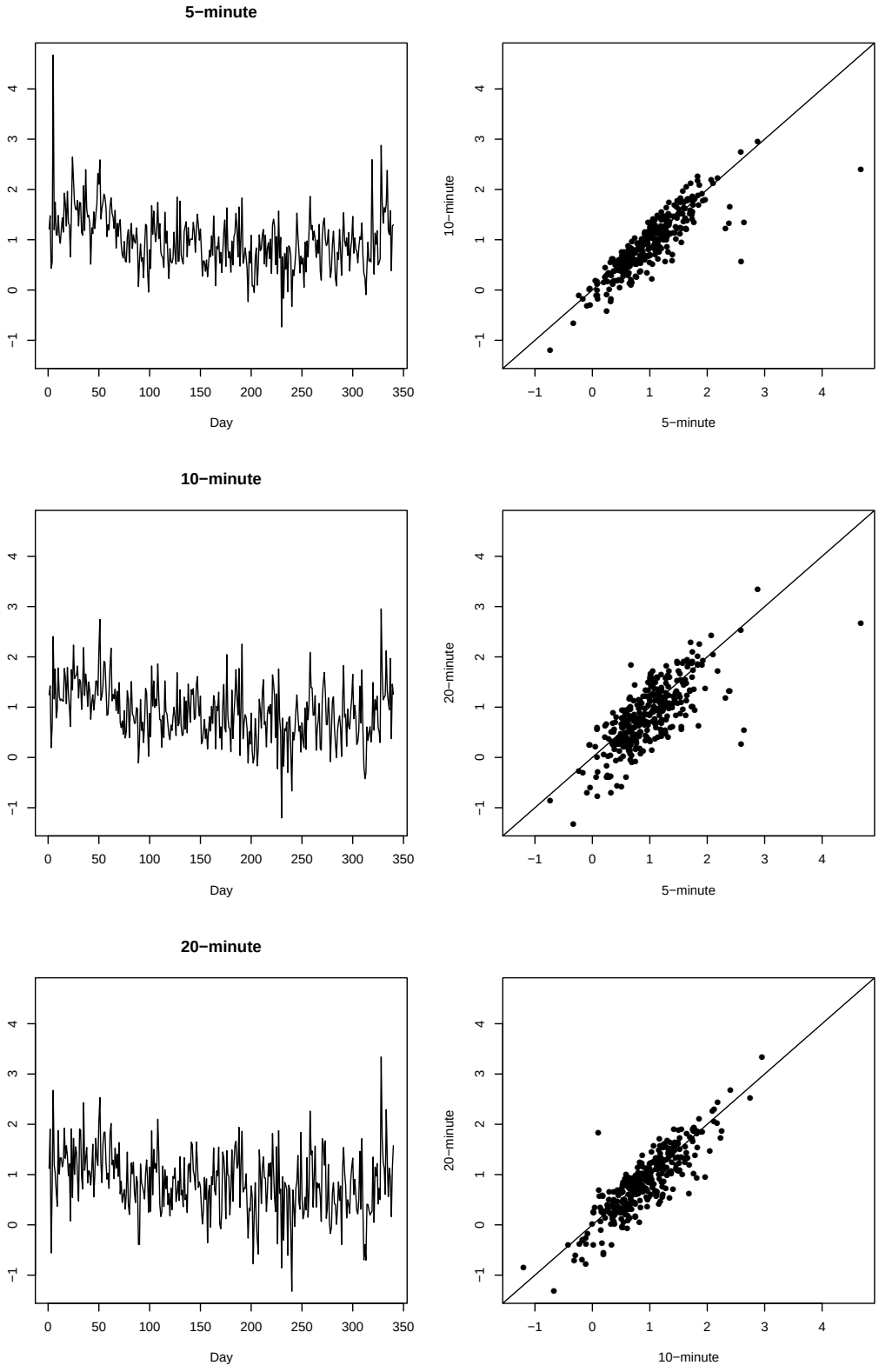

Figure 14: *Realized volatility.* Log realized volatility of Alcoa stock based on the sum of squares of intraday 5-minute, 10-minute and 20-minute log returns measured in percentage.

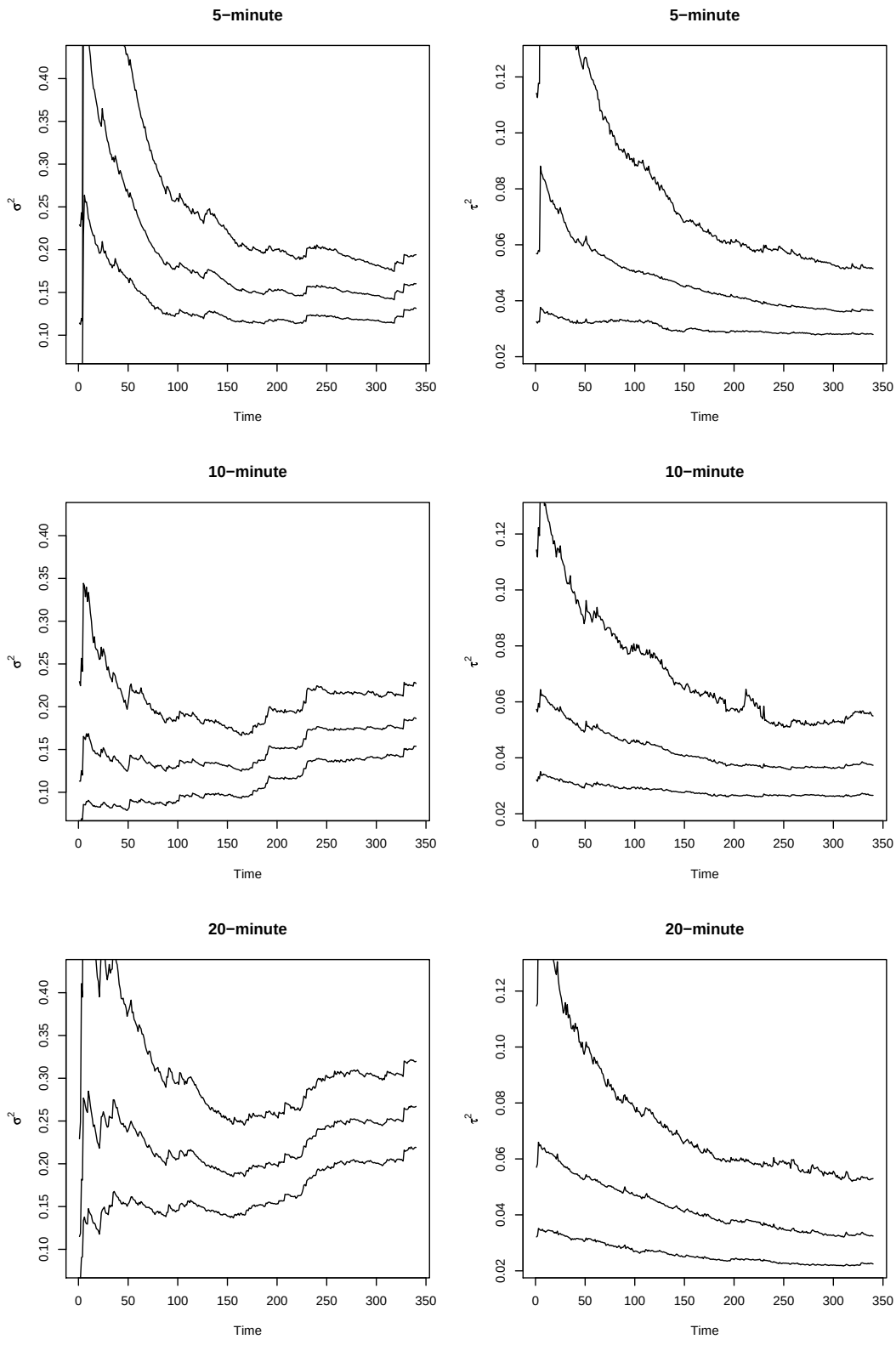

Figure 15: *Realized volatility.* 2.5th, 50th and 97.5th percentiles of $p(\sigma_i^2|y_i^t)$, $p(\tau_i^2|y_i^t)$ and $p(x_{it}|y_i^t)$, where y_i is i-minute log realized volatilities, for $i = 5, 10, 20$.

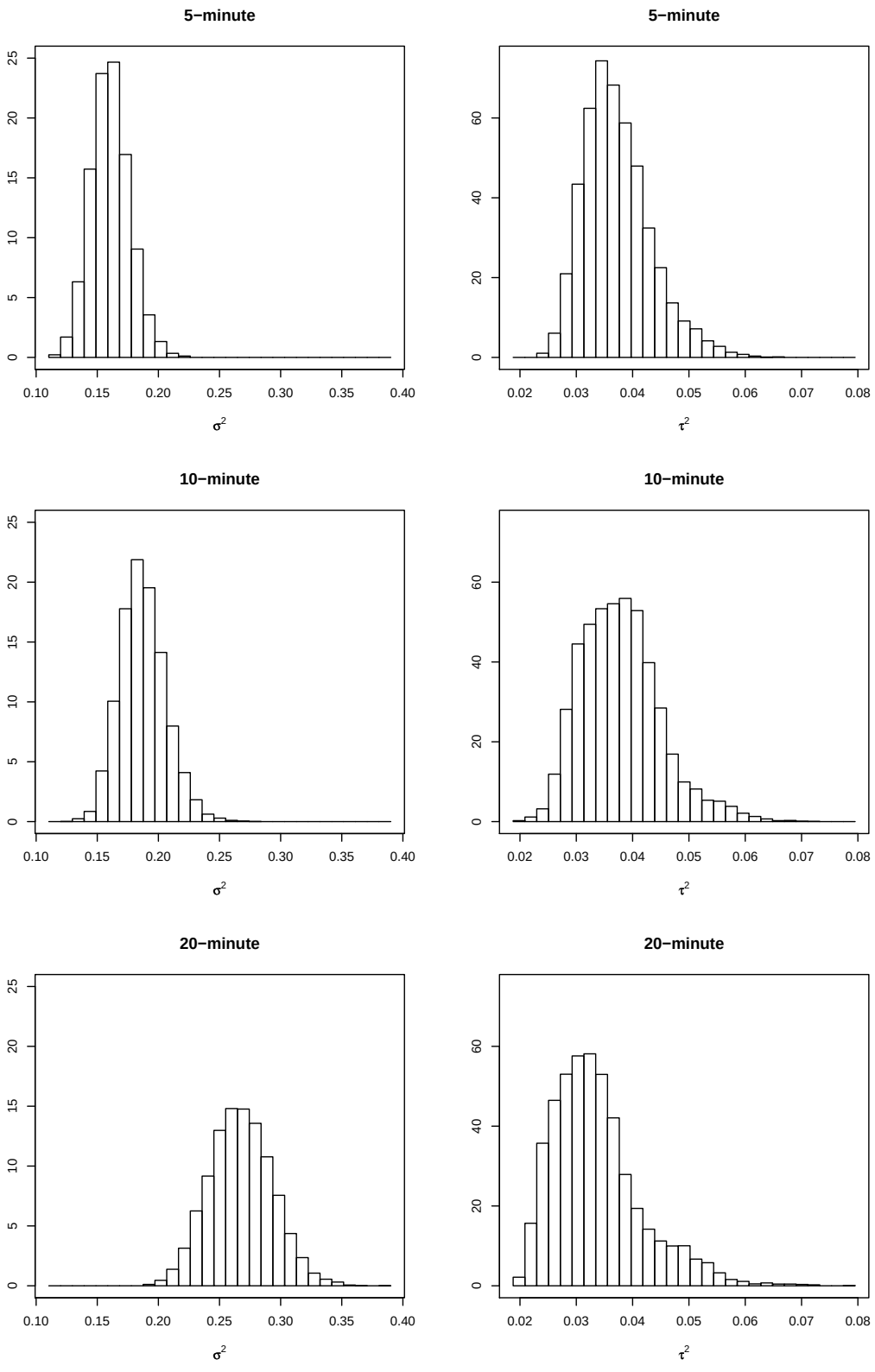

Figure 16: *Realized volatility.* Histogram approximations to $p(\sigma_i^2|y^T)$ and $p(\tau_i^2|y^T)$, for $i = 5, 10, 20$.

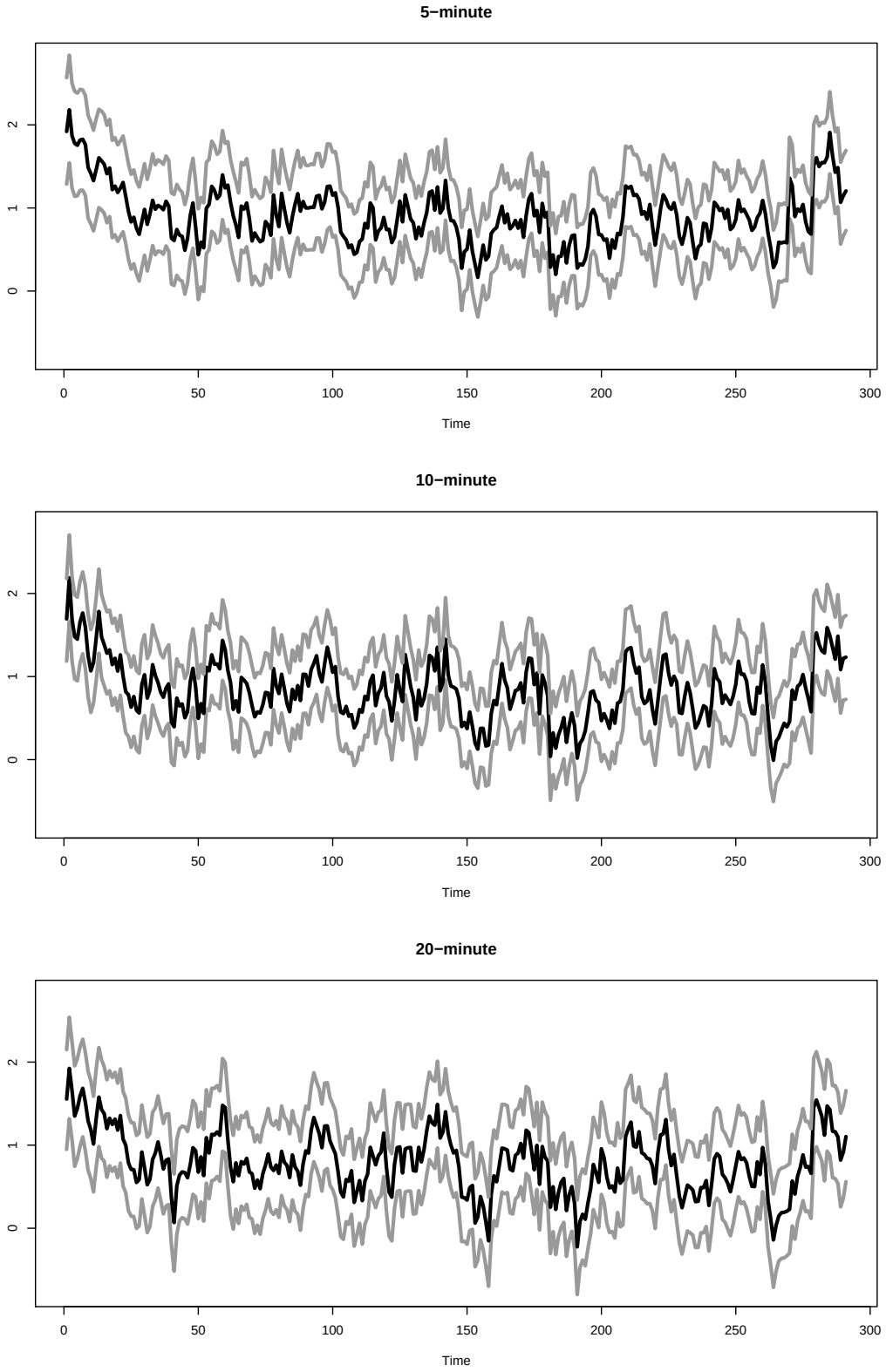

Figure 17: *Realized volatility.* 2.5th, 50th and 97.5th percentiles of $p(x_{it}|y_i^t)$, where y_i is i-minute log realized volatilities, for $i = 5, 10, 20$.

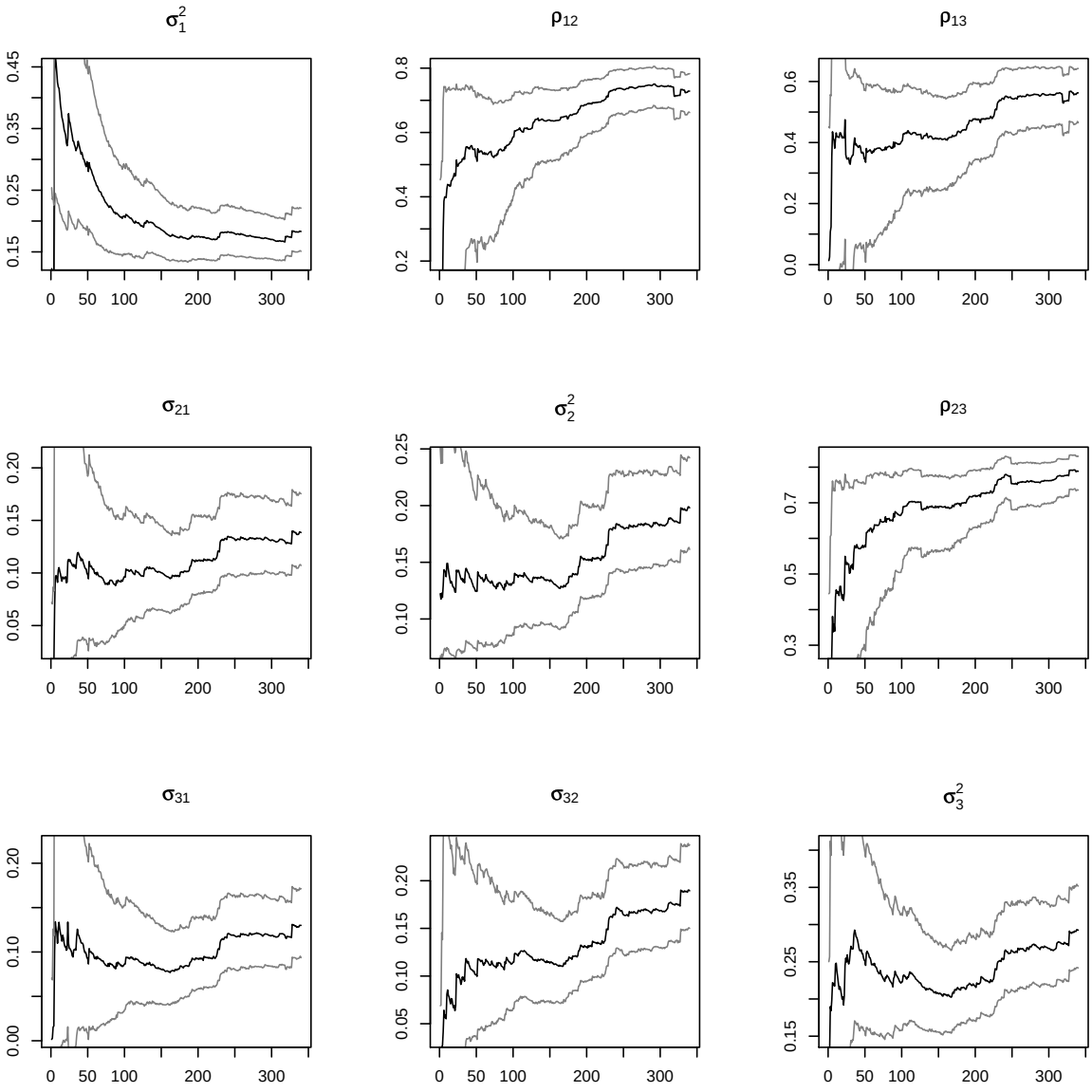

Figure 18: *Realized volatility.* Sequential 2.5th, 50th and 97.5th percentiles for the unique components of Σ and for the correlations.

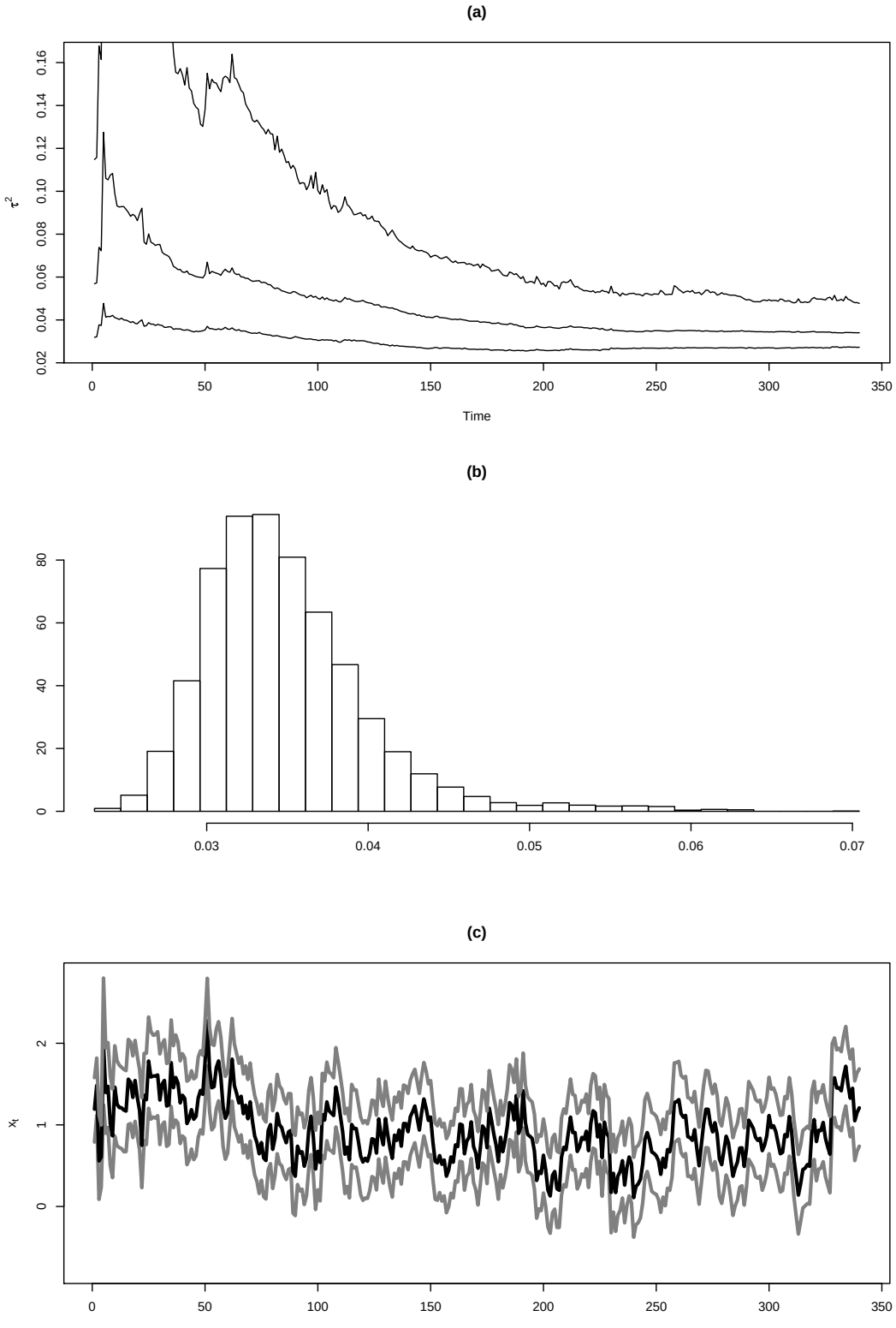

Figure 19: *Realized volatility.* (a) Sequential 2.5th, 50th and 97.5th percentiles of $p(\tau^2|y^t)$. (b) Histogram approximation to $p(\tau^2|y^T)$. (c) Sequential 2.5th, 50th and 97.5th percentiles of $p(x_t|y^t)$.

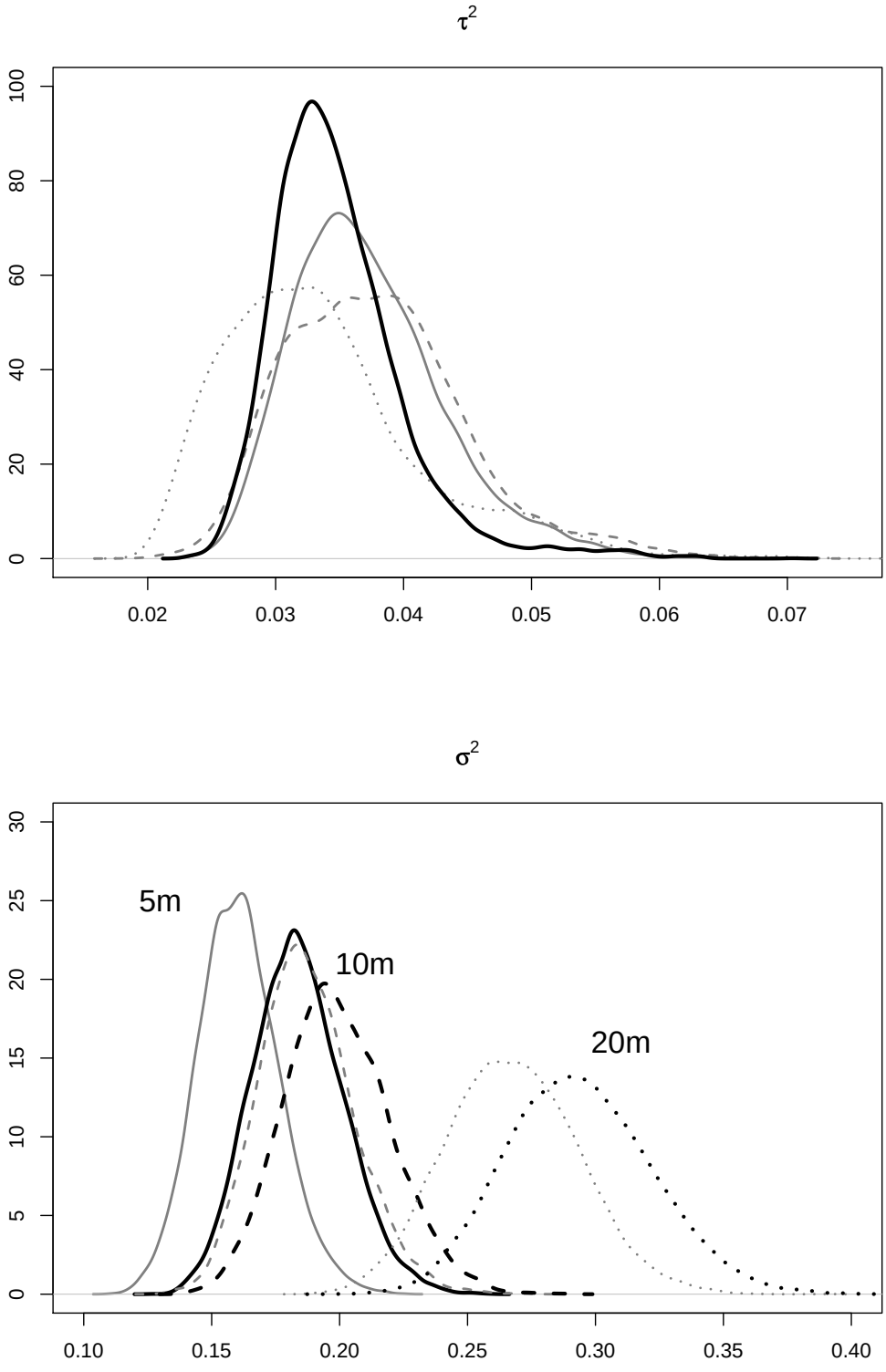

Figure 20: *Realized volatility.* Approximated posterior distributions of τ_i^2 and σ_i^2. Univariate local level models (grey lines) and multivariate local level model (black lines), for $i = 5, 10, 20$ (solid, dashed and dotted lines).

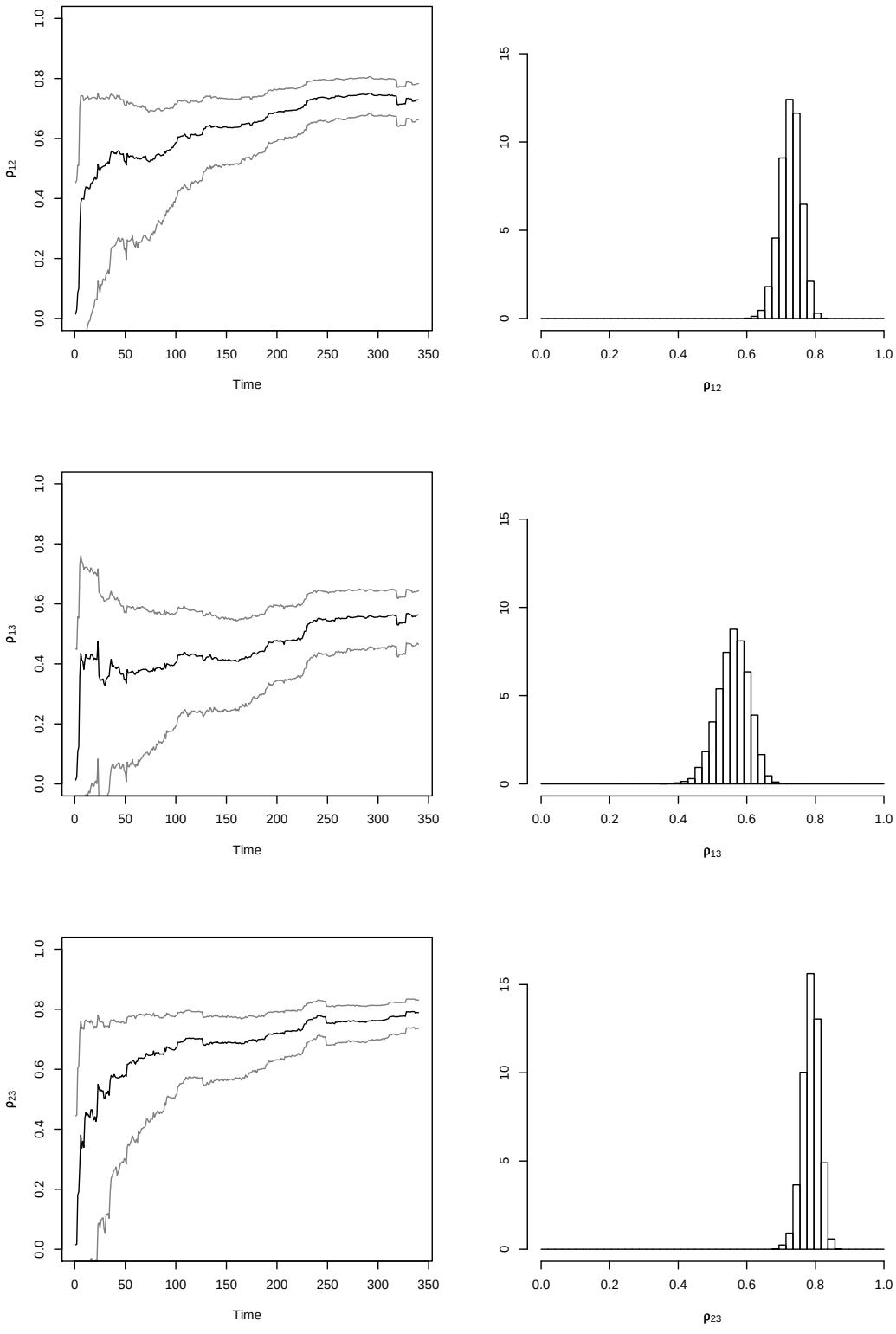

Figure 21: *Realized volatility.* Sequential learning of correlations. Sequential 2.5th, 50th and 97.5th percentiles of $p(\rho_{ij}|y^t)$ (left column). Histogram approximation to $p(\rho_{ij}|y^T)$ (right column).

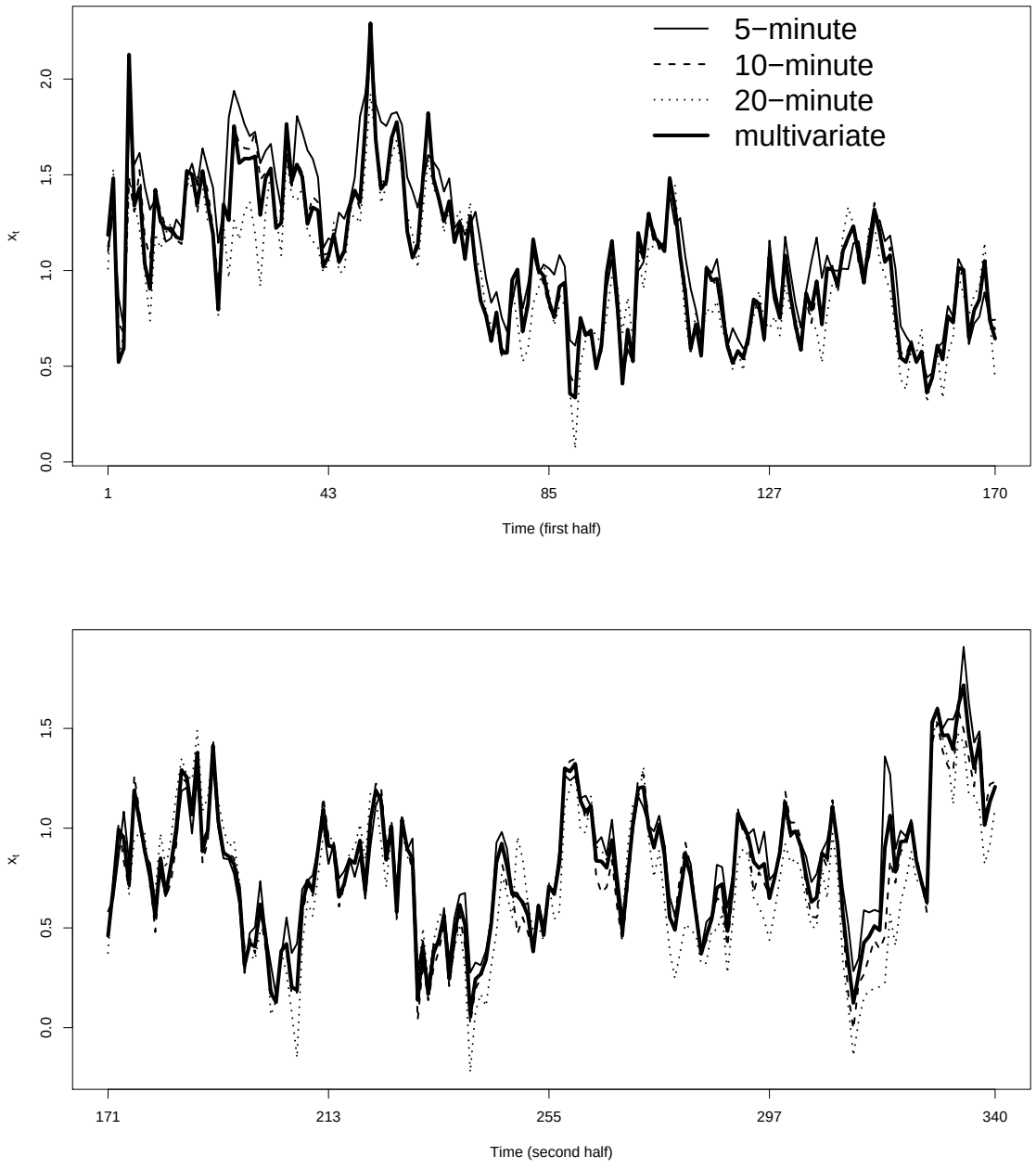

Figure 22: *Realized volatility.* Sequential 50th percentiles of $p(x_{it}|y^t, \mathcal{M}_1)$ for $i = 5, 10, 20$ and $p(x_t|y^t, \mathcal{M}_2)$, where $\mathcal{M}_1$ is the univariate local level model and $\mathcal{M}_2$ its multivariate version.

Finally, we like to express our sincere appreciation of the achievements of Professor Rudolf E. Kalman. His works have led to many new developments in scientific computing, statistical inference, and applications. Particle filters are one more example that will have a long lasting impact in our profession.

References

Alspach, D. L. and Sorenson, H. W. (1972). Nonlinear Bayesian estimation using Gaussian sum approximation. *IEEE Transactions on Automatic Control*, 17, 439-48.

Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society, Series B*, 64, 827-36.

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

Andrieu, C., Doucet, A., Singh, S. S. and Tadić, V. B. (2004). Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92, 423-38.

Andrieu, C., Davy, M. and Doucet, A. (2003). Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions. *IEEE Transactions on Signal Processing*, 51, 1762-70.

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.

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.

Briers, M., Doucet, A. and Maskell, S. (2010). Smoothing algorithms for state-space models. *Annals Institute Statistical Mathematics*, 62, 61-89.

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.

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.

Carpenter J., Clifford P. and Fearnhead P. (1999). An improved particle filter for non-linear problems. *IEE Proceedings on Radar, Sonar and Navigation*, 146, 2-7.

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

Carvalho, C. M., Johannes, M., Lopes, H. F. and Polson, N. (2010). Particle learning and smoothing. *Statistical Science* (to appear).

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.

Carvalho, C. M., Lopes, H. F. and Polson, N. (2009). Particle learning for generalized dynamic conditionally linear models. *Working Paper*. The University of Chicago Booth School of Business.

Carvalho, C. M., Lopes, H. F., Polson, N. G. and Taddy, M. (2009). Particle learning for general mixtures. *Working Paper*. The University of Chicago Booth School of Business.

Chen, H., Petralia, F. and Lopes, H. F. (2009). Sequential Monte Carlo estimation of DSGE models. *Working Paper*. The University of Chicago Booth School of Business.

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

Chen, Y. and Lai, T. L. (2007). Identification and Adaptive Control of Change-Point ARX Models Via Rao-Blackwellized Particle Filters. *IEEE Transactions on Automatic Control*, 52, 67-72.

Chen, Z. (2003). Bayesian filtering: From Kalman filters to particle filters, and beyond. *Working paper*, McMaster University, Canada.

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

Creal, D. (2008). Analysis of filtering and smoothing algorithms for Lévy-driven stochastic volatility models. *Computational Statistics & Data Analysis*, 52, 2863-76.

Da Silva, C. Q., Migon, H. S. and Correira, L. T. (2009). Bayesian beta dynamic model and applications. *Working Paper*. Department of Statistics, Federal University of Rio de Janeiro.

DeJong, D. N., Dharmarajan, H., Liesenfeld, R., Moura, G. V. and Richard, J.-F. (2009). Efficient likelihood evaluation of state-space representations. *Working Paper*. Department of Economics, University of Pittsburgh.

Del Moral, P., Doucet, A. and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B*, 68, 411-36.

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. *Working paper*. Institut Télécom, Paris.

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

Doucet, A., Briers, M. and Sénécal, S. (2006). Efficient block sampling strategies for sequential Monte Carlo. *Journal of Computational and Graphical Statistics*, 15, 693-711.

Doucet, A., De Freitas, N. , and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag.

Doucet, A., Gordon, N. J. and Krishnamurthy, V. (2001). Particle filters for state estimation of jump Markov linear systems, *IEEE Transaction on Signal Processing*, 49, 613-24.

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

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.

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

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.

Dukić, V., Lopes, H. F. and Polson, N. G. (2009). Tracking flu epidemics using Google trends and particle learning. *Working Paper*. The University of Chicago Booth School of Business.

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

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

Fearnhead, F. (2005). Using Random Quasi-Monte-Carlo within Particle Filters, with Application to Financial Time Series. *Journal of Computational and Graphical Statistics*, 14, 751-69.

Fearnhead, P. and Clifford, P. (2003). Online inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society, Series B*, 65, 887-99.

Fearnhead, P., Papaspiliopoulos, O. and Roberts, G. O. (2008). Particle filters for partially-observed diffusions. *Journal of the Royal Statistical Society, Series B*, 70, 755-77.

Fearnhead, P. and Meligkotsidou, L. (2004). Exact filtering for partially-observed continuous-time models. *Journal of the Royal Statistical Society, Series B*, 66, 771-789.

Fearnhead, P., Wyncoll, D. and Tawn, J. (2008). A sequential smoothing algorithm with linear computational cost. *Working paper*. Department of Mathematics and Statistics, Lancaster University.

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

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

Flury, T. and Shephard, N. (2008). Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models. *Working paper*. Department of Economics, University of Oxford.

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

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

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.

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.

Gramacy, R. and Polson, N. G. (2010). Particle learning of Gaussian process models for sequential design and optimization. *Working Paper*. The University of Chicago Booth School of Business.

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

Han, C. and Carlin, B. P. (2000). MCMC Methods for Computing Bayes Factors: A Comparative Review. *Journal of the American Statistical Association*, 96, 1122-32.

Harvey, A. C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press: Cambridge.

Ito, K. and Xiong, K. (2000). Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45, 910-27.

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.

Jasra, A., Stephens, D. A., Doucet, A. and Tsagaris, T. (2008). Inference for Lévy-driven stochastic volatility models via adaptive sequential Monte Carlo. *Working paper*. Institute for Statistical Mathematics, Tokyo, Japan.

Jazwinski, A. (1970). *Stochastic Processes and Filtering Theory*. New York: Academic.

Johannes, M., A. Korteweg, and N. G. Polson (2008). Sequential learning, predictive regressions, and optimal portfolio returns. *Working paper*, Graduate School of Business, Stanford University.

Johannes, M. and Polson, P. (2009). Particle Filtering. In T. G. Andersen, R. A. Davis, J.-P. Kreiss and T. Mikosch, editors, *Handbook of Financial Time Series*. Berlin: Springer-Verlag.

Johannes, M. S., Polson, N. G. and Stroud, J. R. (2009). Optimal filtering of jump diffusions: extracting latent states from asset prices. *Review of Financial Studies*, 22, 2559-99.

Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *The Proceedings of AeroSense: The 11th International Symposium on Aerospace, Defense Sensing, Simulation and Controls*, 3068, 182-93.

Kass, R. E. and Raftery, A. (1995). *Bayes Factors*. *Journal of the American Statistical Association*, 90, 773-95.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5, 1-25.

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.

Li, H., Wells, M. T. and Yu, C. L. (2008). A Bayesian Analysis of Return Dynamics with Lévy Jumps. *Review of Financial Studies*, 21, 2345-78.

Li, H. (2009). Sequential Bayesian Analysis of Time-Changed Infinite Activity Derivatives Pricing Models. *Working paper*. ESSEC Business School, Paris-Singapore.

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

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

Liu, J. and R. Chen (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93, 1032-44.

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.

Lopes, H. F., Carvalho, C. M., Johannes, M. and Polson, N. G. (2010). Particle learning for sequential Bayesian computation. 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 (to appear).

Lopes, H. F. and Polson, N. G. (2010). 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.

Lopes, H. F. and West, M. (2004). Bayesian model assessment in factor analysis. *Statistica Sinica*, 14, 41-67.

Lund, B. and Lopes, H. F. (2009). Learning in a regime switching macro-finance model for the term structure. *Working Paper*. The University of Chicago Booth School of Business.

MacEachern S. N., Clyde M. A., and Liu J. S. (1999). Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics*, 27, 251-67.

Mukherjee, C. and West, M. (2009). Sequential Monte Carlo in model comparison: Example in cellular dynamics in systems biology. *Working Paper*. Department of Statistical Science, Duke University.

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.

Petris, G., Petrone, S. and Campagnoli, P. (2009). *Dynamic Linear Models with R*. New York: Springer.

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

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.

Poyiadjis, G., Doucet, A. and Singh, S. S. (2005). Particle methods for optimal filter derivative: Application to parameter estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 5, v/925-v/928.

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.

Raggi, D. and Bordignon, S. (2006). Sequential Monte Carlo methods for stochastic volatility models with jumps. *Working paper*. Department of Economics, University of Bologna.

Reis, E. A., Salazar, E. and Gamerman, D. (2006). Comparison of sampling schemes for dynamic linear models. *International Statistical Review*, 74, 203-14.

Rios, M. P. and Lopes, H. F. (2009). Sequential parameter estimation in stochastic volatility models. *Working Paper*. The University of Chicago Booth School of Business.

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

Shephard, N. (1994). Partial non-Gaussian state space. *Biometrika*, 81, 115-31.

Shi, M. and Dunson, D. B. (2009). Bayesian variable selection via particle stochastic search. *Working Paper*. Department of Statistical Science, Duke University.

Smith, A. F. M. and Gelfand, A. E. (1992). Bayesian statistics without tears: a sampling-resampling perspective. *American Statistician*, 46, 84-8.

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

Stroud, J., Polson, N. and Müller, P. (2004). Practical filtering for stochastic volatility models. In A. Harvey, S. J. Koopman, and N. Shephard, editors, *State Space and Unobserved Component Models*. Cambridge: Cambridge University Press, 236-247.

Taddy, M., Gramacy, R. and Polson, N. G. (2010). Dynamic trees for learning and design *Working Paper*. The University of Chicago Booth School of Business.

Van der Merwe, R., Doucet, D., De Freitas, N. and Wan, E. (2000). The unscented particle filter. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems, Volume 13*. Cambridge: MIT Press.

West, M. (1992). Modelling with mixtures. In J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith, editors, *Bayesian Statistics 4*. Oxford: Clarendon.

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

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

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

West, M., Harrison, J. and Migon, H. (1985). Dynamic generalized linear models and Bayesian forecasting. *Journal of the American Statistical Association*, 80, 73-83.