On Particle Methods for Parameter Estimation in State-Space Models

Nikolas Kantas, Arnaud Doucet, Sumeetpal S. Singh, Jan Maciejowski and Nicolas Chopin

Abstract. Nonlinear non-Gaussian state-space models are ubiquitous in statistics, econometrics, information engineering and signal processing. Particle methods, also known as Sequential Monte Carlo (SMC) methods, provide reliable numerical approximations to the associated state inference problems. However, in most applications, the state-space model of interest also depends on unknown static parameters that need to be estimated from the data. In this context, standard particle methods fail and it is necessary to rely on more sophisticated algorithms. The aim of this paper is to present a comprehensive review of particle methods that have been proposed to perform static parameter estimation in state-space models. We discuss the advantages and limitations of these methods and illustrate their performance on simple models.

Key words and phrases: Bayesian inference, maximum likelihood inference, particle filtering, Sequential Monte Carlo, state-space models.

1. INTRODUCTION

State-space models, also known as hidden Markov models, are a very popular class of time series models that have found numerous of applications in fields as diverse as statistics, ecology, econometrics, engineering and environmental sciences; see [11, 30, 34, 87]. Formally, a state-space model is defined by two stochastic processes $\{X_n\}_{n\geq0}$ and $\{Y_n\}_{n\geq0}$. The process $\{X_n\}_{n\geq0}$ is an $\mathcal{X}$-valued latent Markov process of initial density $\mu_\theta(x)$ and Markov transition density

$f_\theta(x'|x)$, that is,

$$X_0 \sim \mu_\theta(x_0),$$
$$X_n|(X_{0:n-1} = x_{0:n-1}) \sim f_\theta(x_n|x_{n-1}),$$

(1.1)

whereas the $\mathcal{Y}$-valued observations $\{Y_n\}_{n\geq0}$ satisfy

$$Y_n|(X_{0:n} = x_{0:n}, Y_{0:n-1} = y_{0:n-1})$$
$$\sim g_\theta(y_n|x_n),$$

(1.2)

where $g_\theta(y|x)$ denotes the conditional marginal density, $\theta \in \Theta$ the parameter of the model and $z_{i:j}$ denotes components $(z_i, z_{i+1}, \ldots, z_j)$ of a sequence $\{z_n\}$. The spaces $\mathcal{X}$ and $\mathcal{Y}$ can be Euclidean, but what follows applies to more general state spaces as well.

The popularity of state-space models stems from the fact that they are flexible and easily interpretable. Applications of state-space models include stochastic volatility models where X_n is the volatility of an asset and Y_n its observed log-return [52], biochemical network models where X_n corresponds to the population of various biochemical species and Y_n are imprecise measurements of the size of a subset of these species [93], neuroscience models where X_n is a state vector determining the neuron's stimulus–response function and Y_n some spike train data [77]. However, nonlinear

N. Kantas is Lecturer, Department of Mathematics, Imperial College London, London SW7 2BZ, United Kingdom (e-mail: n.kantas@imperial.ac.uk). A. Doucet is Professor, Department of Statistics, University of Oxford, Oxford OX1 3TG, United Kingdom (e-mail: doucet@stats.ox.ac.uk). S. S. Singh is Senior Lecturer, Department of Engineering, University of Cambridge, Cambridge CB1 2PZ, United Kingdom (e-mail: sss40@eng.cam.ac.uk). J. M. Maciejowski is Professor, Department of Engineering, University of Cambridge, Cambridge CB1 2PZ, United Kingdom (e-mail: jmm@eng.cam.ac.uk). N. Chopin is Professor, CREST-ENSAE and HEC Paris, 99245 Malakoff, France (e-mail: nicolas.chopin@ensae.fr).

non-Gaussian state-space models are also notoriously difficult to fit to data and it is only recently, thanks to the advent of powerful simulation techniques, that it has been possible to fully realize their potential.

To illustrate the complexity of inference in state-space models, consider first the scenario where the parameter θ is *known*. On-line and off-line inference about the state process $\{X_n\}$ given the observations $\{Y_n\}$ is only feasible analytically for simple models such as the linear Gaussian state-space model. In non-linear non-Gaussian scenarios, numerous approximation schemes, such as the Extended Kalman filter or the Gaussian sum filter [1], have been proposed over the past fifty years to solve these so-called optimal filtering and smoothing problems, but these methods lack rigor and can be unreliable in practice in terms of accuracy, while deterministic integration methods are difficult to implement. Markov chain Monte Carlo (MCMC) methods can obviously be used, but they are impractical for on-line inference; and even for off-line inference, it can be difficult to build efficient high-dimensional proposal distributions for such algorithms. For nonlinear non-Gaussian state-space models *particle algorithms* have emerged as the most successful. Their widespread popularity is due to the fact that they are easy to implement, suitable for parallel implementation [60] and, more importantly, have been demonstrated in numerous settings to yield more accurate estimates than the standard alternatives, for example, see [11, 23, 30, 67].

In most practical situations, the model (1.1)–(1.2) depends on an *unknown* parameter vector θ that needs to be inferred from the data either in an on-line or off-line manner. In fact inferring the parameter θ is often the primary problem of interest; for example, for biochemical networks, we are not interested in the population of the species per se, but we want to infer some chemical rate constants, which are parameters of the transition prior $f_\theta(x'|x)$. Although it is possible to define an extended state that includes the original state X_n and the parameter θ and then apply standard particle methods to perform parameter inference, it was recognized very early on that this naive approach is problematic [54] due to the parameter space not being explored adequately. This has motivated over the past fifteen years the development of many particle methods for the parameter estimation problem, but numerically robust methods have only been proposed recently. The main objective of this paper is to provide a comprehensive overview of this literature. This paper thus differs from recent survey papers on particle methods

which all primarily focus on estimating the state sequence $X_{0:n}$ or discuss a much wider range of topics, for example, [32, 55, 58, 65]. We will present the main features of each method and comment on their pros and cons. No attempt, however, is made to discuss the intricacies of the specific implementations. For this we refer the reader to the original references.

We have chosen to broadly classify the methods as follows: Bayesian or Maximum Likelihood (ML) and whether they are implemented off-line or on-line. In the Bayesian approach, the unknown parameter is assigned a prior distribution and the posterior density of this parameter given the observations is to be characterized. In the ML approach, the parameter estimate is the maximizing argument of the likelihood of θ given the data. Both these inference procedures can be carried out off-line or on-line. Specifically, in an off-line framework we infer θ using a fixed observation record $y_{0:T}$. In contrast, on-line methods update the parameter estimate sequentially as observations $\{y_n\}_{n\geq 0}$ become available.

The rest of the paper is organized as follows. In Section 2 we present the main computational challenges associated to parameter inference in state-space models. In Section 3 we review particle methods for filtering when the model does not include any unknown parameters, whereas Section 4 is dedicated to smoothing. These filtering and smoothing techniques are at the core of the off-line and on-line ML parameter procedures described in Section 5. In Section 6 we discuss particle methods for off-line and on-line Bayesian parameter inference. The performance of some of these algorithms are illustrated on simple examples in Section 7. Finally, we summarize the main advantages and drawbacks of the methods presented and discuss some open problems in Section 8.

2. COMPUTATIONAL CHALLENGES ASSOCIATED TO PARAMETER INFERENCE

A key ingredient of ML and Bayesian parameter inference is the likelihood function $p_\theta(y_{0:n})$ of θ which satisfies

$$(2.1) \qquad p_\theta(y_{0:n}) = \int p_\theta(x_{0:n}, y_{0:n}) \, dx_{0:n},$$

where $p_\theta(x_{0:n}, y_{0:n})$ denotes the joint density of $(X_{0:n}, Y_{0:n})$ which is given from equations (1.1)–(1.2) by

$$(2.2) \begin{aligned} &p_\theta(x_{0:n}, y_{0:n}) \\ &= \mu_\theta(x_0) \prod_{k=1}^{n} f_\theta(x_k|x_{k-1}) \prod_{k=0}^{n} g_\theta(y_k|x_k). \end{aligned}$$

The likelihood function is also the normalizing constant of the posterior density $p_\theta(x_{0:n}|y_{0:n})$ of the latent states $X_{0:n}$ given data $y_{0:n}$,

$$(2.3) \qquad p_\theta(x_{0:n}|y_{0:n}) = \frac{p_\theta(x_{0:n}, y_{0:n})}{p_\theta(y_{0:n})}.$$

This posterior density is itself useful for computing the score vector $\nabla_\theta \ell_n(\theta)$ associated to the log-likelihood $\ell_n(\theta) = \log p_\theta(y_{0:n})$, as Fisher's identity yields

$$\nabla_\theta \ell_n(\theta) = \int \nabla_\theta \log p_\theta(x_{0:n}, y_{0:n})$$
$$(2.4)$$
$$\cdot p_\theta(x_{0:n}|y_{0:n}) \, dx_{0:n}.$$

The main practical issue associated to parameter inference in nonlinear non-Gaussian state-space models is that the likelihood function is intractable. As performing ML parameter inference requires maximizing this intractable function, it means practically that it is necessary to obtain reasonably low-variance Monte Carlo estimates of it, or of the associated score vector if this maximization is carried out using gradient-based methods. Both tasks involve approximating high-dimensional integrals, (2.1) and (2.4), whenever n is large. On-line inference requires additionally that these integrals be approximated on the fly, ruling out the applications of standard computational tools such as MCMC.

Bayesian parameter inference is even more challenging, as it requires approximating the posterior density

$$(2.5) \qquad p(\theta|y_{0:n}) = \frac{p_\theta(y_{0:n}) p(\theta)}{\int p_\theta(y_{0:n}) p(\theta) \, d\theta},$$

where $p(\theta)$ is the prior density. Here not only $p_\theta(y_{0:n})$ but also $p(y_{0:n}) = \int p_\theta(y_{0:n}) p(\theta) \, d\theta$ are intractable and, once more, these integrals must be approximated on-line if one wants to update the posterior density sequentially. We will show in this review that particle methods are particularly well suited to these integration tasks.

3. FILTERING AND PARTICLE APPROXIMATIONS

In this section the parameter θ is assumed known and we focus on the problem of estimating the latent process $\{X_n\}_{n\geq0}$ sequentially given the observations. An important by-product of this so-called filtering task from a parameter estimation viewpoint is that it provides us with an on-line scheme to compute $\{p_\theta(y_{0:n})\}_{n\geq0}$. As outlined in Section 2, the particle approximation of these likelihood terms is a key ingredient of numerous particle-based parameter inference techniques discussed further on.

3.1 Filtering

Filtering usually denotes the task of estimating recursively in time the sequence of marginal posteriors $\{p_\theta(x_n|y_{0:n})\}_{n\geq0}$, known as the filtering densities. However, we will adopt here a more general definition and will refer to filtering as the task of estimating the sequence of joint posteriors $\{p_\theta(x_{0:n}|y_{0:n})\}_{n\geq0}$ recursively in time, but we will still refer to the marginals $\{p_\theta(x_n|y_{0:n})\}_{n\geq0}$ as the filtering densities.

It is easy to verify from (2.1) and (2.3) that the posterior $p_\theta(x_{0:n}|y_{0:n})$ and the likelihood $p_\theta(y_{0:n})$ satisfy the following fundamental recursions: for $n \geq 1$,

$$(3.1) \qquad \begin{aligned} &p_\theta(x_{0:n}|y_{0:n}) \\ &= p_\theta(x_{0:n-1}|y_{0:n-1}) \frac{f_\theta(x_n|x_{n-1}) g_\theta(y_n|x_n)}{p_\theta(y_n|y_{0:n-1})} \end{aligned}$$

and

$$(3.2) \qquad p_\theta(y_{0:n}) = p_\theta(y_{0:n-1}) p_\theta(y_n|y_{0:n-1}),$$

where

$$(3.3) \qquad \begin{aligned} &p_\theta(y_n|y_{0:n-1}) \\ &= \int g_\theta(y_n|x_n) f_\theta(x_n|x_{n-1}) \\ &\qquad \cdot p_\theta(x_{n-1}|y_{0:n-1}) \, dx_{n-1:n}. \end{aligned}$$

There are essentially two classes of models for which $p_\theta(x_{0:n}|y_{0:n})$ and $p_\theta(y_{0:n})$ can be computed exactly: the class of linear Gaussian models, for which the above recursions may be implemented using Kalman techniques, and when $\mathcal{X}$ is a finite state space; see, for example, [11]. For other models these quantities are typically intractable, that is, the densities in (3.1)–(3.3) cannot be computed exactly.

3.2 Particle Filtering

3.2.1 *Algorithm.* Particle filtering methods are a set of simulation-based techniques which approximate numerically the recursions (3.1) to (3.3). We focus here on the APF (auxiliary particle filter [78]) for two reasons: first, this is a popular approach, in particular, in the context of parameter estimation (see, e.g., Section 6.2.3); second, the APF covers as special cases a large class of particle algorithms, such as the bootstrap filter [46] and SISR (Sequential Importance Sampling Resampling [31, 69]).

Let

$$(3.4) \quad q_\theta(x_n, y_n|x_{n-1}) = q_\theta(x_n|y_n, x_{n-1}) q_\theta(y_n|x_{n-1}),$$

where $q_\theta(x_n|y_n, x_{n-1})$ is a probability density function which is easy to sample from and $q_\theta(y_n|x_{n-1})$ is not

necessarily required to be a probability density function but just a nonnegative function of $(x_{n-1}, y_n) \in \mathcal{X} \times \mathcal{Y}$ one can evaluate. [For $n = 0$, remove the dependency on x_{n-1}, i.e., $q_\theta(x_0, y_0) = q_\theta(x_0|y_0)q_\theta(y_0)$.]

The algorithm relies on the following importance weights:

$$(3.5) \qquad w_0(x_0) = \frac{g_\theta(y_0|x_0)\mu_\theta(x_0)}{q_\theta(x_0|y_0)},$$

$$w_n(x_{n-1:n}) = \frac{g_\theta(y_n|x_n)f_\theta(x_n|x_{n-1})}{q_\theta(x_n, y_n|x_{n-1})}$$

$$(3.6)$$
$$\text{for } n \geq 1.$$

In order to alleviate the notational burden, we omit the dependence of the importance weights on θ; we will do so in the remainder of the paper when no confusion is possible. The auxiliary particle filter can be summarized in Algorithm 1 [12, 78].

One recovers the SISR algorithm as a special case of Algorithm 1 by taking $q_\theta(y_n|x_{n-1}) = 1$ [or, more generally, by taking $q_\theta(y_n|x_{n-1}) = h_\theta(y_n)$, some arbitrary positive function]. Further, one recovers the bootstrap filter by taking $q_\theta(x_n|y_n, x_{n-1}) = f_\theta(x_n|x_{n-1})$. This is an important special case, as some complex models are such that one may sample from $f_\theta(x_n|x_{n-1})$, but not compute the corresponding density; in such a case the bootstrap filter is the only implementable algorithm. For models such that the density $f_\theta(x_n|x_{n-1})$ is tractable, [78] recommend selecting $q_\theta(x_n|y_n, x_{n-1}) = p_\theta(x_n|y_n, x_{n-1})$ and $q_\theta(y_n|x_{n-1}) = p_\theta(y_n|x_{n-1})$ when these quantities are tractable, and using approximations of these quantities in scenarios when they are not. The intuition for these recommendations is that this should make the weight function (3.6) nearly constant.

The computational complexity of Algorithm 1 is $\mathcal{O}(N)$ per time step; in particular, see, for example,

[31], page 201, for a $\mathcal{O}(N)$ implementation of the resampling step. At time n, the approximations of $p_\theta(x_{0:n}|y_{0:n})$ and $p_\theta(y_n|y_{0:n-1})$ presented earlier in (2.3) and (3.3), respectively, are given by

$$(3.7) \quad \hat{p}_\theta(dx_{0:n}|y_{0:n}) = \sum_{i=1}^N W_n^i \delta_{X_{0:n}^i}(dx_{0:n}),$$

$$\hat{p}_\theta(y_n|y_{0:n-1}) = \left(\frac{1}{N}\sum_{i=1}^N w_n(X_{n-1:n}^i)\right)$$
$$(3.8)$$
$$\cdot \left(\sum_{i=1}^N W_{n-1}^i q_\theta(y_n|X_{n-1}^i)\right),$$

where $W_n^i \propto w_n(X_{n-1:n}^i)$, $\sum_{i=1}^N W_n^i = 1$ and $\hat{p}_\theta(y_0) = \frac{1}{N}\sum_{i=1}^N w_0(X_0^i)$. In practice, one uses (3.7) mostly to obtain approximations of posterior moments

$$\sum_{i=1}^N W_n^i \varphi(X_{0:n}^i) \approx \mathbb{E}[\varphi(X_{0:n})|y_{0:n}],$$

but expressing particle filtering as a method for approximating distributions (rather than moments) turns out to be a more convenient formalization. The likelihood (3.2) is then estimated through

$$(3.9) \qquad \hat{p}_\theta(y_{0:n}) = \hat{p}_\theta(y_0) \prod_{k=1}^n \hat{p}_\theta(y_k|y_{0:k-1}).$$

The resampling procedure is introduced to replicate particles with high weights and discard particles with low weights. It serves to focus the computational efforts on the "promising" regions of the state space. We have presented above the simplest resampling scheme. Lower variance resampling schemes have been proposed in [53, 69], as well as more advanced particle

---

Algorithm 1 Auxiliary particle filtering

- *At time $n = 0$, for all $i \in \{1, \ldots, N\}$:*

  1. Sample $X_0^i \sim q_\theta(x_0|y_0)$.
  2. Compute $\overline{W}_1^i \propto w_0(X_0^i)q_\theta(y_1|X_0^i)$, $\sum_{i=1}^N \overline{W}_1^i = 1$.
  3. Resample $\overline{X}_0^i \sim \sum_{i=1}^N \overline{W}_1^i \delta_{X_0^i}(dx_0)$.

- *At time $n \geq 1$, for all $i \in \{1, \ldots, N\}$:*

  1. Sample $X_n^i \sim q_\theta(x_n|y_n, \overline{X}_{n-1}^i)$ and set $X_{0:n}^i \leftarrow (\overline{X}_{0:n-1}^i, X_n^i)$.
  2. Compute $\overline{W}_{n+1}^i \propto w_n(X_{n-1:n}^i)q_\theta(y_{n+1}|X_n^i)$, $\sum_{i=1}^N \overline{W}_{n+1}^i = 1$.
  3. Resample $\overline{X}_{0:n}^i \sim \sum_{i=1}^N \overline{W}_{n+1}^i \delta_{X_{0:n}^i}(dx_{0:n})$.

algorithms with better overall performance, for example, the Resample–Move algorithm [44]. For the sake of simplicity, we have also presented a version of the algorithm that operates resampling at every iteration n. It may be more efficient to trigger resampling only when a certain criterion regarding the degeneracy of the weights is met; see [31] and [68], pages 35 and 74.

3.2.2 *Convergence results.* Many sharp convergence results are available for particle methods [23]. A selection of these results that gives useful insights on the difficulties of estimating static parameters with particle methods is presented below.

Under minor regularity assumptions, one can show that for any $n \geq 0$, $N > 1$ and any bounded test function $\varphi_n : \mathcal{X}^{n+1} \to [-1, 1]$, there exist constants $A_{\theta,n,p} < \infty$ such that for any $p \geq 1$

$$
\begin{aligned}
\mathbb{E}\bigg[\bigg|\int \varphi_n(x_{0:n}) \\
(3.10) \qquad \cdot \big\{\hat{p}_\theta(dx_{0:n}|y_{0:n}) - p_\theta(dx_{0:n}|y_{0:n})\big\}\bigg|^p\bigg] \\
\leq \frac{A_{\theta,n,p}}{N^{p/2}},
\end{aligned}
$$

where the expectation is with respect to the law of the particle filter. In addition, for more general classes of functions, we can obtain for any fixed n a Central Limit Theorem (CLT) as $N \to +\infty$ ([17] and [23], Proposition 9.4.2). Such results are reassuring but weak, as they reveal nothing regarding long-time behavior. For instance, without further restrictions on the class of functions φ_n and the state-space model, $A_{\theta,n,p}$ typically grows exponentially with n. This is intuitively not surprising, as the dimension of the target density $p_\theta(x_{0:n}|y_{0:n})$ is increasing with n. Moreover, the successive resampling steps lead to a depletion of the particle population; $p_\theta(x_{0:m}|y_{0:n})$ will eventually be approximated by a single unique particle as $n - m$ increases. This is referred to as the *degeneracy* problem in the literature ([11], Figure 8.4, page 282). This is a fundamental weakness of particle methods: given a fixed number of particles N, it is impossible to approximate $p_\theta(x_{0:n}|y_{0:n})$ accurately when n is large enough.

Fortunately, it is also possible to establish much more positive results. Many state-space models possess the so-called *exponential forgetting* property ([23], Chapter 4). This property states that for any $x_0, x'_0 \in \mathcal{X}$ and data $y_{0:n}$, there exist constants $B_\theta < \infty$ and $\lambda \in [0, 1)$ such that

$$
\begin{aligned}
\|p_\theta(dx_n|y_{1:n}, x_0) - p_\theta(dx_n|y_{1:n}, x'_0)\|_{\mathrm{TV}} \\
(3.11) \qquad \leq B_\theta \lambda^n,
\end{aligned}
$$

where $\|\cdot\|_{\mathrm{TV}}$ is the total variation distance, that is, the optimal filter forgets exponentially fast its initial condition. This property is typically satisfied when the signal process $\{X_n\}_{n\geq 0}$ is a uniformly ergodic Markov chain and the observations $\{Y_n\}_{n\geq 0}$ are not too informative ([23], Chapter 4), or when $\{Y_n\}_{n\geq 0}$ are informative enough that it effectively restricts the hidden state to a bounded region around it [76]. Weaker conditions can be found in [29, 90]. When exponential forgetting holds, it is possible to establish much stronger uniform-in-time convergence results *for functions φ_n that depend only on recent states*. Specifically, for an integer $L > 0$ and any bounded test function $\Psi_L : \mathcal{X}^L \to [-1, 1]$, there exist constants $C_{\theta,L,p} < \infty$ such that for any $p \geq 1$, $n \geq L - 1$,

$$
\begin{aligned}
\mathbb{E}\bigg[\bigg|\int_{\mathcal{X}^L} \Psi(x_{n-L+1:n})\Delta_{\theta,n}(dx_{n-L+1:n})\bigg|^p\bigg] \\
(3.12) \qquad \leq \frac{C_{\theta,L,p}}{N^{p/2}},
\end{aligned}
$$

where

$$
\begin{aligned}
\Delta_{\theta,n}(dx_{n-L+1:n}) \\
(3.13) \qquad = \int_{x_{0:n-L}\in\mathcal{X}^{n-L+1}} \big\{\hat{p}_\theta(dx_{0:n}|y_{0:n}) \\
- p_\theta(dx_{0:n}|y_{0:n})\big\}.
\end{aligned}
$$

This result explains why particle filtering is an effective computational tool in many applications such as tracking, where one is only interested in $p_\theta(x_{n-L+1:n}|y_{0:n})$, as the approximation error is uniformly bounded over time.

Similar positive results hold for $\hat{p}_\theta(y_{0:n})$. This estimate is unbiased for any $N \geq 1$ ([23], Theorem 7.4.2, page 239), and, under assumption (3.11), the *relative variance* of the likelihood estimate $\hat{p}_\theta(y_{0:n})$, that is the variance of the ratio $\hat{p}_\theta(y_{0:n})/p_\theta(y_{0:n})$, is bounded above by $D_\theta n/N$ [14, 90]. This is a great improvement over the exponential increase with n that holds for standard importance sampling techniques; see, for instance, [32]. However, the constants $C_{\theta,L,p}$ and D_θ are typically exponential in n_x, the dimension of the state vector X_n. We note that nonstandard particle methods designed to minimize the variance of the estimate of $p_\theta(y_{0:n})$ have recently been proposed [92].

Finally, we recall the theoretical properties of particles estimates of the following so-called smoothed additive functional ([11], Section 8.3 and [74]),

$$
\begin{aligned}
(3.14) \qquad \mathcal{S}_n^\theta = \int_{\mathcal{X}^{n+1}} \bigg\{\sum_{k=1}^n s_k(x_{k-1:k})\bigg\} \\
\cdot p_\theta(x_{0:n}|y_{0:n})\,dx_{0:n}.
\end{aligned}
$$

Such quantities are critical when implementing ML parameter estimation procedures; see Section 5. If we substitute $\hat{p}_\theta(dx_{0:n}|y_{0:n})$ to $p_\theta(x_{0:n}|y_{0:n})\,dx_{0:n}$ to approximate $\mathcal{S}_n^\theta$, then we obtain an estimate $\widehat{\mathcal{S}}_n^\theta$ which can be computed recursively in time; see, for example, [11], Section 8.3. For the remainder of this paper we will refer to this approximation as the *path space* approximation. Even when (3.11) holds, there exists $0 < F_\theta, G_\theta < \infty$ such that the asymptotic bias [23] and variance [81] satisfy

$$(3.15) \quad |\mathbb{E}(\widehat{\mathcal{S}}_n^\theta) - \mathcal{S}_n^\theta| \leq F_\theta \frac{n}{N}, \quad \mathbb{V}(\widehat{\mathcal{S}}_n^\theta) \geq G_\theta \frac{n^2}{N}$$

for $s_p : \mathcal{X}^2 \to [-1, 1]$ where the variance is w.r.t. the law of the particle filter. The fact that the variance grows at least quadratically in time follows from the degeneracy problem and makes $\widehat{\mathcal{S}}_n^\theta$ unsuitable for some on-line likelihood based parameter estimation schemes discussed in Section 5.

4. SMOOTHING

In this section the parameter θ is still assumed known and we focus on smoothing, that is, the problem of estimating the latent variables $X_{0:T}$ given a fixed batch of observations $y_{0:T}$. Smoothing for a fixed parameter θ is at the core of the two main particle ML parameter inference techniques described in Section 5, as these procedures require computing smoothed additive functionals of the form (3.14). Clearly, one could unfold the recursion (3.1) from $n = 0$ to $n = T$ to obtain $p_\theta(x_{0:T}|y_{0:T})$. However, as pointed out in the previous section, the path space approximation (3.7) suffers from the degeneracy problem and yields potentially high variance estimates of (3.14) as (3.15) holds. This has motivated the development of alternative particle approaches to approximate $p_\theta(x_{0:T}|y_{0:T})$ and its marginals.

4.1 Fixed-lag Approximation

For state-space models with "good" forgetting properties [e.g., (3.11)], we have

$$(4.1) \quad p_\theta(x_{0:n}|y_{0:T}) \approx p_\theta(x_{0:n}|y_{0:(n+L)\wedge T})$$

for L large enough, that is, observations collected at times $k > n + L$ do not bring any significant additional information about $X_{0:n}$. In particular, when having to evaluate $\mathcal{S}_T^\theta$ of the form (3.14) we can approximate the expectation of $s_n(x_{n-1:n})$ w.r.t. $p_\theta(x_{n-1:n}|y_{0:T})$ by its expectation w.r.t. $p_\theta(x_{n-1:n}|y_{0:(n+L)\wedge T})$.

Algorithmically, a particle implementation of (4.1) means not resampling the components $X_{0:n}^i$ of the particles $X_{0:k}^i$ obtained by particle filtering at times $k > n + L$. This was first suggested in [56] and used in [11], Section 8.3, and [74]. This algorithm is simple to implement, but the main practical problem is the choice of L. If taken too small, then $p_\theta(x_{0:n}|y_{0:(n+L)\wedge T})$ is a poor approximation of $p_\theta(x_{0:n}|y_{0:T})$. If taken too large, the degeneracy remains substantial. Moreover, even as $N \to \infty$, this particle approximation will have a nonvanishing bias since $p_\theta(x_{0:n}|y_{0:T}) \neq p_\theta(x_{0:n}|y_{0:(n+L)\wedge T})$.

4.2 Forward–Backward Smoothing

4.2.1 *Principle.* The joint smoothing density $p_\theta(x_{0:T}|y_{0:T})$ can be expressed as a function of the filtering densities $\{p_\theta(x_n|y_{0:n})\}_{n=0}^T$ using the following key decomposition:

$$(4.2) \quad \begin{aligned} &p_\theta(x_{0:T}|y_{0:T}) \\ &= p_\theta(x_T|y_{0:T}) \prod_{n=0}^{T-1} p_\theta(x_n|y_{0:n}, x_{n+1}), \end{aligned}$$

where $p_\theta(x_n|y_{0:n}, x_{n+1})$ is a backward (in time) Markov transition density given by

$$(4.3) \quad p_\theta(x_n|y_{0:n}, x_{n+1}) = \frac{f_\theta(x_{n+1}|x_n)\,p_\theta(x_n|y_{0:n})}{p_\theta(x_{n+1}|y_{0:n})}.$$

A backward in time recursion for $\{p_\theta(x_n|y_{0:T})\}_{n=0}^T$ follows by integrating out $x_{0:n-1}$ and $x_{n+1:T}$ in (4.2) while applying (4.3),

$$(4.4) \quad \begin{aligned} &p_\theta(x_n|y_{0:T}) \\ &= p_\theta(x_n|y_{0:n}) \\ &\quad \cdot \int \frac{f_\theta(x_{n+1}|x_n)\,p_\theta(x_{n+1}|y_{0:T})}{p_\theta(x_{n+1}|y_{0:n})}\,dx_{n+1}. \end{aligned}$$

This is referred to as forward–backward smoothing, as a forward pass yields $\{p_\theta(x_n|y_{0:n})\}_{n=0}^T$ which can be used in a backward pass to obtain $\{p_\theta(x_n|y_{0:T})\}_{n=0}^T$. Combined to $\{p_\theta(x_n|y_{0:n}, x_{n+1})\}_{n=0}^{T-1}$, this allows us to obtain $\mathcal{S}_T^\theta$. An alternative to these forward–backward procedures is the generalized two-filter formula [6].

4.2.2 *Particle implementation.* The decomposition (4.2) suggests that it is possible to sample approximately from $p_\theta(x_{0:T}|y_{0:T})$ by running a particle filter from time $n = 0$ to T, storing the approximate filtering distributions $\{\hat{p}_\theta(dx_n|y_{0:n})\}_{n=0}^T$, that is, the marginals of (3.7), then sampling $X_T \sim \hat{p}_\theta(dx_T|y_{0:T})$ and for $n = T - 1, T - 2, \ldots, 0$ sampling $X_n \sim$

$\hat{p}_\theta(dx_n|y_{0:n}, X_{n+1})$ where this distribution is obtained by substituting $\hat{p}_\theta(dx_n|y_{0:n})$ for $p_\theta(dx_n|y_{0:n})$ in (4.3):

$$(4.5) \quad \begin{aligned} &\hat{p}_\theta(dx_n|y_{0:n}, X_{n+1}) \\ &= \frac{\sum_{i=1}^N W_n^i f_\theta(X_{n+1}|X_n^i)\delta_{X_n^i}(dx_n)}{\sum_{i=1}^N W_n^i f_\theta(X_{n+1}|X_n^i)}. \end{aligned}$$

This Forward Filtering Backward Sampling (FFBSa) procedure was proposed in [45]. It requires $\mathcal{O}(N(T+1))$ operations to generate a single path $X_{0:T}$, as sampling from (4.5) costs $\mathcal{O}(N)$ operations. However, as noted in [28], it is possible to sample using rejection from an alternative approximation of $p_\theta(x_n|y_{0:n}, X_{n+1})$ in $\mathcal{O}(1)$ operations if we use an unweighted particle approximation of $p_\theta(x_n|y_{0:n})$ in (4.3) and if the transition prior satisfies $f_\theta(x'|x) \le C < \infty$. Hence, with this approach, sampling a path $X_{0:T}$ costs, on average, only $\mathcal{O}(T+1)$ operations. A related rejection technique was proposed in [48]. In practice, one may generate N such trajectories to compute Monte Carlo averages that approximate smoothing expectations $\mathbb{E}[\varphi(X_{0:T})|y_{0:T}]$. In that scenario, the first approach costs $\mathcal{O}(N^2(T+1))$, while the second approach costs $\mathcal{O}(N(T+1))$ on average. In some applications, the rejection sampling procedure can be computationally costly as the acceptance probability can be very small for some particles; see, for example, Section 4.3 in [75] for empirical results. This has motivated the development of hybrid procedures combining FFBSa and rejection sampling [85].

We can also directly approximate the marginals $\{p_\theta(x_n|y_{0:T})\}_{n=0}^T$. Assuming we have an approximation $\bar{p}_\theta(dx_{n+1}|y_{0:T}) = \sum_{i=1}^N W_{n+1|T}^i \delta_{X_{n+1}^i}(dx_{n+1})$ where $W_{T|T}^i = W_T^i$, then by using (4.4) and (4.5), we obtain the approximation $\bar{p}_\theta(dx_n|y_{0:T}) = \sum_{i=1}^N W_{n|T}^i \delta_{X_n^i}(dx_n)$ with

$$(4.6) \quad W_{n|T}^i = W_n^i \times \sum_{j=1}^N \frac{W_{n+1|T}^j f_\theta(X_{n+1}^j|X_n^i)}{\sum_{l=1}^N W_n^l f_\theta(X_{n+1}^j|X_n^l)}.$$

This Forward Filtering Backward Smoothing (FFBSm, where "m" stands for "marginal") procedure requires $\mathcal{O}(N^2(T+1))$ operations to approximate $\{p_\theta(x_n|y_{0:T})\}_{n=0}^T$ instead of $\mathcal{O}(N(T+1))$ for the path space and fixed-lag methods. However, this high computational complexity of forward–backward estimates can be reduced using fast computational methods [57]. Particle approximations of generalized two-filter smoothing procedures have also been proposed in [6, 38].

4.3 Forward Smoothing

4.3.1 *Principle*. Whenever we are interested in computing the sequence $\{\mathcal{S}_n^\theta\}_{n\ge0}$ recursively in time, the forward–backward procedure described above is cumbersome, as it requires performing a new backward pass with $n+1$ steps at time n. An important but not well-known result is that it is possible to implement exactly the forward–backward procedure using only a forward procedure. This result is at the core of [34], but its exposition relies on tools which are nonstandard for statisticians. We follow here the simpler derivation proposed in [24, 25] which simply consists of rewriting (3.14) as

$$(4.7) \qquad \mathcal{S}_n^\theta = \int V_n^\theta(x_n) p_\theta(x_n|y_{0:n})\, dx_n,$$

where

$$(4.8) \quad \begin{aligned} V_n^\theta(x_n) &:= \int \left\{ \sum_{k=1}^n s_k(x_{k-1:k}) \right\} \\ &\quad \cdot p_\theta(x_{0:n-1}|y_{0:n-1}, x_n)\, dx_{0:n-1}. \end{aligned}$$

It can be easily checked using (4.2) that $V_n^\theta(x_n)$ satisfies the following forward recursion for $n \ge 0$:

$$(4.9) \quad \begin{aligned} V_{n+1}^\theta(x_{n+1}) &= \int \{V_n^\theta(x_n) + s_{n+1}(x_{n:n+1})\} \\ &\quad \cdot p_\theta(x_n|y_{0:n}, x_{n+1})\, dx_n, \end{aligned}$$

with $V_0^\theta(x_0) = 0$ and where $p_\theta(x_n|y_{0:n}, x_{n+1})$ is given by (4.3). In practice, we shall approximate the function V_n^θ on a certain grid of values x_n, as explained in the next section.

4.3.2 *Particle implementation*. We can easily provide a particle approximation of the forward smoothing recursion. Assume you have access to approximations $\{\widehat{V}_n^\theta(X_n^i)\}$ of $\{V_n^\theta(X_n^i)\}$ at time n, where $\hat{p}_\theta(dx_n|y_{0:n}) = \sum_{i=1}^N W_n^i \delta_{X_n^i}(dx_n)$. Then when updating our particle filter to obtain $\hat{p}_\theta(dx_{n+1}|y_{0:n+1}) = \sum_{i=1}^N W_{n+1}^i \delta_{X_{n+1}^i}(dx_{n+1})$, we can directly compute the particle approximations $\{\widehat{V}_{n+1}^\theta(X_{n+1}^i)\}$ by plugging (4.5) and $\hat{p}_\theta(dx_n|y_{0:n})$ in (4.7)–(4.9) to obtain

$$(4.10) \quad \begin{aligned} \widehat{V}_{n+1}^\theta(X_{n+1}^i) &= \left(\sum_{j=1}^N W_n^j f_\theta(X_{n+1}^i|X_n^j) \right. \\ &\quad \left. \cdot \{\widehat{V}_n^\theta(X_n^j) + s_{n+1}(X_n^j, X_{n+1}^i)\} \right) \\ &\quad \Big/ \left(\sum_{j=1}^N W_n^j f_\theta(X_{n+1}^i|X_n^j) \right), \end{aligned}$$

$$(4.11) \qquad \widehat{\mathcal{S}}_n^\theta = \sum_{i=1}^{N} W_n^i \widehat{V}_n^\theta (X_n^i).$$

This approach requires $\mathcal{O}(N^2(n+1))$ operations to compute $\widehat{\mathcal{S}}_n^\theta$ at iteration n. A variation over this idea recently proposed in [75] and [88] consists of approximating $V_{n+1}^\theta(X_{n+1}^i)$ by sampling $X_n^{i,j} \sim \hat{p}_\theta(dx_n|y_{0:n}, X_{n+1}^i)$ for $j = 1, \ldots, K$ to obtain

$$(4.12) \qquad \begin{aligned} &\widehat{V}_{n+1}^\theta(X_{n+1}^i) \\ &= \frac{1}{K} \sum_{j=1}^{K} \{ \widehat{V}_n^\theta(X_n^{i,j}) + s_{n+1}(X_n^{i,j}, X_{n+1}^i) \}. \end{aligned}$$

When it is possible to sample from $\hat{p}_\theta(dx_n|y_{0:n}, X_{n+1}^i)$ in $\mathcal{O}(1)$ operations using rejection sampling, (4.12) provides a Monte Carlo approximation to (4.10) of overall complexity $\mathcal{O}(NK)$.

4.4 Convergence Results for Particle Smoothing

Empirically, for a fixed number of particles, these smoothing procedures perform significantly much better than the naive path space approach to smoothing (i.e., simply propagating forward the complete state trajectory within a particle filtering algorithm). Many theoretical results validating these empirical findings have been established under assumption (3.11) and additional regularity assumptions. The particle estimate of $\mathcal{S}_n^\theta$ based on the fixed-lag approximation (4.1) has an asymptotic variance in n/N with a nonvanishing (as $N \to \infty$) bias proportional to n and a constant decreasing exponentially fast with L [74]. In [25, 24, 28], it is shown that when (3.11) holds, there exists $0 < F_\theta$, $H_\theta < \infty$ such that the asymptotic bias and variance of the particle estimate of $\mathcal{S}_n^\theta$ computed using the forward–backward procedures satisfy

$$(4.13) \quad |\mathbb{E}(\widehat{\mathcal{S}}_n^\theta) - \mathcal{S}_n^\theta| \le F_\theta \frac{n}{N}, \qquad \mathbb{V}(\widehat{\mathcal{S}}_n^\theta) \le H_\theta \frac{n}{N}.$$

The bias for the path space and forward–backward estimators of $\mathcal{S}_n^\theta$ are actually equal [24]. Recently, it has also been established in [75] that, under similar regularity assumptions, the estimate obtained through (4.12) also admits an asymptotic variance in n/N whenever $K \ge 2$.

5. MAXIMUM LIKELIHOOD PARAMETER ESTIMATION

We describe in this section how the particle filtering and smoothing techniques introduced in Sections 3 and 4 can be used to implement maximum likelihood parameter estimation techniques.

5.1 Off-Line Methods

We recall that $\ell_T(\theta)$ denote the log-likelihood function associated to data $y_{0:T}$ introduced in Section 2. So as to maximize $\ell_T(\theta)$, one can rely on standard nonlinear optimization methods, for example, using quasi-Newton or gradient-ascent techniques. We will limit ourselves to these approaches even if they are sensitive to initialization and might get trapped in a local maximum.

5.1.1 *Likelihood function evaluation.* We have seen in Section 3 that $\ell_T(\theta)$ can be approximated using particle methods, for any fixed $\theta \in \Theta$. One may wish then to treat ML estimation as an optimization problem using Monte Carlo evaluations of $\ell_T(\theta)$. When optimizing a function calculated with a Monte Carlo error, a popular strategy is to make the evaluated function continuous by using common random numbers over different evaluations to ease the optimization. Unfortunately, this strategy is not helpful in the particle context. Indeed, in the resampling stage, particles $\{\overline{X}_n^i\}_{i=1}^N$ are resampled according to the distribution $\sum_{i=1}^N \overline{W}_{n+1}^i \delta_{X_n^i}(dx_n)$ which admits a piecewise constant and hence discontinuous cumulative distribution function (c.d.f.). A small change in θ will cause a small change in the importance weights $\{\overline{W}_{n+1}^i\}_{i=1}^N$ and this will potentially generate a different set of resampled particles. As a result, the log-likelihood function estimate will not be continuous in θ even if $\ell_T(\theta)$ is continuous.

To bypass this problem, an importance sampling method was introduced in [49], but it has computational complexity $\mathcal{O}(N^2(T+1))$ and only provides low variance estimates in the neighborhood of a suitably preselected parameter value. In the restricted scenario where $\mathcal{X} \subseteq \mathbb{R}$, an elegant solution to the discontinuity problem was proposed in [72]. The method uses common random numbers and introduces a "continuous" version of the resampling step by finding a permutation σ such that $X_n^{\sigma(1)} \le X_n^{\sigma(2)} \le \cdots \le X_n^{\sigma(N)}$ and defining a piecewise linear approximation of the resulting c.d.f. from which particles are resampled, that is,

$$F_n(x) = \left(\sum_{i=1}^{k-1} \overline{W}_{n+1}^{\sigma(i)} \right) + \overline{W}_{n+1}^{\sigma(k)} \frac{x - X_n^{\sigma(k-1)}}{X_n^{\sigma(k)} - X_n^{\sigma(k-1)}},$$

$$X_n^{\sigma(k-1)} \le x \le X_n^{\sigma(k)}.$$

This method requires $\mathcal{O}(N(T+1)\log N)$ operations due to the sorting of the particles, but the resulting

continuous estimate of $\ell_T(\theta)$ can be maximized using standard optimization techniques. Extensions to the multivariate case where $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ (with $n_x > 1$) have been proposed in [59] and [22]. However, the scheme [59] does not guarantee continuity of the likelihood function estimate and only provides log-likelihood estimates which are positively correlated for neighboring values in the parameter space, whereas the scheme in [22] has $\mathcal{O}(N^2)$ computational complexity and relies on a nonstandard particle filtering scheme.

When θ is high dimensional, the optimization over the parameter space may be made more efficient if provided with estimates of the gradient. This is exploited by the algorithms described in the forthcoming sections.

5.1.2 *Gradient ascent.* The log-likelihood $\ell_T(\theta)$ may be maximized with the following steepest ascent algorithm: at iteration $k + 1$

$$(5.1) \qquad \theta_{k+1} = \theta_k + \gamma_{k+1} \nabla_\theta \ell_T(\theta)|_{\theta=\theta_k},$$

where $\nabla_\theta \ell_T(\theta)|_{\theta=\theta_k}$ is the gradient of $\ell_T(\theta)$ w.r.t. θ evaluated at $\theta = \theta_k$ and $\{\gamma_k\}$ is a sequence of positive real numbers, called the step-size sequence. Typically, γ_k is determined adaptively at iteration k using a line search or the popular Barzilai–Borwein alternative. Both schemes guarantee convergence to a local maximum under weak regularity assumptions; see [95] for a survey.

The *score* vector $\nabla_\theta \ell_T(\theta)$ can be computed by using Fisher's identity given in (2.4). Given (2.2), it is easy to check that the score is of the form (3.14). An alternative to Fisher's identity to compute the score is presented in [20], but this also requires computing an expectation of the form (3.14).

These score estimation methods are not applicable in complex scenarios where it is possible to sample from $f_\theta(x'|x)$, but the analytical expression of this transition kernel is unavailable [51]. For those models, a naive approach is to use a finite difference estimate of the gradient; however, this might generate too high a variance estimate. An interesting alternative presented in [50], under the name of iterated filtering, consists of deriving an approximation of $\nabla_\theta \ell_T(\theta)|_{\theta=\theta_k}$ based on the posterior moments $\{\mathbb{E}(\vartheta_n|y_{0:n}), \mathbb{V}(\vartheta_n|y_{0:n})\}_{n=0}^T$ of an artificial state-space model with latent Markov process $\{Z_n = (X_n, \vartheta_n)\}_{n=0}^T$,

$$(5.2) \quad \vartheta_{n+1} = \vartheta_n + \varepsilon_{n+1}, \quad X_{n+1} \sim f_{\vartheta_{n+1}}(\cdot|x_n),$$

and observed process $Y_{n+1} \sim g_{\vartheta_{n+1}}(\cdot|x_{n+1})$. Here $\{\varepsilon_n\}_{n \geq 1}$ is a zero-mean white noise sequence with variance $\sigma^2 \Sigma$, $\mathbb{E}(\vartheta_{n+1}|\vartheta_n) = \vartheta_n$, $\mathbb{E}(\vartheta_0) = \theta_k$, $\mathbb{V}(\vartheta_0) =$

$\tau^2 \Sigma$. It is shown in [50] that this approximation improves as $\sigma^2, \tau^2 \to 0$ and $\sigma^2/\tau^2 \to 0$. Clearly, as the variance σ^2 of the artificial dynamic noise $\{\varepsilon_n\}$ on the θ-component decreases, it will be necessary to use more particles to approximate $\nabla_\theta \ell_T(\theta)|_{\theta=\theta_k}$ as the mixing properties of the artificial dynamic model deteriorates.

5.1.3 *Expectation–Maximization.* Gradient ascent algorithms can be numerically unstable as they require to scale carefully the components of the score vector. The Expectation Maximization (EM) algorithm is a very popular alternative procedure for maximizing $\ell_T(\theta)$ [27]. At iteration $k + 1$, we set

$$(5.3) \qquad \theta_{k+1} = \arg\max_\theta Q(\theta_k, \theta),$$

where

$$(5.4) \quad \begin{aligned} Q(\theta_k, \theta) = \int \log p_\theta(x_{0:T}, y_{0:T}) \\ \cdot p_{\theta_k}(x_{0:T}|y_{0:T}) \, dx_{0:T}. \end{aligned}$$

The sequence $\{\ell_T(\theta_k)\}_{k \geq 0}$ generated by this algorithm is nondecreasing. The EM is usually favored by practitioners whenever it is applicable, as it is numerically more stable than gradient techniques.

In terms of implementation, the EM consists of computing a n_s-dimensional summary statistic of the form (3.14) when $p_\theta(x_{0:T}, y_{0:T})$ belongs to the exponential family, and the maximizing argument of $Q(\theta_k, \theta)$ can be characterized explicitly through a suitable function $\Lambda : \mathbb{R}^{n_s} \to \Theta$, that is,

$$(5.5) \qquad \theta_{k+1} = \Lambda(T^{-1} \mathcal{S}_T^{\theta_k}).$$

5.1.4 *Discussion of particle implementations.* The path space approximation (3.7) can be used to approximate the score (2.4) and the summary statistics of the EM algorithm at the computational cost of $\mathcal{O}(N(T + 1))$; see [11], Section 8.3, and [74, 81]. Experimentally, the variance of the associated estimates increases typically quadratically with T [81]. To obtain estimates whose variance increases only typically linearly with T with similar computational cost, one can use the fixed-lag approximation presented in Section 4.1 or a more recent alternative where the path space method is used, but the additive functional of interest, which is a sum of terms over $n = 0, \ldots, T$, is approximated by a sum of similar terms which are now exponentially weighted w.r.t. n [73]. These methods introduce a nonvanishing asymptotic bias difficult to quantify but appear to perform well in practice.

To improve over the path space method without introducing any such asymptotic bias, the FFBSm and forward smoothing discussed in Sections 4.2 and 4.3 as well as the generalized two-filter smoother have been used [82, 25, 24, 81, 6]. Experimentally, the variance of the associated estimates increases typically linearly with T [81] in agreement with the theoretical results in [25, 24, 28]. However, the computational complexity of these techniques is $\mathcal{O}(N^2(T+1))$. For a fixed computational complexity of order $\mathcal{O}(N^2(T+1))$, an informal comparison of the performance of the path space estimate using N^2 particles and the forward–backward estimate using N particles suggest that both estimates admit a Mean Square Error (MSE) of order $\mathcal{O}(N^{-2}(T+1))$, but the MSE of the path space estimate is variance dominated, whereas the forward–backward estimates are bias dominated. This can be understood by decomposing the MSE as the sum of the squared bias and the variance and then substituting appropriately for N^2 particles in (3.15) for the path space method and for N particles in (4.13) for the forward–backward estimates. We confirm experimentally this fact in Section 7.1.

These experimental results suggest that these particle smoothing estimates might thus be of limited interest compared to the path based estimates for ML parameter inference when accounting for computational complexity. However, this comparison ignores that the $\mathcal{O}(N^2)$ computational complexity of these particle smoothing estimates can be reduced to $\mathcal{O}(N)$ by sampling approximately from $p_\theta(x_{0:T}|y_{0:T})$ with the FFBSa procedure in Section 4.2 or by using fast computational methods [57]. Related $\mathcal{O}(N)$ approaches have been developed for generalized two-filter smoothing [7, 38]. When applicable, these fast computational methods should be favored.

5.2 On-Line Methods

For a long observation sequence the computation of the gradient of $\ell_T(\theta)$ can be prohibitive, and moreover, we might have real-time constraints. An alternative would be a recursive procedure in which the data is run through once sequentially. If θ_n is the estimate of the model parameter after the first n observations, a recursive method would update the estimate to θ_{n+1} after receiving the new data y_n. Several on-line variants of the ML procedures described earlier are now presented. For these methods to be justified, it is crucial for the observation process to be ergodic for the limiting averaged likelihood function $\ell_T(\theta)/T$ to have a well-defined limit $\ell(\theta)$ as $T \to +\infty$.

5.2.1 *On-line gradient ascent.* An alternative to gradient ascent is the following parameter update scheme at time $n \geq 0$:

$$(5.6) \quad \theta_{n+1} = \theta_n + \gamma_{n+1} \nabla \log p_\theta(y_n|y_{0:n-1})|_{\theta=\theta_n},$$

where the positive nonincreasing step-size sequence $\{\gamma_n\}_{n \geq 1}$ satisfies $\sum_n \gamma_n = \infty$ and $\sum_n \gamma_n^2 < \infty$ [5, 64], for example, $\gamma_n = n^{-\alpha}$ for $0.5 < \alpha \leq 1$. Upon receiving y_n, the parameter estimate is updated in the direction of ascent of the conditional density of this new observation. In other words, one recognizes in (5.6) the update of the gradient ascent algorithm (5.1), except that the partial (up to time n) likelihood is used. The algorithm in the present form is, however, not suitable for on-line implementation, because evaluating the gradient of $\log p_\theta(y_n|y_{0:n-1})$ at the current parameter estimate requires computing the filter from time 0 to time n using the current parameter value θ_n.

An algorithm bypassing this problem has been proposed in the literature for a finite state-space latent process in [64]. It relies on the following update scheme:

$$(5.7) \quad \theta_{n+1} = \theta_n + \gamma_{n+1} \nabla \log p_{\theta_{0:n}}(y_n|y_{0:n-1}),$$

where $\nabla \log p_{\theta_{0:n}}(y_n|y_{0:n-1})$ is defined as

$$(5.8) \quad \begin{aligned} &\nabla \log p_{\theta_{0:n}}(y_n|y_{0:n-1}) \\ &= \nabla \log p_{\theta_{0:n}}(y_{0:n}) - \nabla \log p_{\theta_{0:n-1}}(y_{0:n-1}), \end{aligned}$$

with the notation $\nabla \log p_{\theta_{0:n}}(y_{0:n})$ corresponding to a "time-varying" score which is computed with a filter using the parameter θ_p at time p. The update rule (5.7) can be thought of as an approximation to the update rule (5.6). If we use Fisher's identity to compute this "time-varying" score, then we have for $1 \leq p \leq n$,

$$(5.9) \quad \begin{aligned} s_p(x_{p-1:p}) &= \nabla \log f_\theta(x_p|x_{p-1})|_{\theta=\theta_p} \\ &\quad + \nabla \log g_\theta(y_p|x_p)|_{\theta=\theta_p}. \end{aligned}$$

The asymptotic properties of the recursion (5.7) (i.e., the behavior of θ_n in the limit as n goes to infinity) has been studied in [64] for a finite state-space HMM. It is shown that under regularity conditions this algorithm converges toward a local maximum of the average log-likelihood $\ell(\theta)$, $\ell(\theta)$ being maximized at the "true" parameter value under identifiability assumptions. Similar results hold for the recursion (5.6).

5.2.2 *On-line Expectation–Maximization.* It is also possible to propose an on-line version of the EM algorithm. This was originally proposed for finite state-space and linear Gaussian models in [35, 42]; see [9] for a detailed presentation in the finite state-space case.

Assume that $p_\theta(x_{0:n}, y_{0:n})$ is in the exponential family. In the on-line implementation of EM, running averages of the sufficient statistics $n^{-1}\mathcal{S}_n^\theta$ are computed [8, 35]. Let $\{\theta_p\}_{0 \le p \le n}$ be the sequence of parameter estimates of the on-line EM algorithm computed sequentially based on $y_{0:n-1}$. When y_n is received, we compute

$$
\begin{aligned}
\mathcal{S}_{\theta_{0:n}} = {} & \gamma_{n+1} \int s_n(x_{n-1:n}) \\
& \cdot p_{\theta_{0:n}}(x_{n-1}, x_n | y_{0:n}) \, dx_{n-1:n} \\
& + (1 - \gamma_{n+1}) \sum_{k=0}^{n} \left(\prod_{i=k+2}^{n} (1 - \gamma_i) \right) \gamma_{k+1} \\
& \cdot \int s_k(x_{k-1:k}) p_{\theta_{0:k}}(x_{k-1:k} | y_{0:k}) \, dx_{k-1:k},
\end{aligned}
$$
(5.10)

where $\{\gamma_n\}_{n \ge 1}$ needs to satisfy $\sum_n \gamma_n = \infty$ and $\sum_n \gamma_n^2 < \infty$. Then the standard maximization step (5.5) is used as in the batch version

$$
\theta_{n+1} = \Lambda(\mathcal{S}_{\theta_{0:n}}).
\tag{5.11}
$$

The recursive calculation of $\mathcal{S}_{\theta_{0:n}}$ is achieved by setting $V_{\theta_0} = 0$, then computing

$$
\begin{aligned}
V_{\theta_{0:n}}(x_n) = \int \big\{ & \gamma_{n+1} s_n(x_{n-1}, x_n) \\
& + (1 - \gamma_{n+1}) V_{\theta_{0:n-1}}(x_{n-1}) \big\} \\
& \cdot p_{\theta_{0:n}}(x_{n-1} | y_{0:n-1}, x_n) \, dx_{n-1}
\end{aligned}
$$
(5.12)

and, finally,

$$
\mathcal{S}_{\theta_{0:n}} = \int V_{\theta_{0:n}}(x_n) p_{\theta_{0:n}}(x_n | y_{0:n}) \, dx_n.
\tag{5.13}
$$

Again, the subscript $\theta_{0:n}$ on $p_{\theta_{0:n}}(x_{0:n} | y_{0:n})$ indicates that the posterior density is being computed sequentially using the parameter θ_p at time $p \le n$. The filtering density then is advanced from time $n - 1$ to time n by using $f_{\theta_n}(x_n | x_{n-1})$, $g_{\theta_n}(y_n | x_n)$ and $p_{\theta_n}(y_n | y_{0:n})$ in the fraction of the r.h.s. of (3.1). Whereas the convergence of the EM algorithm toward a local maximum of the average log-likelihood $\ell(\theta)$ has been established for i.i.d. data [10], its convergence for state-space models remains an open problem despite empirical evidence it does [8, 9, 24]. This has motivated the development of modified versions of the on-line EM algorithm for which convergence results are easier to establish [4, 62]. However, the on-line EM presented here usually performs empirically better [63].

5.2.3 *Discussion of particle implementations.* Both the on-line gradient and EM procedures require approximating terms (5.8) and (5.10) of the form (3.14), except that the expectation is now w.r.t. the posterior density $p_{\theta_{0:n}}(x_{0:n} | y_{0:n})$ which is updated using the parameter θ_p at time $p \le n$. In this on-line framework, only the path space, fixed-lag smoothing and forward smoothing estimates are applicable; the fixed-lag approximation is applicable but introduces a nonvanishing bias. For the on-line EM algorithm, similarly to the batch case discussed in Section 5.1.4, the benefits of using the forward smoothing estimate [24] compared to the path space estimate [8] with N^2 particles are rather limited, as experimentally demonstrated in Section 7.1. However, for the on-line gradient ascent algorithm, the gradient term $\nabla \log p_{\theta_{0:n}}(y_n | y_{0:n-1})$ in (5.7) is a difference between two score-like vectors (5.8) and the behavior of its particle estimates differs significantly from its EM counterpart. Indeed, the variance of the particle path estimate of $\nabla \log p_{\theta_{0:n}}(y_n | y_{0:n-1})$ increases linearly with n, yielding an unreliable gradient ascent procedure, whereas the particle forward smoothing estimate has a variance uniformly bounded in time under appropriate regularity assumptions and yields a stable gradient ascent procedure [26]. Hence, the use of a procedure of computational complexity $\mathcal{O}(N^2)$ is clearly justified in this context. The very recent paper [88] reports that the computationally cheaper estimate (4.12) appears to exhibit similar properties whenever $K \ge 2$ and might prove an attractive alternative.

6. BAYESIAN PARAMETER ESTIMATION

In the Bayesian setting, we assign a suitable prior density $p(\theta)$ for θ and inference is based on the joint posterior density $p(x_{0:T}, \theta | y_{0:T})$ in the off-line case or the sequence of posterior densities $\{p(x_{0:n}, \theta | y_{0:n})\}_{n \ge 0}$ in the on-line case.

6.1 Off-Line Methods

6.1.1 *Particle Markov chain Monte Carlo methods.* Using MCMC is a standard approach to approximate $p(x_{0:T}, \theta | y_{0:T})$. Unfortunately, designing efficient MCMC sampling algorithms for nonlinear non-Gaussian state-space models is a difficult task: one-variable-at-a-time Gibbs sampling typically mixes very poorly for such models, whereas blocking strategies that have been proposed in the literature are typically very model-dependent; see, for instance, [52].

Particle MCMC are a class of MCMC techniques which rely on particle methods to build efficient high-dimensional proposal distributions in a generic manner [3]. We limit ourselves here to the presentation of the Particle Marginal Metropolis–Hastings (PMMH) sampler, which is an approximation of an ideal MMH sampler for sampling from $p(x_{0:T}, \theta | y_{0:T})$ which would utilize the following proposal density:

$$
\begin{aligned}
(6.1) \quad & q\big((x'_{0:T}, \theta') | (x_{0:T}, \theta)\big) \\
& = q(\theta' | \theta) p_{\theta'}(x'_{0:T} | y_{0:T}),
\end{aligned}
$$

where $q(\theta' | \theta)$ is a proposal density to obtain a candidate θ' when we are at location θ. The acceptance probability of this sampler is

$$
(6.2) \quad 1 \wedge \frac{p_{\theta'}(y_{0:T}) p(\theta') q(\theta | \theta')}{p_{\theta}(y_{0:T}) p(\theta) q(\theta' | \theta)}.
$$

Unfortunately, this ideal algorithm cannot be implemented, as we cannot sample exactly from $p_{\theta'}(x_{0:T} | y_{0:T})$ and we cannot compute the likelihood terms $p_{\theta}(y_{0:T})$ and $p_{\theta'}(y_{0:T})$ appearing in the acceptance probability.

The PMMH sampler is an approximation of this ideal MMH sampler which relies on the particle approximations of these unknown terms. Given θ and a particle approximation $\hat{p}_{\theta}(y_{0:T})$ of $p_{\theta}(y_{0:T})$, we sample $\theta' \sim q(\theta' | \theta)$, then run a particle filter to obtain approximations $\hat{p}_{\theta'}(dx_{0:T} | y_{0:T})$ and $\hat{p}_{\theta'}(y_{0:T})$ of $p_{\theta'}(dx_{0:T} | y_{0:T})$ and $p_{\theta'}(y_{0:T})$. We then sample $X'_{0:T} \sim \hat{p}_{\theta'}(dx_{0:T} | y_{0:T})$, that is, we choose randomly one of N particles generated by the particle filter, with probability W^i_T for particle i, and accept $(\theta', X'_{0:T})$ [and $\hat{p}_{\theta'}(y_{0:T})$] with probability

$$
(6.3) \quad 1 \wedge \frac{\hat{p}_{\theta'}(y_{0:T}) p(\theta') q(\theta | \theta')}{\hat{p}_{\theta}(y_{0:T}) p(\theta) q(\theta' | \theta)}.
$$

The acceptance probability (6.3) is a simple approximation of the "ideal" acceptance probability (6.2).

This algorithm was first proposed as a heuristic to sample from $p(\theta | y_{0:T})$ in [39]. Its remarkable feature established in [3] is that it does admit $p(x_{0:T}, \theta | y_{0:T})$ as invariant distribution whatever the number of particles N used in the particle approximation [3]. However, the choice of N has an impact on the performance of the algorithm. Using large values of N usually results in PMMH averages with variances lower than the corresponding averages using fewer samples, but the computational cost of constructing $\hat{p}_{\theta}(y_{0:T})$ increases with N. A simplified analysis of this algorithm suggests that N should be selected such that the standard deviation of the logarithm of the particle likelihood estimate should be around 0.9 if the ideal MMH sampler was using the perfect proposal $q(\theta' | \theta) = p(\theta' | y_{0:n})$ [79] and around 1.8 if one uses an isotropic normal random walk proposal for a target that is a product of d i.i.d. components with $d \to \infty$ [83]. For general proposal and target densities, a recent theoretical analysis and empirical results suggest that this standard deviation should be selected around 1.2–1.3 [33]. As the variance of this estimate typically increases linearly with T, this means that the computational complexity is of order $\mathcal{O}(T^2)$ by iteration.

A particle version of the Gibbs sampler is also available [3] which mimics the two-component Gibbs sampler sampling iteratively from $p(\theta | x_{0:T}, y_{0:T})$ and $p_{\theta}(x_{0:T} | y_{0:T})$. These algorithms rely on a nonstandard version of the particle filter where $N - 1$ particles are generated conditional upon a "fixed" particle. Recent improvements over this particle Gibbs sampler introduce mechanisms to rejuvenate the fixed particle, using forward or backward sampling procedures [89, 66, 91]. These methods perform empirically extremely well, but, contrary to the PMMH, it is still unclear how one should scale N with T.

6.2 On-Line Methods

In this context, we are interested in approximating on-line the sequence of posterior densities $\{p(x_{0:n}, \theta | y_{0:n})\}_{n \geq 0}$. We emphasize that, contrary to the online ML parameter estimation procedures, none of the methods presented in this section bypass the particle degeneracy problem. This should come as no surprise. As discussed in Section 3.2.2, even for a *fixed* θ, the particle estimate of $p_{\theta}(y_{0:n})$ has a relative variance that increases linearly with n under favorable mixing assumptions. The methods in this section attempt to approximate $p(\theta | y_{0:n}) \propto p_{\theta}(y_{0:n}) p(\theta)$. This is a harder problem, as it implicitly requires having to approximate $p_{\theta^i}(y_{0:n})$ for all the particles $\{\theta^i\}$ approximating $p(\theta | y_{0:n})$.

6.2.1 *Augmenting the state with the parameter.* At first sight, it seems that estimating the sequence of posterior densities $\{p(x_{0:n}, \theta | y_{0:n})\}_{n \geq 0}$ can be easily achieved using standard particle methods by merely introducing the extended state $Z_n = (X_n, \theta_n)$, with initial density $p(\theta_0) \mu_{\theta_0}(x_0)$ and transition density $f_{\theta_n}(x_n | x_{n-1}) \delta_{\theta_{n-1}}(\theta_n)$, that is, $\theta_n = \theta_{n-1}$. However, this extended process Z_n clearly does not possess any *forgetting* property (as discussed in Section 3), so the algorithm is bound to degenerate. Specifically, the parameter space is explored only in the initial step of the

algorithm. Then, each successive resampling step reduces the diversity of the sample of θ values; after a certain time n, the approximation $\hat{p}(d\theta|y_{0:n})$ contains a single unique value for θ. This is clearly a poor approach. Even in the much simpler case when there is no latent variable $X_{0:n}$, it is shown in [17], Theorem 4, that the asymptotic variance of the corresponding particle estimates diverges at least at a polynomial rate, which grows with the dimension of θ.

A pragmatic approach that has proven useful in some applications is to introduce artificial dynamics for the parameter θ [54],

$$(6.4) \qquad \theta_{n+1} = \theta_n + \varepsilon_{n+1},$$

where $\{\varepsilon_n\}_{n\geq 0}$ is an artificial dynamic noise with decreasing variance. Standard particle methods can now be applied to approximate $\{p(x_{0:n}, \theta_{0:n}|y_{0:n})\}_{n\geq 0}$. A related kernel density estimation method also appeared in [67], which proposes to use a kernel density estimate $p(\theta|y_{0:n})$ from which one samples from. As before, the static parameter is transformed to a slowly time-varying one, whose dynamics is related to the kernel bandwidth. To mitigate the artificial variance inflation, a shrinkage correction is introduced. An improved version of this method has been recently proposed in [41].

It is difficult to quantify how much bias is introduced in the resulting estimates by the introduction of this artificial dynamics. Additionally, these methods require a significant amount of tuning, for example, choosing the variance of the artificial dynamic noise or the kernel width. However, they can perform satisfactorily in practice [41, 67].

6.2.2 *Practical filtering.* The practical filtering approach proposed in [80] relies on the following fixed-lag approximation:

$$(6.5) \qquad p(x_{0:n-L}, \theta|y_{0:n-1}) \approx p(x_{0:n-L}, \theta|y_{0:n})$$

for L large enough; that is, observations coming after $n-1$ presumably bring little information on $x_{0:n-L}$. To sample approximately from $p(\theta|y_{0:n})$, one uses the following iterative process: at time n, several MCMC chains are run in parallel to sample from

$$p(x_{n-L+1:n}, \theta|y_{0:n}, X^i_{0:n-L})$$
$$= p(x_{n-L+1:n}, \theta|y_{n-L+1:n}, X^i_{n-L}),$$

where the X^i_{n-L} have been obtained at the previous iteration and are such that (approximately) $X^i_{n-L} \sim p(x_{n-L}|y_{0:n-1}) \approx p(x_{n-L}|y_{0:n})$. Then one collects

the first component X^i_{n-L+1} of the simulated sample $X^i_{n-L+1:n}$, increments the time index and runs several new MCMC chains in parallel to sample from $p(x_{n-L+2:n+1}, \theta|y_{n-L+2:n+1}, X^i_{n-L+1})$ and so on. The algorithm is started at time $L-1$, with MCMC chains that target $p(x_{0:L-1}|y_{0:L-1})$. Like all methods based on fixed-lag approximation, the choice of the lag L is difficult and this introduces a nonvanishing bias which is difficult to quantify. However, the method performs well on the examples presented in [80].

6.2.3 *Using MCMC steps within particle methods.* To avoid the introduction of an artificial dynamic model or of a fixed-lag approximation, an approach originally proposed independently in [36] and [44] consists of adding MCMC steps to re-introduce "diversity" among the particles. Assuming we use an auxiliary particle filter to approximate $\{p(x_{0:n}, \theta|y_{0:n})\}_{n\geq 0}$, then the particles $\{X^i_{0:n}, \theta^i_n\}$ obtained after the sampling step at time n are approximately distributed according to

$$\tilde{p}(x_{0:n}, \theta|y_{0:n})$$
$$\propto p(x_{0:n-1}, \theta|y_{0:n-1})q_\theta(x_n, y_n|x_{n-1}).$$

We have $\tilde{p}(x_{0:n}, \theta|y_{0:n}) = p(x_{0:n}, \theta|y_{0:n})$ if $q_\theta(x_n|y_n, x_{n-1}) = p_\theta(x_n|y_n, x_{n-1})$ and $q_\theta(y_n|x_{n-1}) = p_\theta(y_n|x_{n-1})$. To add diversity in this population of particles, we introduce an MCMC kernel $K_n(d(x'_{0:n}, \theta')|(x_{0:n}, \theta))$ with invariant density $\tilde{p}(x_{0:n}, \theta|y_{0:n})$ and replace, at the end of each iteration, the set of resampled particles, $(\overline{X}^i_{0:n}, \bar{\theta}^i_n)$ with N "mutated" particles $(\widetilde{X}^i_{0:n}, \tilde{\theta}^i_n)$ simulated from, for $i = 1, \ldots, N$,

$$\left(\widetilde{X}^i_{0:n}, \tilde{\theta}^i_n\right) \sim K_n\left(d(x_{0:n}, \theta)|(\overline{X}^i_{0:n}, \bar{\theta}^i_n)\right).$$

If we use the SISR algorithm, then we can alternatively use an MCMC step of invariant density $p(x_{0:n}, \theta|y_{0:n})$ after the resampling step at time n.

Contrary to standard applications of MCMC, the kernel does not have to be ergodic. Ensuring ergodicity would indeed require one to sample an increasing number of variables as n increases—this algorithm would have an increasing cost per iteration, which would prevents its use in on-line scenarios, but it can be an interesting alternative to standard MCMC and was suggested in [61]. In practice, one therefore sets $\widetilde{X}^i_{0:n-L} = X^i_{0:n-L}$ and only samples θ^i and $\widetilde{X}^i_{n-L+1:n}$, where L is a small integer; often $L = 0$ (only θ is updated). Note that the memory requirements for this method do not increase over time if $\tilde{p}_\theta(x_{0:n}, y_{0:n})$ is in the exponential family and thus

can be summarized by a set of fixed-dimensional sufficient statistics $s^n(x_{0:n}, y_{0:n})$. This type of method was first used to perform on-line Bayesian parameter estimation in a context where $\tilde{p}_\theta(x_{0:n}, y_{0:n})$ is in the exponential family [44, 36]. Similar strategies were adopted in [2] and [84]. In the particular scenario where $q_\theta(x_n|y_n, x_{n-1}) = p_\theta(x_n|y_n, x_{n-1})$ and $q_\theta(y_n|x_{n-1}) = p_\theta(y_n|x_{n-1})$, this method was mentioned in [2, 86] and is discussed at length in [70] who named it particle learning. Extensions of this strategy to parameter estimation in conditionally linear Gaussian models, where a part of the state is integrated out using Kalman techniques [15, 31], is proposed in [13].

As opposed to the methods relying on kernel or artificial dynamics, these MCMC-based approaches have the advantage of adding diversity to the particles approximating $p(\theta|y_{0:n})$ without perturbing the target distribution. Unfortunately, these algorithms rely implicitly on the particle approximation of the density $p(x_{0:n}|y_{0:n})$ even if algorithmically it is only necessary to store some fixed-dimensional sufficient statistics $\{s^n(X_{0:n}^i, y_{0:n})\}$. Hence, in this respect they suffer from the degeneracy problem. This was noticed as early as in [2]; see also the word of caution in the conclusion of [4, 36] and [18]. The practical implications are that one observes empirically that the resulting Monte Carlo estimates can display quite a lot of variability over multiple runs as demonstrated in Section 7.2. This should not come as a surprise, as the sequence of posterior distributions does not have exponential forgetting properties, hence, there is an accumulation of Monte Carlo errors over time.

6.2.4 *The SMC^2 algorithm.* The SMC^2 algorithm introduced simultaneously in [19] and [43] may be considered as the particle equivalent of Particle MCMC. It mimics an "ideal" particle algorithm proposed in [16] approximating sequentially $\{p(\theta|y_{0:n})\}_{n\geq 0}$ where N_θ particles (in the θ-space) are used to explore these distributions. The N_θ particles at time n are reweighted according to $p_\theta(y_{0:n+1})/p_\theta(y_{0:n})$ at time $n + 1$. As these likelihood terms are unknown, we substitute to them $\hat{p}_\theta(y_{0:n+1})/\hat{p}_\theta(y_{0:n})$ where $\hat{p}_\theta(y_{0:n})$ is a particle approximation of the partial likelihood $p_\theta(y_{0:n})$, obtained by a running a particle filter of N_x particles in the x-dimension, up to time n, for each of the N_θ θ-particles. When particle degeneracy (in the θ-dimension) reaches a certain threshold, θ-particles are refreshed through the succession of a resampling step, and an MCMC step, which in these particular settings takes the form of a PMCMC update.

The cost per iteration of this algorithm is not constant and, additionally, it is advised to increase N_x with n for the relative variance of $\hat{p}_\theta(y_{0:n})$ not to increase, therefore, it cannot be used in truly on-line scenarios. Yet there are practical situations where it may be useful to approximate jointly all the posteriors $p(\theta|y_{1:n})$, for $1 \leq n \leq T$, for instance, to assess the predictive power of the model.

7. EXPERIMENTAL RESULTS

We focus on illustrating numerically a few algorithms and the impact of the degeneracy problem on parameter inference. This last point is motivated by the fact that particle degeneracy seems to have been overlooked by many practitioners. In this way numerical results may provide valuable insights.

We will consider the following simple scalar linear Gaussian state space model:

$$(7.1) \quad X_n = \rho X_{n-1} + \tau W_n, \quad Y_n = X_n + \sigma V_n,$$

where V_n, W_n are independent zero-mean and unit-variance Gaussians and $\rho \in [-1, 1]$. The main reason for choosing this model is that Kalman recursions can be implemented to provide the exact values of the summary statistics $\mathcal{S}_n^\theta$ used for ML estimation through the EM algorithm and to compute the exact likelihood $p_\theta(y_{0:n})$. Hence, using a fine discretization of the low-dimensional parameter space, we can compute a very good approximation of the true posterior density $p(\theta|y_{0:n})$. In this model it is straightforward to present numerical evidence of some effects of degeneracy for parameter estimation and to show how it can be overcome by choosing an appropriate particle method.

7.1 Maximum Likelihood Methods

As ML methods require approximating smoothed additive functionals $\mathcal{S}_n^\theta$ of the form (3.14), we begin by investigating the empirical bias, variance and MSE of two standard particle estimates of $\mathcal{S}_n^\theta$, where we set $s_k(x_{k-1}, x_k) = x_{k-1}x_k$ for the model described in (7.1). The first estimate relies on the path space method with computational cost $\mathcal{O}(N)$ per time, which uses $\hat{p}_\theta(dx_{0:n}|y_{0:n})$ in (3.7) to approximate $\mathcal{S}_n^\theta$ as $\widehat{\mathcal{S}}_n^\theta$; see [11], Section 8.3, for more details. The second estimate relies on the forward implementation of FFBSm presented in Section 4.3 using (4.7)–(4.11); see [24]. Recall that this procedure has a computational cost that is $\mathcal{O}(N^2)$ per time for N particles and provides the same estimates as the standard forward–backward implementation of FFBSm. For the sake of brevity, we

will not consider the remaining smoothing methods of Section 4; for the fixed-lag and the exponentially weighted approximations we refer the reader to [74], respectively, [73] for numerical experiments.

We use a simulated data set of size 6×10^4 obtained using $\theta^* = (\rho^*, \tau^{2*}, \sigma^{2*}) = (0.8, 0.1, 1)$ and then generate 300 independent replications of each method in order to compute the empirical bias and variance of $\widehat{\mathcal{S}}_n^{\theta^*}$ when θ is fixed to θ^*. In order to make a comparison that takes into account the computational cost, we use N^2 particles for the $\mathcal{O}(N)$ method and N for the $\mathcal{O}(N^2)$ one. We look separately at the behavior of the bias of $\widehat{\mathcal{S}}_n^\theta$ and the variance and MSE of the rescaled estimates $\widehat{\mathcal{S}}_n^\theta / \sqrt{n}$. The results are presented in Figure 1 for $N = 50, 100, 200$.

For both methods the bias grows linearly with time, this growth being higher for the $\mathcal{O}(N^2)$ method. For

the variance of $\widehat{\mathcal{S}}_n^\theta / \sqrt{n}$, we observe a linear growth with time for the $\mathcal{O}(N)$ method with N^2 particles, whereas this variance appears roughly constant for the $\mathcal{O}(N^2)$ method. Finally, the MSE of $\widehat{\mathcal{S}}_n^\theta / \sqrt{n}$ grows for both methods linearly as expected. In this particular scenario, the constants of proportionality are such that the MSE is lower for the $\mathcal{O}(N)$ method than for the $\mathcal{O}(N^2)$ method. In general, we can expect that the $\mathcal{O}(N)$ method be superior in terms of the bias and the $\mathcal{O}(N^2)$ method superior in terms of the variance. These results are in agreement with the theoretical results in the literature [25, 24, 28], but additionally show that the lower bound on the variance growth of $\widehat{\mathcal{S}}_n^\theta$ for the $\mathcal{O}(N)$ method of [81] appears sharp.

We proceed to see how the bias and variance of the estimates of $\mathcal{S}_n^\theta$ affect the ML estimates, when the former are used within both an off-line and an on-line EM
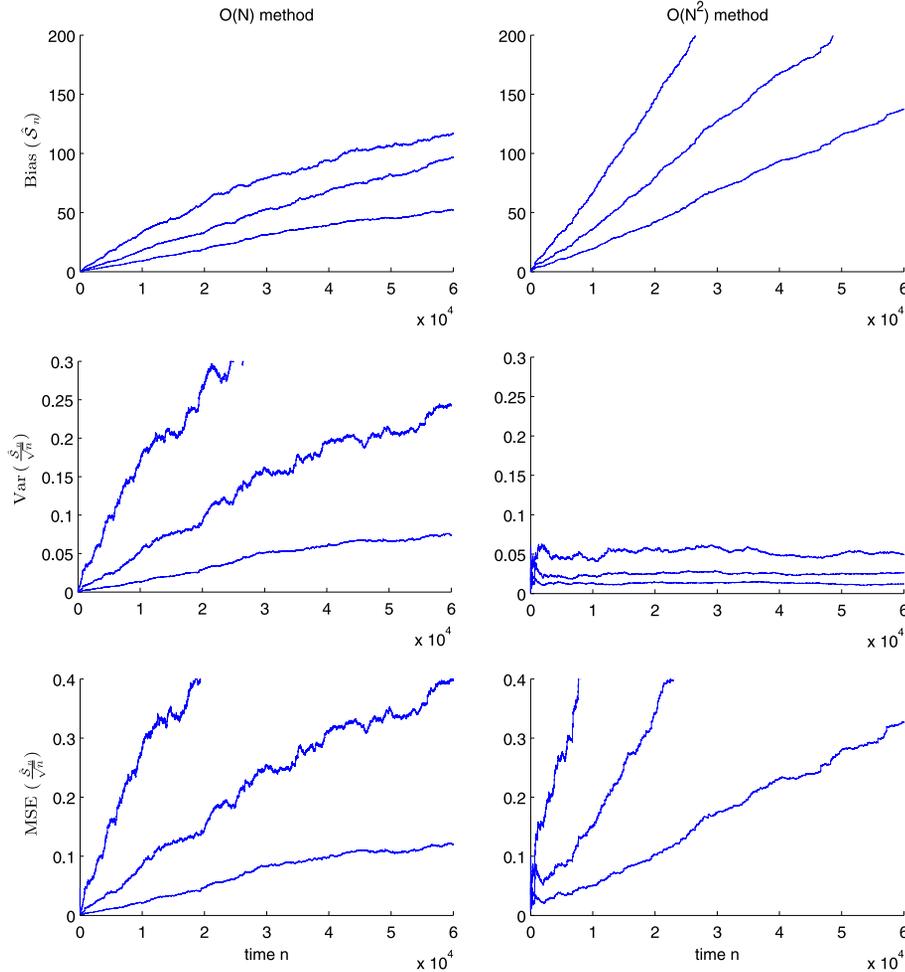

FIG. 1. *Estimating smoothed additive functionals*: *empirical bias of the estimate of $\mathcal{S}_n^\theta$ (top panel), empirical variance (middle panel) and MSE (bottom panel) for the estimate of $\mathcal{S}_n^\theta / \sqrt{n}$. Left column*: $\mathcal{O}(N)$ *method using $N^2 = 2500, 10{,}000, 40{,}000$ particles. Right column*: $\mathcal{O}(N^2)$ *method using $N = 50, 100, 200$ particles. In every subplot, the top line corresponds to using $N = 50$, the middle for $N = 100$ and the lower for $N = 200$.*
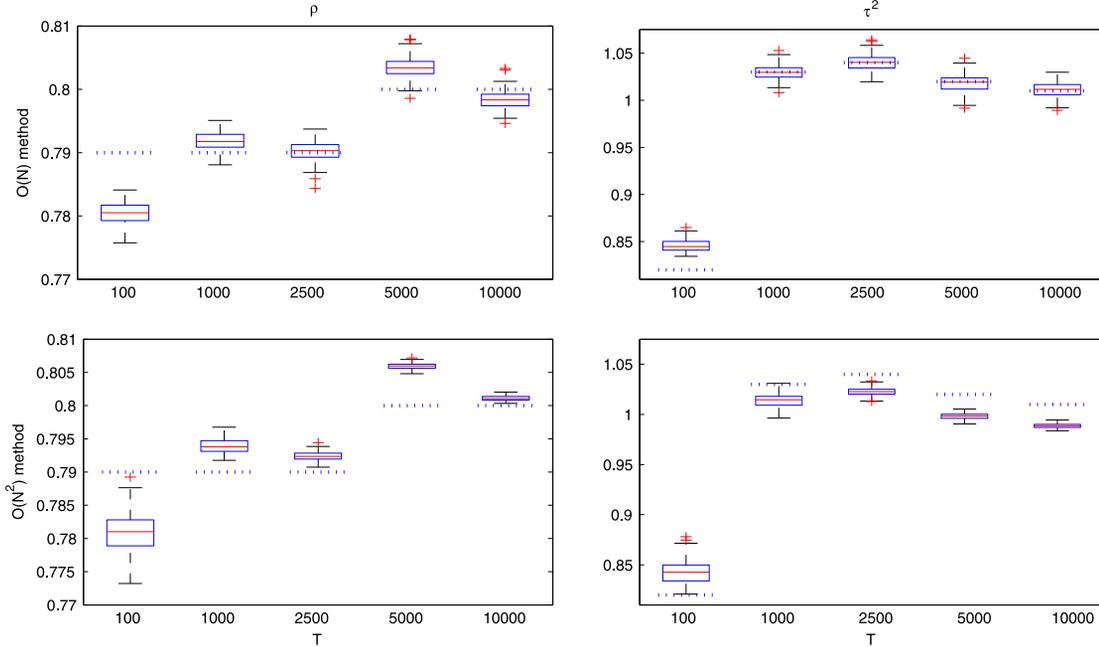
FIG. 2. *Off-line EM*: *boxplots of $\hat{\theta}_n$ for various T using* 25 *iterations of off-line EM and* 150 *realizations of the algorithms. Top panels*: $\mathcal{O}(N)$ *method using* $N = 150^2$ *particles. Bottom panels*: $\mathcal{O}(N^2)$ *with* $N = 150$. *The dotted horizontal lines are the ML estimate for each time T obtained using Kalman filtering on a grid.*

algorithm; see Figures 2 and 3, respectively. For the model in (7.1) the E-step corresponds to computing $\mathcal{S}_n^\theta$ where $s_k(x_{k-1}, x_k) = ((y_k - x_k)^2, x_{k-1}^2, x_{k-1}x_k, x_k^2)$ and the M-step update function is given by

$$\Lambda(z_1, z_2, z_3, z_4) = \left(\frac{z_3}{z_4}, z_4 - \frac{z_3^2}{z_2}, z_1 \right).$$

We compare the estimates of θ^* when the E-step is computed using the $\mathcal{O}(N)$ and the $\mathcal{O}(N^2)$ methods described in the previous section with 150^2 and 150 particles, respectively. A simulated data set for $\theta^* = (\rho^*, \tau^*, \sigma^*) = (0.8, 1, 0.2)$ will be used. In every case we will initialize the algorithm using $\theta_0 =$
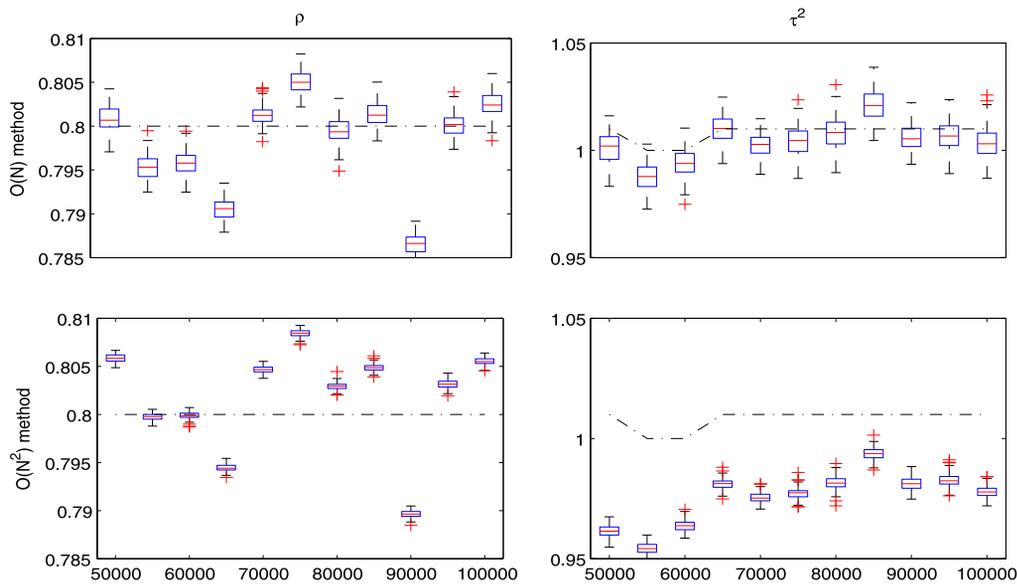
FIG. 3. *On-line EM*: *boxplots of $\hat{\theta}_n$ for $n \geq 5 \times 10^4$ using* 150 *realizations of the algorithms. We also plot the ML estimate at time n obtained using Kalman filtering on a grid (black).*

$(0.1, 0.1, 0.2)$ and assume σ^* is known. In Figures 2 and 3 we present the results obtained using 150 independent replications of the algorithm. For the off-line EM, we use 25 iterations for $T = 100, 1000, 2500, 5000, 10{,}000$. For the on-line EM, we use $T = 10^5$ with the step size set as $\gamma_n = n^{-0.8}$ and for the first 50 iterations no M-step update is performed. This "freezing" phase is required to allow for a reasonable estimation of the summary statistic; see [8, 9] for more details. Note that in Figure 3 we plot only the results after the algorithm has converged, that is, for $n \geq 5 \times 10^4$. In each case, both the $\mathcal{O}(N)$ and the $\mathcal{O}(N^2)$ methods yield fairly accurate results given the low number of particles used. However, we note, as observed previously in the literature, that the on-line EM as well as the on-line gradient ascent method requires a substantial number of observations, that is, over 10,000, before achieving convergence [8, 9, 24, 81]. For smaller data sets, these algorithms can also be used by going through the data, say, K times. Typically, this method is cheaper than iterating (5.1) or (5.4)–(5.5) K times the off-line algorithms and can yield comparable parameter estimates [94]. Experimentally, the properties of the estimates of $\mathcal{S}_n^\theta$ discussed earlier appear to translate into properties of the resulting parameter estimates: the $\mathcal{O}(N)$ method provides estimates with less bias but more variance than the $\mathcal{O}(N^2)$ method.

For more numerical examples regarding the remaining methods discussed in Section 5, we refer the reader to [50, 51] for iterated filtering, to [24, 25, 81] for comparisons of the $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$ methods for EM and gradient ascent, to [8] for the $\mathcal{O}(N)$ on-line EM, to [72] and [59], Chapter 10, for smooth likelihood function methods and to [11], Chapters 10–11, for a detailed exposition of off-line EM methods.

7.2 Bayesian Methods

We still consider the model in (7.1), but simplify it further by fixing either ρ or τ. This is done in order to keep the computations of the benchmarks that use Kalman computations on a grid relatively inexpensive. For those parameters that are not fixed, we shall use the following independent priors: a uniform on $[-1, 1]$ for ρ, and inverse gamma for τ^2, σ^2 with the shape and scale parameter pair being (a, b) and (c, d), respectively, with $a = b = c = d = 1$. In all the subsequent examples, we will initialize the algorithms by sampling θ from the prior.

We proceed to examine the particle algorithms with MCMC moves that we described in Section 6.2.3. We

focus on an efficient implementation of this idea discussed in [70] which can be put in practice for the simple model under consideration. We investigate the effect of the degeneracy problem in this context. The numerical results obtained in this section have been produced in Matlab (code available from the first author) and double-checked using the R program available on the personal web page of the first author of [70, 71].

We first focus on the estimate of the posterior of $\theta = (\tau^2, \sigma^2)$ given a long sequence of simulated observations with $\tau = \sigma = 1$. In this scenario, $p_\theta(x_{0:n}, y_{0:n})$ admits the following two-dimensional sufficient statistics, $s^n(x_{0:n}, y_{0:n}) = (\sum_{k=1}^n (x_k - x_{k-1})^2, \sum_{k=0}^n (y_k - x_k)^2)$, and θ can be updated using Gibbs steps. We use $T = 5 \times 10^4$ and $N = 5000$. We ran the algorithm over 100 independent runs over the same data set. We present the results only for τ^2 and omit the ones for σ^2, as these were very similar. The top left panel of Figure 4 shows the box plots for the estimates of the posterior mean, and the top right panel shows how the corresponding relative variance of the estimator for the posterior mean evolves with time. Here the relative variance is defined as the ratio of the empirical variance (over different independent runs) of the posterior mean estimates at time n over the true posterior variance at time n, which in this case is approximated using a Kalman filter on a fine grid. This quantity exhibits a steep increasing trend when $n \geq 15{,}000$ and confirms the aforementioned variability of the estimates of the posterior mean. In the bottom left panel of Figure 4 we plot the average (over different runs) of the estimators of the variance of $p(\tau^2 | y_{0:n})$. This average variance is also scaled/normalized by the actual posterior variance. The latter is again computed using Kalman filtering on a grid. This ratio between the average estimated variance of the posterior over the true one decreases with time n and it shows that the supports of the approximate posterior densities provided by this method cover, on average, only a small portion of the support of the true posterior. These experiments confirm that in this example the particle method with MCMC steps fails to adequately explore the space of θ. Although the box plots provide some false sense of security, the relative and scaled average variance clearly indicate that any posterior estimates obtained from a single run of particle method with MCMC steps should be used with caution. Furthermore, in the bottom right panel of Figure 4 we also investigate experimentally the empirical relative variance of the marginal likelihood estimates $\{\hat{p}(y_{0:n})\}_{n \geq 0}$. This relative variance
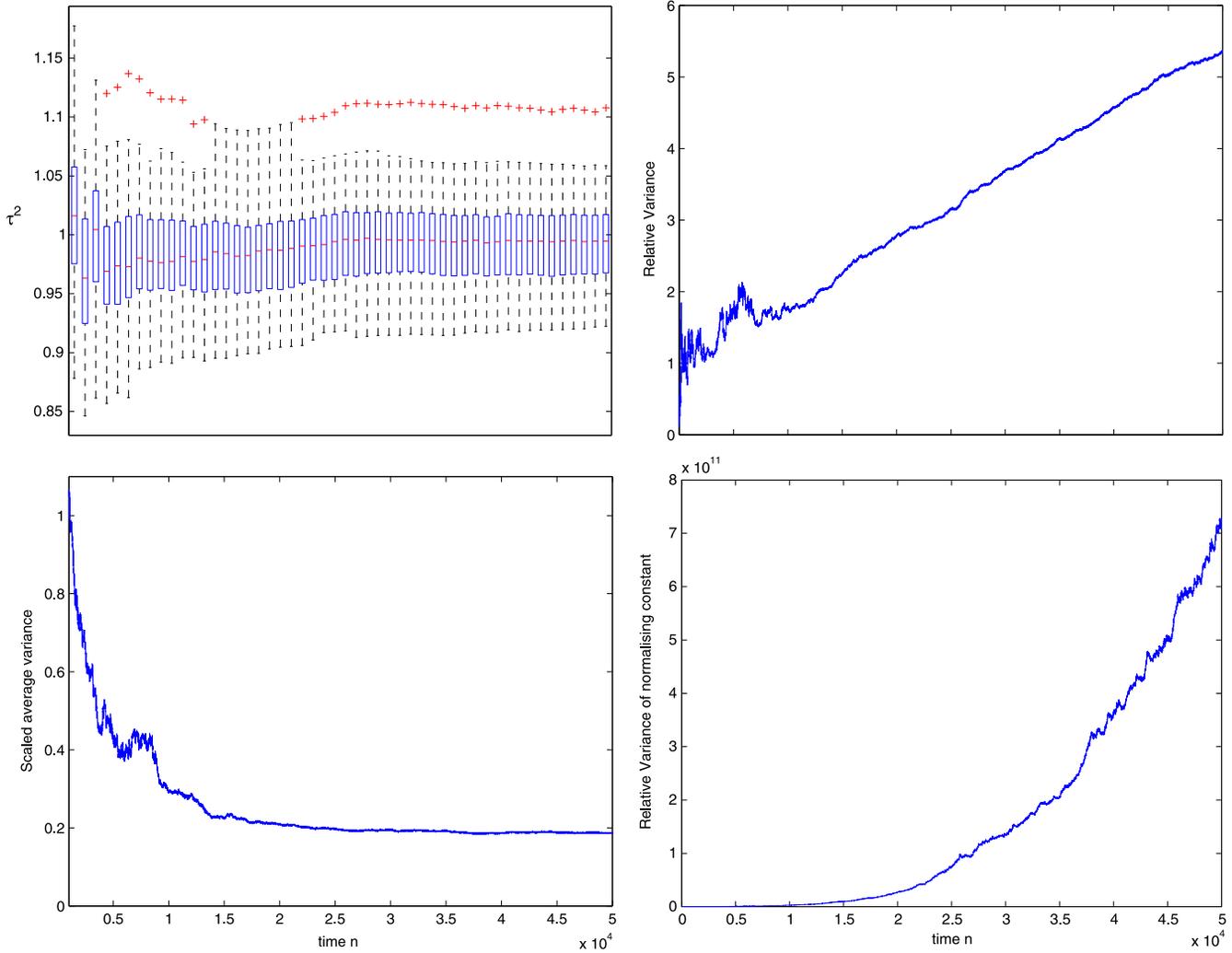
FIG. 4.   *Top left: box plots for estimates of posterior mean of τ^2 at $n = 1000, 2000, \ldots, 50{,}000$. Top right: relative variance, that is, empirical variance (over independent runs) for the estimator of the mean of $p(\tau^2|y_{0:n})$ using particle method with MCMC steps normalized with the true posterior variance computed using Kalman filtering on a grid. Bottom left: average (over independent runs) of the estimated variance of $p(\tau^2|y_{0:n})$ using particle method with MCMC normalized with the true posterior variance. Bottom right: relative variance of the $\{\hat{p}(y_{0:n})\}_{n\geq0}$; All plots are computed using $N = 5000$ and over 100 different independent runs.*

appears to increase quadratically with n for the particle method with MCMC moves instead of linearly as it does for state-space models with good mixing properties. This suggests that one should increase the number of particles quadratically with the time index to obtain an estimate of the marginal likelihood whose relative variance remains uniformly bounded with respect to the time index. Although we attribute this quadratic relative variance growth to the degeneracy problem, the estimate $\hat{p}(y_{0:n})$ is not the particle approximation of a smoothed additive functional, thus there is not yet any theoretical convergence result explaining rigorously this phenomenon.

One might argue that these particle methods with MCMC moves are meant to be used with larger N and/or shorter data sets T. We shall consider this time a slightly different example where $\tau = 0.1$ is known and we are interested in estimating the posterior of $\theta = (\rho, \sigma^2)$ given a sequence of observations obtained using $\rho = 0.5$ and $\sigma = 1$. In that case, the sufficient statistics are $s^n(x_{0:n}, y_{0:n}) = (\sum_{k=1}^{n} x_{k-1}x_k, \sum_{k=0}^{n-1} x_{k-1}^2, \sum_{k=0}^{n} (y_k - x_k)^2)$, and the parameters can be rejuvenated through a single Gibbs update. In addition, we let $T = 5000$ and use $N = 10^4$ particles. In Figure 5 we display the estimated marginal posteriors $p(\rho|y_{0:n})$ and $p(\sigma^2|y_{0:n})$ obtained from
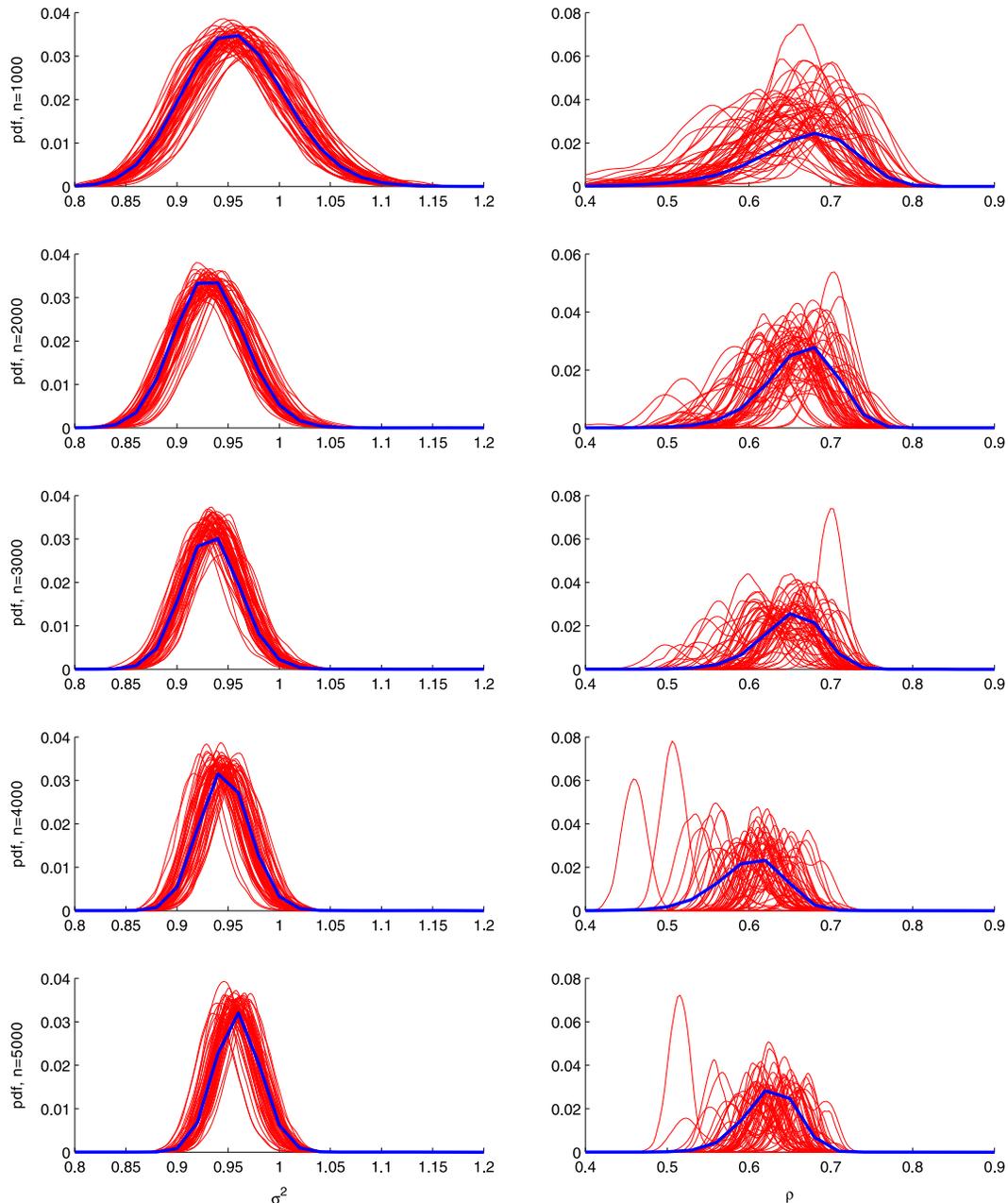
FIG. 5.   *Particle method with MCMC steps,* $\theta = (\rho, \sigma^2)$; *estimated marginal posterior densities for* $n = 10^3, 2 \times 10^3, \ldots, 5 \times 10^3$ *over* 50 *runs* (*red*) *versus ground truth* (*blue*).

50 independent replications of the particle method. On this simple problem, the estimated posteriors seem consistently rather inaccurate for ρ, whereas they perform better for σ^2 but with some nonnegligible variability over runs, which increases as T increases. Similar observations have been reported in [18] and remain unexplained: for some parameters this methodology appears to provide reasonable results despite the degeneracy problem and for others it provides very unreliable results.

We investigate further the performance of this method in this simple example by considering the same example for $T = 1000$, but now consider two larger numbers of particles, $N = 7.5 \times 10^4$ and $N = 6 \times 10^5$, over 50 different runs. Additionally, we compare the resulting estimates with estimates provided by the particle Gibbs sampler of [66] using the same computational cost, that is, $N = 50$ particles with 3000 and 24,000 iterations, respectively. The results are displayed in Figures 6 and 7. As expected, we improve
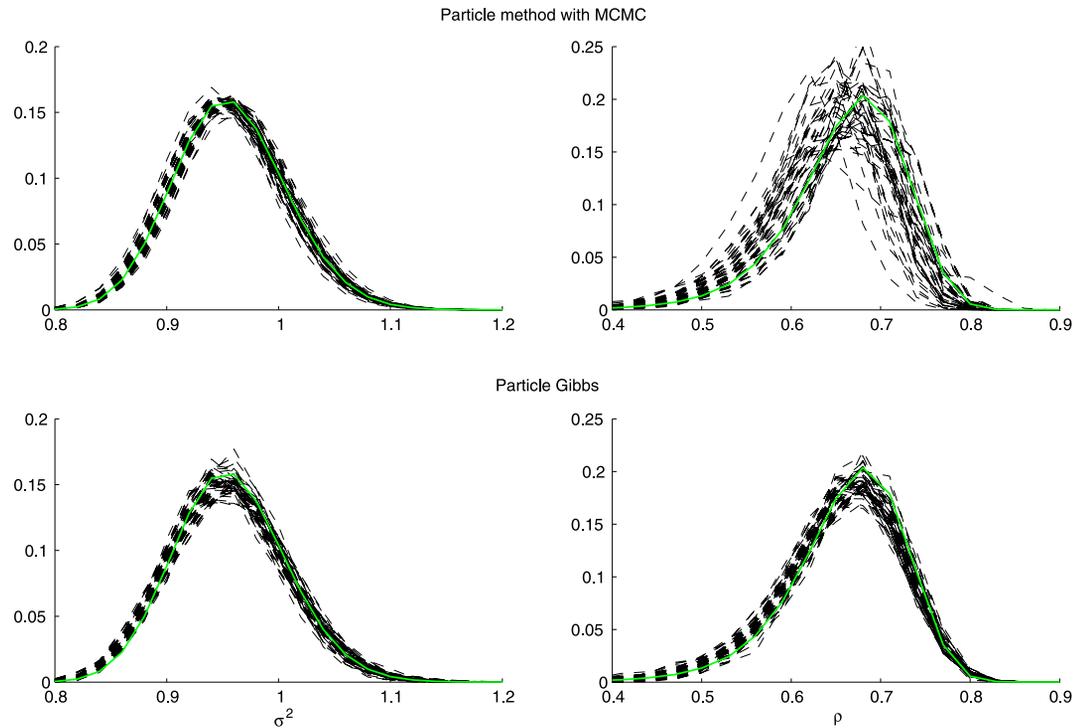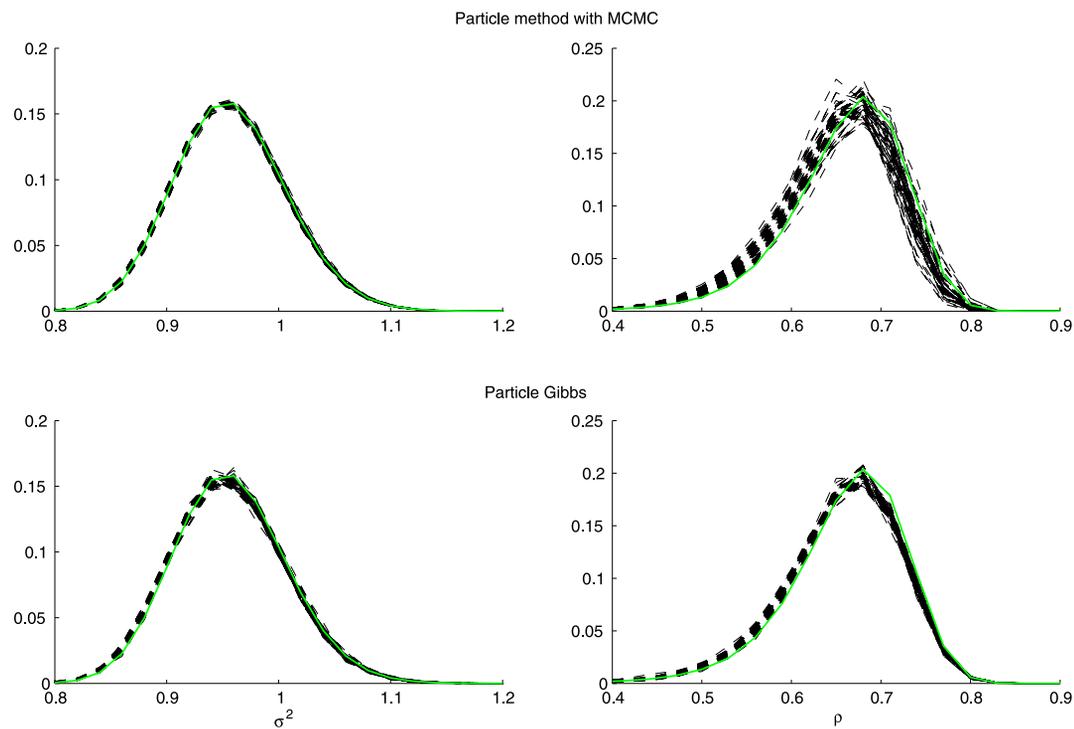
FIG. 6. *Estimated marginal posterior densities for* $\theta = (\rho, \sigma^2)$ *with* $T = 10^3$ *over* 50 *runs* (*black-dashed*) *versus ground truth* (*green*). *Top*: *particle method with MCMC steps*, $N = 7.5 \times 10^4$. *Bottom*: *particle Gibbs with* 3000 *iterations and* $N = 50$.

FIG. 7. *Estimated marginal posterior densities for* $\theta = (\rho, \sigma^2)$ *with* $T = 10^3$ *over* 50 *runs* (*black-dashed*) *versus ground truth* (*green*). *Top*: *particle method with MCMC steps*, $N = 6 \times 10^5$. *Bottom*: *particle Gibbs with* 24,000 *iterations and* $N = 50$.

the performance of the particle with MCMC moves when N increases for a fixed time horizon T. For a fixed computational complexity, the particle Gibbs sampler estimates appear to display less variability. For a higher-dimensional parameter θ and/or very vague priors, this comparison would be more favorable to the particle Gibbs sampler as illustrated in [3], pages 336–338.

8. CONCLUSION

Most particle methods proposed originally in the literature to perform inference about static parameters in general state-space models were computationally inefficient as they suffered from the degeneracy problem. Several approaches have been proposed to deal with this problem by either adding an artificial dynamic on the static parameter [40, 54, 67] or introducing a fixed-lag approximation [56, 74, 80]. These methods can work very well in practice, but it remains unfortunately difficult/impossible to quantify the bias introduced in most realistic applications. Various asymptotically bias-free methods with good statistical properties and a reasonable computational cost have recently appeared in the literature.

To perform batch ML estimation, the forward filter backward sampler/smoother and generalized two-filter procedures are recommended whenever the $\mathcal{O}(N^2 T)$ computational complexity per iteration of their direct implementations can be lowered to $\mathcal{O}(NT)$ using, for example, the methods described in [7, 28, 38, 57]. Otherwise, besides a lowering of memory requirements, not much can be gained from these techniques compared to simply using a standard particle filter with N^2 particles. In an on-line ML context, the situation is markedly different. Whereas for the on-line EM algorithm, the forward smoothing approach in [24, 81] of complexity $\mathcal{O}(N^2)$ per time step will be similarly of limited interest compared to a standard particle filter using N^2 particles; it is crucial to use this approach when performing on-line gradient ascent as demonstrated empirically and established theoretically in [26]. In on-line scenarios where one can admit a random computational complexity at each time step, the method presented in [75] is an interesting alternative when it is applicable. Empirically, these on-line ML methods converge rather slowly and will be primarily useful for large data sets.

In a Bayesian framework, batch inference can be conducted using particle MCMC methods [3, 66]. However, these methods are computationally expensive as, for example, an efficient implementation of the PMMH has a computational complexity of order $\mathcal{O}(T^2)$ per iteration [33]. On-line Bayesian inference remains a challenging open problem as all methods currently available, including particle methods with MCMC moves [13, 36, 84], suffer from the degeneracy problem. These methods should not be ruled out, but should be used cautiously, as they can provide unreliable results even in simple scenarios as demonstrated in our experiments.

Very recent papers in this dynamic research area have proposed to combine individual parameter estimation techniques so as to design more efficient inference algorithms. For example, [21] suggests to use the score estimation techniques developed for ML parameter estimation to design better proposal distributions for the PMMH algorithm, whereas [37] demonstrates that particle methods with MCMC moves might be fruitfully used in batch scenarios when plugged into a particle MCMC scheme.

ACKNOWLEDGMENTS

REFERENCES

[1] ALSPACH, D. and SORENSON, H. (1972). Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans. Automat. Control* **17** 439–448.

[2] ANDRIEU, C., DE FREITAS, J. F. G. and DOUCET, A. (1999). Sequential MCMC for Bayesian model selection. In *Proc. IEEE Workshop Higher Order Statistics* 130–134. IEEE, New York.

[3] ANDRIEU, C., DOUCET, A. and HOLENSTEIN, R. (2010). Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **72** 269–342. MR2758115

[4] ANDRIEU, C., DOUCET, A. and TADIĆ, V. B. (2005). On-line parameter estimation in general state-space models. In *Proc. 44th IEEE Conf. on Decision and Control* 332–337. IEEE, New York.

[5] BENVENISTE, A., MÉTIVIER, M. and PRIOURET, P. (1990). *Adaptive Algorithms and Stochastic Approximations. Applications of Mathematics* (*New York*) **22**. Springer, Berlin. MR1082341

[6] BRIERS, M., DOUCET, A. and MASKELL, S. (2010). Smoothing algorithms for state-space models. *Ann. Inst. Statist. Math.* **62** 61–89. MR2577439

[7] BRIERS, M., DOUCET, A. and SINGH, S. S. (2005). Sequential auxiliary particle belief propagation. In *Proc. Conf. Fusion*. Philadelphia, PA.

[8] CAPPÉ, O. (2009). Online sequential Monte Carlo EM algorithm. In *Proc.* 15*th IEEE Workshop on Statistical Signal Processing* 37–40. IEEE, New York.

[9] CAPPÉ, O. (2011). Online EM algorithm for hidden Markov models. *J. Comput. Graph. Statist.* **20** 728–749. MR2878999

[10] CAPPÉ, O. and MOULINES, E. (2009). On-line expectation–maximization algorithm for latent data models. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **71** 593–613. MR2749909

[11] CAPPÉ, O., MOULINES, E. and RYDÉN, T. (2005). *Inference in Hidden Markov Models*. Springer, New York. MR2159833

[12] CARPENTER, J., CLIFFORD, P. and FEARNHEAD, P. (1999). An improved particle filter for non-linear problems. *IEE Proceedings—Radar, Sonar and Navigation* **146** 2–7.

[13] CARVALHO, C. M., JOHANNES, M. S., LOPES, H. F. and POLSON, N. G. (2010). Particle learning and smoothing. *Statist. Sci.* **25** 88–106. MR2741816

[14] CÉROU, F., DEL MORAL, P. and GUYADER, A. (2011). A nonasymptotic theorem for unnormalized Feynman–Kac particle models. *Ann. Inst. Henri Poincaré, B Probab. Stat.* **47** 629–649. MR2841068

[15] CHEN, R. and LIU, J. S. (2000). Mixture Kalman filters. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **62** 493–508. MR1772411

[16] CHOPIN, N. (2002). A sequential particle filter method for static models. *Biometrika* **89** 539–551. MR1929161

[17] CHOPIN, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Ann. Statist.* **32** 2385–2411. MR2153989

[18] CHOPIN, N., IACOBUCCI, A., MARIN, J. M., MENGERSEN, K., ROBERT, C. P., RYDER, R. and SCHÄUFER, C. (2011). On particle learning. In *Bayesian Statistics* 9 (J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D., Heckerman A. F. M. Smith and M., West, eds.) 317–360. Oxford Univ. Press, Oxford. MR3204011

[19] CHOPIN, N., JACOB, P. E. and PAPASPILIOPOULOS, O. (2013). SMC^2: An efficient algorithm for sequential analysis of state space models. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **75** 397–426. MR3065473

[20] COQUELIN, P. A., DEGUEST, R. and MUNOS, R. (2009). Sensitivity analysis in HMMs with application to likelihood maximization. In *Proc.* 22*th Conf. NIPS*. Vancouver.

[21] DAHLIN, J., LINDSTEN, F. and SCHÖN, T. B. (2015). Particle Metropolis–Hastings using gradient and Hessian information. *Stat. Comput.* **25** 81–92. MR3304908

[22] DEJONG, D. N., LIESENFELD, R., MOURA, G. V., RICHARD, J.-F. and DHARMARAJAN, H. (2013). Efficient likelihood evaluation of state-space representations. *Rev. Econ. Stud.* **80** 538–567. MR3054070

[23] DEL MORAL, P. (2004). *Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, New York. MR2044973

[24] DEL MORAL, P., DOUCET, A. and SINGH, S. S. (2009). Forward smoothing using sequential Monte Carlo. Technical Report 638, CUED-F-INFENG, Cambridge Univ. Preprint. Available at arXiv:1012.5390.

[25] DEL MORAL, P., DOUCET, A. and SINGH, S. S. (2010). A backward particle interpretation of Feynman–Kac formulae. *ESAIM Math. Model. Numer. Anal.* **44** 947–975. MR2731399

[26] DEL MORAL, P., DOUCET, A. and SINGH, S. S. (2015). Uniform stability of a particle approximation of the optimal filter derivative. *SIAM J. Control Optim.* **53** 1278–1304. MR3348115

[27] DEMPSTER, A. P., LAIRD, N. M. and RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B* **39** 1–38. MR0501537

[28] DOUC, R., GARIVIER, A., MOULINES, E. and OLSSON, J. (2011). Sequential Monte Carlo smoothing for general state space hidden Markov models. *Ann. Appl. Probab.* **21** 2109–2145. MR2895411

[29] DOUC, R., MOULINES, E. and RITOV, Y. (2009). Forgetting of the initial condition for the filter in general state-space hidden Markov chain: A coupling approach. *Electron. J. Probab.* **14** 27–49. MR2471658

[30] DOUCET, A., DE FREITAS, J. F. G. and GORDON, N. J., eds. (2001). *Sequential Monte Carlo Methods in Practice*. Springer, New York. MR1847783

[31] DOUCET, A., GODSILL, S. J. and ANDRIEU, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput.* **10** 197–208.

[32] DOUCET, A. and JOHANSEN, A. M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In *The Oxford Handbook of Nonlinear Filtering* 656–704. Oxford Univ. Press, Oxford. MR2884612

[33] DOUCET, A., PITT, M. K., DELIGIANNIDIS, G. and KOHN, R. (2015). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika* **102** 295–313.

[34] ELLIOTT, R. J., AGGOUN, L. and MOORE, J. B. (1995). *Hidden Markov Models: Estimation and Control. Applications of Mathematics* (*New York*) **29**. Springer, New York. MR1323178

[35] ELLIOTT, R. J., FORD, J. J. and MOORE, J. B. (2000). On-line consistent estimation of hidden Markov models. Technical report, Dept. Systems Engineering, Australian National Univ., Canberra.

[36] FEARNHEAD, P. (2002). Markov chain Monte Carlo, sufficient statistics, and particle filters. *J. Comput. Graph. Statist.* **11** 848–862. MR1951601

[37] FEARNHEAD, P. and MELIGKOTSIDOU, L. (2014). Augmentation schemes for particle MCMC. Preprint. Available at arXiv:1408.6980.

[38] FEARNHEAD, P., WYNCOLL, D. and TAWN, J. (2010). A sequential smoothing algorithm with linear computational cost. *Biometrika* **97** 447–464. MR2650750

[39] FERNÁNDEZ-VILLAVERDE, J. and RUBIO-RAMÍREZ, J. F. (2007). Estimating macroeconomic models: A likelihood approach. *Rev. Econ. Stud.* **74** 1059–1087. MR2353620

[40] FLURY, T. and SHEPHARD, N. (2009). Learning and filtering via simulation: Smoothly jittered particle filters. Economics Series Working Papers 469.

[41] FLURY, T. and SHEPHARD, N. (2011). Bayesian inference based only on simulated likelihood: Particle filter analysis of dynamic economic models. *Econometric Theory* **27** 933–956. MR2843833

[42] FORD, J. J. (1998). Adaptive hidden Markov model estimation and applications. Ph.D. thesis, Dept. Systems Engineering, Australian National Univ., Canberra. Available at http://infoeng.rsise.anu.edu.au/files/jason_ford_thesis.pdf.

[43] FULOP, A. and LI, J. (2013). Efficient learning via simulation: A marginalized resample–move approach. *J. Econometrics* **176** 146–161. MR3084050

[44] GILKS, W. R. and BERZUINI, C. (2001). Following a moving target—Monte Carlo inference for dynamic Bayesian models. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **63** 127–146. MR1811995

[45] GODSILL, S. J., DOUCET, A. and WEST, M. (2004). Monte Carlo smoothing for nonlinear times series. *J. Amer. Statist. Assoc.* **99** 156–168. MR2054295

[46] GORDON, N. J., SALMOND, D. J. and SMITH, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F, Comm., Radar, Signal. Proc.* **140** 107–113.

[47] HIGUCHI, T. (2001). Self-organizing time series model. In *Sequential Monte Carlo Methods in Practice. Stat. Eng. Inf. Sci.* 429–444. Springer, New York. MR1847803

[48] HÜRZELER, M. and KÜNSCH, H. R. (1998). Monte Carlo approximations for general state-space models. *J. Comput. Graph. Statist.* **7** 175–193. MR1649366

[49] HÜRZELER, M. and KÜNSCH, H. R. (2001). Approximating and maximising the likelihood for a general state-space model. In *Sequential Monte Carlo Methods in Practice. Stat. Eng. Inf. Sci.* 159–175. Springer, New York. MR1847791

[50] IONIDES, E. L., BHADRA, A., ATCHADÉ, Y. and KING, A. (2011). Iterated filtering. *Ann. Statist.* **39** 1776–1802. MR2850220

[51] IONIDES, E. L., BRETÓ, C. and KING, A. A. (2006). Inference for nonlinear dynamical systems. *Proc. Natl. Acad. Sci. USA* **103** 18438–18443.

[52] KIM, S., SHEPHARD, N. and CHIB, S. (1998). Stochastic volatility: Likelihood inference and comparison with ARCH models. *Rev. Econ. Stud.* **65** 361–393.

[53] KITAGAWA, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statist.* **5** 1–25. MR1380850

[54] KITAGAWA, G. (1998). A self-organizing state-space model. *J. Amer. Statist. Assoc.* **93** 1203–1215.

[55] KITAGAWA, G. (2014). Computational aspects of sequential Monte Carlo filter and smoother. *Ann. Inst. Statist. Math.* **66** 443–471. MR3211870

[56] KITAGAWA, G. and SATO, S. (2001). Monte Carlo smoothing and self-organising state-space model. In *Sequential Monte Carlo Methods in Practice. Stat. Eng. Inf. Sci.* 177–195. Springer, New York. MR1847792

[57] KLAAS, M., BRIERS, M., DE FREITAS, N., DOUCET, A., MASKELL, S. and LANG, D. (2006). Fast particle smoothing: If I had a million particles. In *Proc. International Conf. Machine Learning* 481–488. Pittsburgh, PA.

[58] KÜNSCH, H. R. (2013). Particle filters. *Bernoulli* **19** 1391–1403. MR3102556

[59] LEE, A. (2008). Towards smoother multivariate particle filters. M.Sc. Computer Science, Univ. British Columbia, Vancouver, BC.

[60] LEE, A. and WHITELEY, N. (2014). Forest resampling for distributed sequential Monte Carlo. Preprint. Available at arXiv:1406.6010.

[61] LEE, D. S. and CHIA, K. K. (2002). A particle algorithm for sequential Bayesian parameter estimation and model selection. *IEEE Trans. Signal Process.* **50** 326–336.

[62] LE CORFF, S. and FORT, G. (2013). Online expectation maximization based algorithms for inference in hidden Markov models. *Electron. J. Stat.* **7** 763–792. MR3040559

[63] LE CORFF, S. and FORT, G. (2013). Convergence of a particle-based approximation of the block online expectation maximization algorithm. *ACM Trans. Model. Comput. Simul.* **23** Art. 2, 22. MR3034212

[64] LE GLAND, F. and MEVEL, M. (1997). Recursive estimation in hidden Markov models. In *Proc. 36th IEEE Conf. Decision and Control* 3468–3473. San Diego, CA.

[65] LIN, M., CHEN, R. and LIU, J. S. (2013). Lookahead strategies for sequential Monte Carlo. *Statist. Sci.* **28** 69–94. MR3075339

[66] LINDSTEN, F., JORDAN, M. I. and SCHÖN, T. B. (2014). Particle Gibbs with ancestor sampling. *J. Mach. Learn. Res.* **15** 2145–2184. MR3231604

[67] LIU, J. and WEST, M. (2001). Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo Methods in Practice*. Springer, New York. MR1847793

[68] LIU, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer, New York. MR1842342

[69] LIU, J. S. and CHEN, R. (1998). Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.* **93** 1032–1044. MR1649198

[70] LOPES, H. F., CARVALHO, C. M., JOHANNES, M. S. and POLSON, N. G. (2011). Particle learning for sequential Bayesian computation. In *Bayesian Statistics* 9 (J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D., Heckerman A. F. M. Smith and M., West, eds.). Oxford Univ. Press, Oxford. MR3204011

[71] LOPES, H. F. and TSAY, R. S. (2011). Particle filters and Bayesian inference in financial econometrics. *J. Forecast.* **30** 168–209. MR2758809

[72] MALIK, S. and PITT, M. K. (2011). Particle filters for continuous likelihood evaluation and maximisation. *J. Econometrics* **165** 190–209. MR2846644

[73] NEMETH, C., FEARNHEAD, P. and MIHAYLOVA, L. (2013). Particle approximations of the score and observed information matrix for parameter estimation in state space models with linear computational cost. Preprint. Available at arXiv:1306.0735.

[74] OLSSON, J., CAPPÉ, O., DOUC, R. and MOULINES, E. (2008). Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli* **14** 155–179. MR2401658

[75] OLSSON, J. and WESTERBORN, J. (2014). Efficient particle-based online smoothing in general hidden Markov models: The PaRIS algorithm. Preprint. Available at arXiv:1412.7550.

[76] OUDJANE, N. and RUBENTHALER, S. (2005). Stability and uniform particle approximation of nonlinear filters in case of non ergodic signals. *Stoch. Anal. Appl.* **23** 421–448. MR2140972

[77] PANINSKI, L., AHMADIAN, Y., FERREIRA, D. G., KOYAMA, S., RAD, K. R., VIDNE, M., VOGELSTEIN, J. and WU, W. (2010). A new look at state-space models for neural data. *J. Comput. Neurosci.* **29** 107–126. MR2721336

[78] PITT, M. K. and SHEPHARD, N. (1999). Filtering via simulation: Auxiliary particle filters. *J. Amer. Statist. Assoc.* **94** 590–599. MR1702328

[79] PITT, M. K., SILVA, R. D. S., GIORDANI, P. and KOHN, R. (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *J. Econometrics* **171** 134–151. MR2991856

[80] POLSON, N. G., STROUD, J. R. and MÜLLER, P. (2008). Practical filtering with sequential parameter learning. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **70** 413–428. MR2424760

[81] 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. MR2804210

[82] SCHÖN, T. B., WILLS, A. and NINNESS, B. (2011). System identification of nonlinear state-space models. *Automatica J. IFAC* **47** 39–49. MR2878244

[83] SHERLOCK, C., THIERY, A. H., ROBERTS, G. O. and ROSENTHAL, J. S. (2015). On the efficiency of pseudo-marginal random walk Metropolis algorithms. *Ann. Statist.* **43** 238–275. MR3285606

[84] STORVIK, G. (2002). Particle filters in state space models with the presence of unknown static parameters. *IEEE Trans. Signal Process.* **50** 281–289.

[85] TAGHAVI, E., LINDSTEN, F., SVENSSON, L. and SCHÖN, T. B. (2013). Adaptive stopping for fast particle smoothing. In *Proc. IEEE ICASSP* 6293–6297. Vancouver, BC.

[86] VERCAUTEREN, T., TOLEDO, A. and WANG, X. (2005). Online Bayesian estimation of hidden Markov models with unknown transition matrix and applications to IEEE 802.11 networks. In *Proc. IEEE ICASSP, Vol. IV* 13–16. Philadelphia, PA.

[87] WEST, M. and HARRISON, J. (1997). *Bayesian Forecasting and Dynamic Models*, 2nd ed. Springer, New York. MR1482232

[88] WESTERBORN, J. and OLSSON, J. (2014). Efficient particle-based online smoothing in general hidden Markov models. In *Proc. IEEE ICASSP* 8003–8007. Florence.

[89] WHITELEY, N. (2010). Discussion of Particle Markov chain Monte Carlo methods. *J. Royal Stat. Soc.* **72** 306–307.

[90] WHITELEY, N. (2013). Stability properties of some particle filters. *Ann. Appl. Probab.* **23** 2500–2537. MR3127943

[91] WHITELEY, N., ANDRIEU, C. and DOUCET, A. (2010). Efficient Bayesian inference for switching state–space models using discrete particle Markov chain Monte Carlo methods. Preprint. Available at arXiv:1011.2437.

[92] WHITELEY, N. and LEE, A. (2014). Twisted particle filters. *Ann. Statist.* **42** 115–141. MR3178458

[93] WILKINSON, D. J. (2012). *Stochastic Modelling for Systems Biology*, 2nd ed. CRC Press, Boca Raton, FL.

[94] YILDIRIM, S., SINGH, S. S. and DOUCET, A. (2013). An online expectation–maximization algorithm for change-point models. *J. Comput. Graph. Statist.* **22** 906–926. MR3173749

[95] YUAN, Y.-X. (2008). Step-sizes for the gradient method. In *Third International Congress of Chinese Mathematicians. Part* 1, 2. *AMS/IP Stud. Adv. Math.*, 42, *Pt.* 1 **2** 785–796. Amer. Math. Soc., Providence, RI. MR2409671