An overview of existing methods and recent advances in sequential Monte Carlo

Olivier Cappé, Simon J. Godsill and Eric Moulines

***Abstract*— It is now over a decade since the pioneering contribution of Gordon et al. (1993), which is commonly regarded as the first instance of modern sequential Monte Carlo (SMC) approaches. Initially focussed on applications to tracking and vision, these techniques are now very widespread and have had a significant impact in virtually all areas of signal and image processing concerned with Bayesian dynamical models. This article is intended to serve both as an introduction to SMC algorithms for non-specialists and as a reference to recent contributions in domains where the techniques are still under significant development, including smoothing, estimation of fixed parameters and use of SMC methods beyond the standard filtering contexts.**

***Index Terms*— State-space model; Filtering, prediction & smoothing; Sequential Monte Carlo; Bayesian dynamical model; Particle Filter; Hidden Markov Models; Parameter estimation; Tracking; .**

I. Introduction

Consider the following generic nonlinear dynamic system described in state-space form:

- System model

$$x_t = a(x_{t-1}, u_t) \leftrightarrow \overbrace{f(x_t|x_{t-1})}^{\text{Transition Density}} \tag{1}$$

- Measurement model

$$y_t = b(x_t, v_t) \leftrightarrow \overbrace{g(y_t|x_t)}^{\text{Observation Density}} \tag{2}$$

By these equations we mean that the hidden states x_t and data y_t are assumed to be generated by nonlinear functions $a(\cdot)$ and $b(\cdot)$, respectively, of the state and noise disturbances u_t and v_t. The precise form of the functions and the assumed probability distributions of the state u_t and the observation v_t noises imply via a change of variables the transition probability density function $f(x_t|x_{t-1})$ and the observation probability density function $g(y_t|x_t)$. We make the assumption that x_t is Markovian, i.e. its conditional probability density given the past states $x_{0:t-1} \stackrel{\text{def}}{=} (x_0, \ldots, x_{t-1})$ depends only on x_{t-1} through the transition density $f(x_t|x_{t-1})$, and that the conditional probability density of y_t given the states $x_{0:t}$ and the past observations $y_{0:t-1}$ depends only upon x_t through the conditional likelihood $g(y_t|x_t)$. We further assume that the initial state x_0 is distributed according to a density function $\pi_0(x_0)$. Such nonlinear dynamic systems arise frequently from many areas in science and engineering such as target tracking, computer vision, terrain referenced navigation, finance, pollution monitoring, communications, audio engineering, to list but a few.

O. Cappé and E. Moulines are with Télécom Paris / CNRS LTCI, 46 rue Barrault, 75634 Paris Cedex 13, France; S. Godsill is with the Signal Processing and Communications Lab., University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK.

To give a concrete example of such a model consider:

Example 1: Nonlinear time series model

We consider here a simple nonlinear time series model which has been used extensively in the literature for benchmarking numerical filtering techniques [1], [2], [3]. The state-space equations are as follows:

$$x_t = \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + u_t \ ,$$
$$y_t = \frac{x_t^2}{20} + v_t \ ,$$

where $u_t \sim \mathcal{N}(0, \sigma_u^2)$ and $v_t \sim \mathcal{N}(0, \sigma_v^2)$ and here $\sigma_u^2 = 10$ and $\sigma_v^2 = 1$ are considered fixed and known; $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2. The initial state distribution is $x_0 \sim \mathcal{N}(0, 10)$. The representation in terms of densities $f(x_t|x_{t-1})$ and $g(y_t|x_t)$ is given by:

$$f(x_t|x_{t-1}) = \mathcal{N}\left(x_t \middle| \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t), \sigma_u^2\right)$$
$$g(y_t|x_t) = \mathcal{N}\left(y_t \middle| \frac{x_t^2}{20}, \sigma_v^2\right)$$

The form of these densities was straightforward to obtain in this case. For more complex cases a Jacobian term might be required when either x_t or y_t is a nonlinear function of u_t or v_t, respectively. Note that we usually consider only probability density functions $p(x)$ but in some specific cases, we will use the notation $p(dx)$ to refer to the associated probability measure.

A dynamical model of this sort may easily be simulated owing to the Markovian assumptions on x_t and y_t, which imply that the joint probability density of states and observations, denoted $\pi_{0:T,0:T}(x_{0:T}, y_{0:T})$, may be factorised as

$$\pi_{0:T,0:T}(x_{0:T}, y_{0:T}) = \pi_0(x_0)g(y_0|x_0) \times \prod_{t=1}^{T} f(x_t|x_{t-1})g(y_t|x_t) \ .$$

A graphical representation of the dependencies between different states and observations is shown in Figure 1.

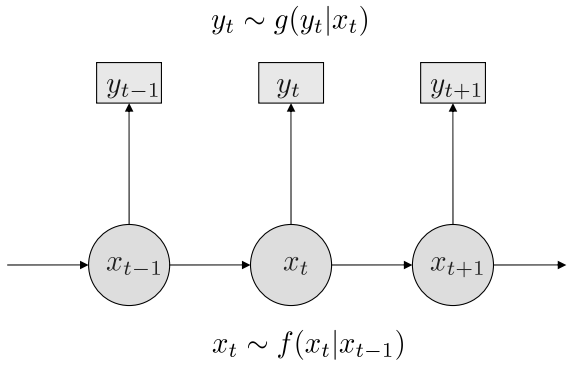

Fig. 1. Graphical model illustrating the Markovian dependencies between states and observations.

In this model, states and data may be sampled one by one by successively drawing random samples from the transition and the observation densities as indicated in Algorithm 1 below.

Algorithm 1 Generating from a State-Space Model

Initialisation: sample $\tilde{x}_0 \sim \pi_0(x_0)$, $\tilde{y}_0 \sim g(y_0|\tilde{x}_0)$.
for $t = 1, \ldots, T$ **do**
  Sample $\tilde{x}_t \sim f(x_t|\tilde{x}_{t-1})$.
  Sample $\tilde{y}_t \sim g(y_t|\tilde{x}_t)$.
end for
$(\tilde{x}_0, \ldots, \tilde{x}_T, \tilde{y}_0, \ldots, \tilde{y}_T)$ is a random draw from $\pi_{0:T,0:T}(x_{0:T}, y_{0:T})$.

The ability to simulate random states and to evaluate the transition and observation densities (at least up to an unknown normalising constant) will be the chief components of the particle filtering algorithms described later.

Statistical inference for the general nonlinear dynamic system above involves computing the *posterior distribution* of a collection of state variables $x_{s:s'} \stackrel{\text{def}}{=} (x_s, \ldots, x_{s'})$ conditioned on a batch of observations, $y_{0:t} = (y_0, \ldots, y_t)$, which we denote $\pi_{s:s'|0:t}(x_{s:s'}|y_{0:t})$. Specific problems include *filtering*, for $s = s' = t$, *fixed lag smoothing*, when $s = s' = t - L$ and fixed interval smoothing, if $s = 0$ and $s' = t$. Despite the apparent simplicity of the above problem, the posterior distribution can be computed in closed form only in very specific cases, principally, the linear Gaussian model (where the functions $a()$ and $b()$ are linear and u_t and v_t are Gaussian) and the discrete hidden Markov model (where x_t takes its values in a finite alphabet). In the vast majority of cases, nonlinearity or non-Gaussianity render an analytic solution intractable [4], [5], [6], [7].

The classical inference methods for nonlinear dynamic systems are the extended Kalman filter (EKF) and its variants, which are based on linearisation of the state and measurement equations along the trajectories [8]. The EKF has been successfully applied to many nonlinear filtering problems. However, the EKF is known to fail if the system exhibits substantial nonlinearity and/or if the state and the measurement noise are significantly non-Gaussian.

Many algorithms have been developed to correct poor performance in the EKF algorithm. One of the earliest approaches was to approximate the posterior distribution by expansion in a pre-specified function basis. For example, the Gaussian sum filter [9] approximates the posterior density by a mixture of Gaussians; (see [10] for an in-depth discussion and some generalisations).

More recently, several algorithms have been proposed that attempt to choose a set of deterministic points to represent the posterior distribution accurately. Two representative algorithms in this class are the unscented Kalman filter (UKF) [11], [12] and the Gaussian quadrature Kalman filter (QKF) [13]. The UKF is based on the so-called "sigma points", and the QKF is based on the Gauss-Hermite quadrature integration rule. One of the significant advantages of these algorithms is that they do not require the evaluation of the Jacobian matrix, which is often a computationally intensive component in the EKF algorithm. Whereas these techniques have been applied successfully in certain settings, they are valid only if the posterior distribution can be closely approximated by a Gaussian distribution, which implies, in particular, that it remains unimodal, which is typically not true in many nonlinear state-space scenarios.

These limitations have stimulated the interest in alternative strategies that can handle more general state and measurement equations, and which do not put strong a priori constraints on the behaviour of the posterior distributions. Among these, Monte Carlo methods, in which the posterior distribution is represented by a collection of random points, play a central role.

The use of Monte Carlo methods for nonlinear filtering can be traced back to the pioneering works of [14] and [15]. These early attempts were based on sequential versions of the importance sampling paradigm, a technique that amounts to simulating samples under an instrumental distribution and then approximating the target distributions by weighting these samples using appropriately defined importance weights. In the nonlinear filtering context, importance sampling algorithms can be implemented sequentially in the sense that, by defining appropriately a sequence of instrumental distributions, it is not necessary to regenerate the population of samples from scratch upon the arrival of each new observation. This algorithm is called sequential importance sampling, often abbreviated to SIS. Although the SIS algorithm has been known since the early 1970s, its use in nonlinear filtering problems was rather limited at that time. Most likely, the available computational power was then too limited to allow convincing applications of these methods. Another less obvious reason is that the SIS algorithm suffers from a major drawback that was not clearly identified and properly cured until [3]. As the number of iterations increases, the importance weights tend to degenerate, a phenomenon known as *sample impoverishment* or *weight degeneracy*. Basically, in the long run most of the samples have very small normalised importance weights and thus do not significantly contribute to the approximation of the target distribution. The solution proposed by [3] is to allow rejuvenation of the set of samples by replicating the samples

with high importance weights and removing samples with low weights, much as in the resampling step for the (non-sequential) sampling and importance resampling (SIR) algorithm [16]. The so-called *bootstrap filter* of [3] was the first successful application of sequential Monte Carlo techniques to the field of nonlinear filtering. Since then, there have been several independent variants of similar filtering ideas, including the Condensation filter [17], Monte Carlo filter [1], Sequential imputations [18], and the Particle filter [19].

Sequential Monte Carlo (SMC) methods offer a number of significant advantages compared with other techniques currently available. These advantages arise principally as a result of the generality of the approach, which allows inference of full posterior distributions in general state-space models, which may be both nonlinear and non-Gaussian. As a result of this, SMC methods allow for computation of all kinds of moments, quantiles and highest posterior density regions, whereas EKF and UKF allow approximation of only the first and second-order statistics. In particular, an appropriate specification of the state-space model allows SMC to handle constraints on the state-space, which may arise, depending on the application, from physical limitations (target speed, presence of obstacles, etc..), or general prior knowledge about the range of the state values. SMC methods are scalable, and the precision of the estimates depends only on the number of particles used in approximating the distribution.

To date, SMC methods have been successfully applied in many different fields including computer vision, signal processing, tracking, control, econometrics, finance, robotics, and statistics; see [20], [21], [6], [7] and the references therein for a good review coverage.

The paper is organised as follows. In section II, we recall the basics of simulation-based inference, importance sampling and particle filters. In subsequent sections we cover a selection of new and recent developments. In section III, we describe methods to perform fixed-lag and fixed-interval smoothing. In section IV, we present methods to estimate unknown system parameters in batch and on-line settings. In section V, we describe applications of SMC outside the filtering context, namely, adaptive simulation of posterior densities over large dimensional spaces and rare event simulations. This tutorial aims to highlight basic methodology and up and coming areas for particle filtering; more established topics and a range of applications are extensively reviewed in a number of excellent papers [22], [23], [24], [25], [26], [27].

II. Simulation basics

A. Importance Sampling and Resampling

In the Monte Carlo method, we are concerned with estimating the properties of some highly complex probability distribution p, for example computing expectations of the form:

$$\bar{h} \stackrel{\text{def}}{=} \int h(x)p(x)dx \ ,$$

where $h(\cdot)$ is some useful function for estimation, for example the mean value is obtained with $h(x) = x$. In cases where this cannot be achieved analytically, the approximation problem can be tackled indirectly by generating *random samples* from p, denote these $\{x^{(i)}\}_{1\leq i\leq N}$, and approximating the distribution p by point masses so that

$$\bar{h} \approx \frac{1}{N}\sum_{i=1}^{N} h(x^{(i)}) \ .$$

See Figs. 2 and 3 for a graphical example where a complex non-Gaussian density function is represented using Monte Carlo samples. Clearly N needs to be large in order to give a good coverage of all regions of interest.

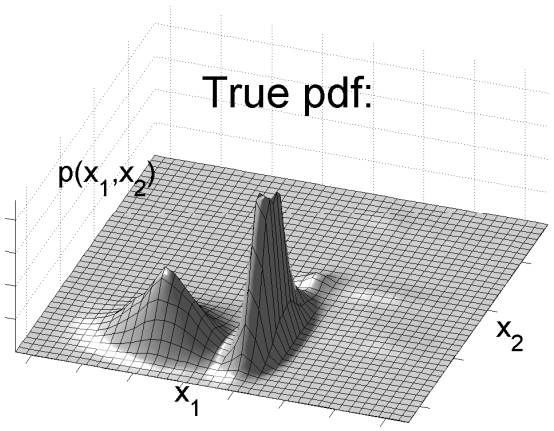

Fig. 2. Two-dimensional probability density function.

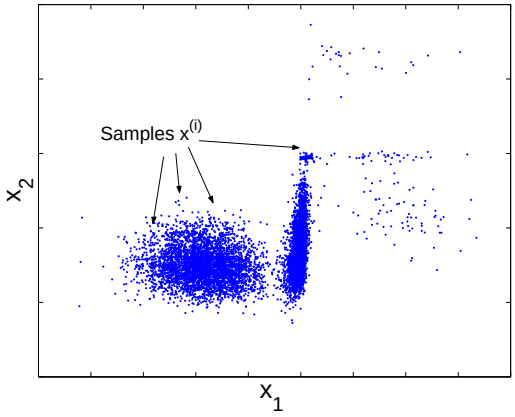

Fig. 3. Two-dimensional probability density function - representation by random points, or 'particles'.

More generally, when we cannot sample directly from the distribution p, we can sample from another distribution q (the *importance distribution*, or *instrumental distribution*) having a support larger than p. So we make N random draws $x^{(i)}$, $i = 1, \ldots, N$ from q instead of p. Now we have to make a correction to ensure that the obtained estimate is an unbiased estimator of $\bar{h}$. This correction involves assigning a positive weight to each of the random points. It turns out that the required value of the weights is proportional to the ratio $r \stackrel{\text{def}}{=} p/q$ evaluated at the random points; the function r is

termed the *importance function*. The expectation $\bar{h}$ can thus be estimated using a *weighted average*:

$$\bar{h} = \int h(x)\frac{q(x)p(x)}{q(x)}dx$$
$$= \int h(x)r(x)q(x)dx \approx \sum_{i=1}^{N} \frac{\tilde{\omega}^{(i)}}{\sum_{j=1}^{N}\tilde{\omega}^{(j)}} h(x^{(i)}) \quad (3)$$

where $\tilde{\omega}^{(i)} \stackrel{\text{def}}{=} r(x^{(i)}) = p(x^{(i)})/q(x^{(i)})$ is termed the *unnormalised importance weight*.

Remark 1: In many situations, the target distribution p or the importance distribution q are known only up to a normalising factor (this is particularly true when applying importance sampling ideas to state-space models and, more generally, in Bayesian statistical inference; see below). The importance function $r = p/q$ is then known only up to a (constant) scaling factor. In (3), the weights are renormalised to sum to unity and hence the estimator of $\bar{h}$ does not require knowledge of the actual normalising factor. Theoretical issues relating to this renormalisation are discussed in [28].

Although importance sampling is primarily intended to overcome difficulties with direct sampling from p when approximating expectations under p, it can also be used for sampling from the distribution p. The latter can be achieved by the *sampling importance resampling* (or SIR) method originally introduced by [16], [29]. Sampling importance resampling is a two-stage procedure in which importance sampling is followed by an additional random sampling step, as discussed below. In the first stage, an i.i.d. sample $(\tilde{x}^{(1)}, \ldots, \tilde{x}^{(M)})$ is drawn from the importance distribution q, and one computes the normalised version of the importance weights,

$$\omega^{(i)} \stackrel{\text{def}}{=} \frac{\tilde{\omega}^{(i)}}{\sum_{i=1}^{M}\tilde{\omega}^{(i)}}\ ,\ i = 1, \ldots, M\ . \quad (4)$$

In the resampling stage, a sample of size N denoted by $x^{(1)}, \ldots, x^{(N)}$ is drawn from the intermediate set of points $\tilde{x}^{(1)}, \ldots, \tilde{x}^{(M)}$, taking proper account of the weights computed in (4). This principle is illustrated in Figure 4.

There are several ways of implementing this basic idea, the most obvious approach being sampling with replacement, with the probability of sampling each $x^{(i)}$ set equal to the normalised importance weight $\omega^{(i)}$. Hence the number of times $N^{(i)}$ that each particular point $\tilde{x}^{(i)}$ in the first-stage sample is selected follows a binomial $\mathrm{Bin}(N, \omega^{(i)})$ distribution. The vector $(N^{(1)}, \ldots, N^{(M)})$ is distributed according to $\mathrm{Mult}(N, \omega^{(1)}, \ldots, \omega^{(M)})$, the multinomial distribution with parameter N and probabilities of success $(\omega^{(1)}, \ldots, \omega^{(M)})$. In this resampling step, the points in the first-stage sample that are associated with small normalised importance weights are most likely to be discarded, whereas the best points in the sample are replicated in proportion to their importance weights. In most applications, it is typical to choose M, the size of the first-stage sample, larger (and sometimes much larger) than N.

While this resampling step is unnecessary in the non-recursive framework, and would always increase the Monte Carlo variance of our estimators, it is a vital component of the sequential schemes which follow, avoiding degeneracy of the importance weights over time. While the multinomial resampling scheme above is the most natural first approach, it should be noted that improvements can be achieved through variance reduction strategies such as stratification. Some of these alternative sampling schemes guarantee a reduced Monte Carlo variance, at a comparable computational cost [30], [31], [32], [33]. We will sometimes refer to the resampling step as a *selection* step in the sequel.

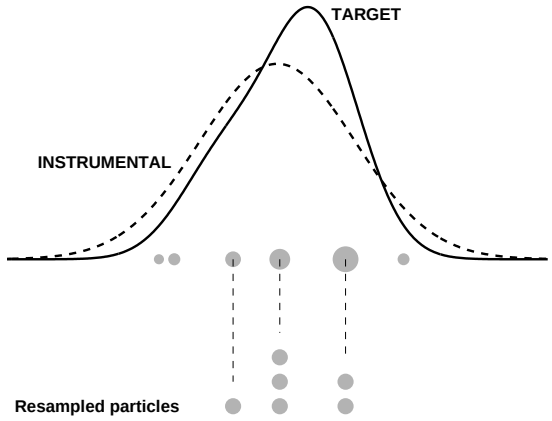

Fig. 4. Principle of resampling. Top: the sample drawn from q (dashed line) with associated normalised importance weights depicted by bullets with radii proportional to the normalised weights (the target density corresponding to p is plotted as a solid line); Bottom: after resampling, all particles have the same importance weight, and some of them have been either discarded or duplicated (here $M = N = 6$).

B. Sequential Monte-Carlo methods

We now specialise the sampling techniques considered above to the sequential setting of the state-space model. Starting with the initial, or 'prior', distribution $\pi_0(x_0)$, the posterior density $\pi_{0:t|0:t}(x_{0:t}|y_{0:t})$ can be obtained using the following *prediction-correction* recursion [34]:

- *Prediction*

$$\pi_{0:t|0:t-1}(x_{0:t}|y_{0:t-1})$$
$$= \pi_{0:t-1|0:t-1}(x_{0:t-1}|y_{0:t-1})f(x_t|x_{t-1})\ , \quad (5)$$

- *Correction*

$$\pi_{0:t|0:t}(x_{0:t}|y_{0:t}) = \frac{g(y_t|x_t)\pi_{0:t|0:t-1}(x_{0:t}|y_{0:t-1})}{\ell_{t|0:t-1}(y_t|y_{0:t-1})}\ , \quad (6)$$

  where $\ell_{t|0:t-1}$ is the predictive distribution of y_t given the past observations $y_{0:t-1}$. For a fixed data realization, this term is a normalising constant (independent of the state); it will not be necessary to compute this term in standard implementations of SMC methods.

We would like to sample from $\pi_{0:t|0:t}(x_{0:t}|y_{0:t})$; since it is generally impossible to sample directly from this distribution, we resort to a sequential version of the importance sampling and resampling procedure outlined above. Conceptually, we sample N *particle paths* $\tilde{x}_{0:t}^{(i)}$, $i = 1, \ldots, N$, from a convenient importance distribution $q_{0:t}(x_{0:t}|y_{0:t})$, and compute the

unnormalised importance weights

$$\tilde{\omega}_t^{(i)} = \frac{\pi_{0:t|0:t}(\tilde{x}_{0:t}^{(i)}|y_{0:t})}{q_{0:t}(\tilde{x}_{0:t}^{(i)}|y_{0:t})} , \quad i = 1, \ldots, N. \tag{7}$$

Using this *weighted sample* $\{(\tilde{x}_{0:t}^{(i)}, \tilde{\omega}_t^{(i)})\}_{1 \leq i \leq N}$, we may approximate the expectation of any function h defined on the path space using the self-normalised importance sampling estimator,

$$\bar{h} = \int h(x_{0:t}) \pi_{0:t|0:t}(x_{0:t}|y_{0:t}) dx_{0:t} \approx \sum_{i=1}^{N} \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^{N} \tilde{\omega}_t^{(j)}} h(\tilde{x}_{0:t}^{(i)}) . \tag{8}$$

As in the case of the non-sequential importance sampling above, we will use in the following the notation $\omega_t^{(i)}$ to refer to the *normalised weight*, so that $\omega_t^{(i)} = \tilde{\omega}_t^{(i)} / \sum_{j=1}^{N} \tilde{\omega}_t^{(j)}$. We may also sample (approximately) from the posterior distribution $\pi_{0:t|0:t}$ by drawing N particle paths $\{x_{0:t}^{(i)}\}_{1 \leq i \leq N}$ from the collection $\{\tilde{x}_{0:t}^{(i)}\}_{1 \leq i \leq N}$ according to the importance weights $\{\omega_t^{(i)}\}_{1 \leq i \leq N}$.

The trick behind the sequential importance sampling procedure is to choose the importance distribution in a clever way so that all these steps can be carried out sequentially. To achieve this we construct the proposal such that it factorises in a form similar to that of the target posterior distribution:

$$q_{0:t}(x_{0:t}|y_{0:t}) = \overbrace{q_{0:t-1}(x_{0:t-1}|y_{0:t-1})}^{\text{Keep existing path}} \overbrace{q_t(x_t|x_{t-1}, y_t)}^{\text{extend path}} . \tag{9}$$

The unnormalised importance weights then take the following appealing form

$$\begin{aligned} \tilde{\omega}_t^{(i)} &= \frac{\pi_{0:t|0:t}(\tilde{x}_{0:t}^{(i)}|y_{0:t})}{q_{0:t}(\tilde{x}_{0:t}^{(i)}|y_{0:t})} \\ &\propto \omega_{t-1}^{(i)} \times \frac{f(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)}) g(y_t|\tilde{x}_t^{(i)})}{q_t(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)}, y_t) \ell_{t|0:t-1}(y_t|y_{0:t-1})} , \end{aligned} \tag{10}$$

where the symbol $\propto$ is used to denote proportionality, up to a normalisation constant (which does not matter here due to the use of the self-normalised form of importance sampling). This multiplicative decomposition implies that the importance weights may be computed recursively in time as successive observations become available, and without having to modify past paths, prior to time t. In the sequential Monte Carlo literature, the multiplicative update factor on the right-hand side of (10) is often called the *incremental weight*. Note that the scaling factor $\ell_{t|0:t-1}(y_t|y_{0:t-1})$, which would in general cases be difficult to evaluate, does not depend on the state sequence, and hence need not in fact be computed, since the weights will subsequently be renormalised as in (8).

An important feature of the basic sequential importance sampling method, as originally proposed in [14], [15], is that the N trajectories $\tilde{x}_{0:t}^{(1)}, \ldots, \tilde{x}_{0:t}^{(N)}$ are independent and identically distributed. Following the terminology in use in the nonlinear filtering community, we shall refer to the sample at time index t, $\tilde{x}_t^{(1)}, \ldots, \tilde{x}_t^{(N)}$, as the population (or system) of *particles* and to $\tilde{x}_{0:t}^{(i)}$ for a specific value of the particle index i as the history (or trajectory, or path) of the ith particle. The sequential importance sampling method is summarised in Algorithm 2.

Algorithm 2 Sequential Importance Sampling (SIS)

for $i = 1, \ldots, N$ **do** ▷ Initialisation
  Sample $\tilde{x}_0^{(i)} \sim q_0(x_0|y_0)$.
  Assign initial importance weights

$$\tilde{\omega}_0^{(i)} = \frac{g(y_0|\tilde{x}_0^{(i)}) \pi_0(\tilde{x}_0^{(i)})}{q_0(\tilde{x}_0^{(i)}|y_0)} .$$

end for
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, N$ **do**
    Propagate particles:

$$\tilde{x}_t^{(i)} \sim q_t(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)}, y_t) .$$

    Compute weight:

$$\tilde{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \frac{g(y_t|\tilde{x}_t^{(i)}) f(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)})}{q_t(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)}, y_t)} .$$

  end for
  Normalise weights:

$$\omega_t^{(i)} = \tilde{\omega}_t^{(i)} / \sum_{j=1}^{N} \tilde{\omega}_t^{(j)} , \quad i = 1, \ldots, N.$$

  Compute filtering estimate:

$$\bar{h}_t = \sum_{i=1}^{N} \omega_t^{(i)} h_t(\tilde{x}_t^{(i)})$$

end for

Despite quite successful results for short data records, it turns out that the sequential importance sampling approach discussed so far is bound to fail in the long run. In particular, the weights will become highly degenerate after a few time steps, in the sense that a small proportion of them contain nearly all of the probability mass, and hence most particles contribute nothing significant to the expectation estimates; see for example [35]. The reason for this is that we are effectively sampling from a very high dimensional state-space, i.e. the entire path history of state variables up to time t, which increases with each time increment. Hence it is naive to imagine that the distribution may be sampled effectively using a fixed and practically realisable sample size. In practice we will often be concerned with low-dimensional marginals such as the filtering distribution $\pi_{t|0:t}$ or predictive distribution $\pi_{t+1|0:t}$, and this suggests a solution based on the resampling ideas discussed above. In the short-term, resampling does imply some additional Monte Carlo variance; however, resampling avoids accumulation of error over time and renders the approximation to the filtering and predictive distributions much more stable.

The basic resampling method comprises sampling N draws from the current population of particles using the normalised

weights as probabilities of selection. Thus, trajectories with small importance weights are eliminated, whereas those with large importance weights are replicated. After resampling, the normalised importance weights are reset to $1/N$. Resampling will however have two important consequences. First, the overall algorithm cannot anymore be seen as a simple instance of the importance sampling approach since it implies repeated applications of the importance sampling and resampling steps. This obviously renders the complete algorithm much harder to analyse from a theoretical perspective. Next, the resampled trajectories $x_{0:t}^{(i)}$ are no longer independent.

We now state in Algorithm 3 the standard particle filtering algorithm, with general proposal function and optional resampling at every step. There are straightforward variants of the algorithm that propagate more particles than are selected, and which have variable numbers of particles at each time step.

Algorithm 3 Particle Filter

for $i = 1, \ldots, N$ **do** ▷ Initialisation
  Sample $\tilde{x}_0^{(i)} \sim q_0(x_0|y_0)$.
  Assign initial importance weights

$$\tilde{\omega}_0^{(i)} = \frac{g(y_0|\tilde{x}_0^{(i)})\pi_0(\tilde{x}_0^{(i)})}{q_0(\tilde{x}_0^{(i)}|y_0)} .$$

end for
for $t = 1, \ldots, T$ **do**
  if Resampling **then**
    Select N particle indices $j_i \in \{1, \ldots, N\}$ according to weights

$$\{\omega_{t-1}^{(j)}\}_{1 \le j \le N} .$$

    Set $x_{t-1}^{(i)} = \tilde{x}_{t-1}^{(j_i)}$, and $\omega_{t-1}^{(i)} = 1/N$, $i = 1, \ldots, N$.
  else
    Set $x_{t-1}^{(i)} = \tilde{x}_{t-1}^{(i)}$, $i = 1, \ldots, N$.
  end if
  for $i = 1, \ldots, N$ **do**
    Propagate:

$$\tilde{x}_t^{(i)} \sim q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, y_t) .$$

    Compute weight:

$$\tilde{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \frac{g(y_t|\tilde{x}_t^{(i)}) f(\tilde{x}_t^{(i)}|x_{t-1}^{(i)})}{q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, y_t)} .$$

  end for
  Normalise weights:

$$\omega_t^{(i)} = \tilde{\omega}_t^{(i)} \Big/ \sum_{j=1}^{N} \tilde{\omega}_t^{(j)} , \quad i = 1, \ldots, N.$$

end for

Notice that even when resampling (or *selection*) does occur, estimation should be carried out using the weighted particles, i.e. with $\sum_{i=1}^{N} \omega_t^{(i)} h(\tilde{x}_t^{(i)})$, since the particle representation after resampling has lower Monte Carlo error than that before resampling.

A practical issue concerning the weight normalisation is numerical precision, since weights can be extremely large or small. Thus weights are typically stored on a log-scale and updated by addition of the log-incremental weight to the previous log-weight. The normalisation step can still fail, however, owing to numerical overflow or underflow. A simple solution involves subtracting the largest log-weight value at each time t from all log-weights, and then performing normalisation using these adjusted log-weights. This ensures that the largest (most important) weights are easily computable within machine accuracy, while very small weights (which are unimportant in any case) may be set to zero by underflow.

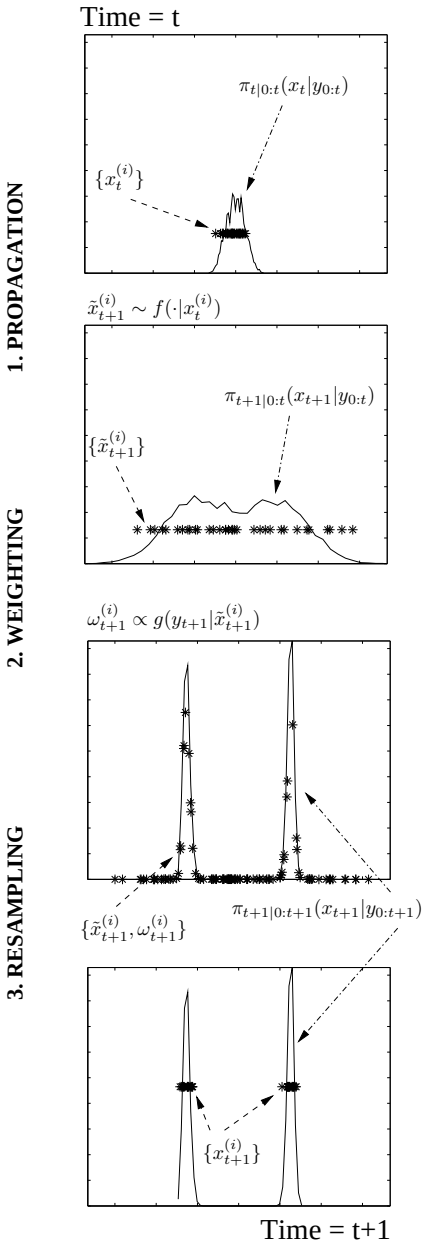

Fig. 5. The bootstrap filter in operation from time t to $t+1$, nonlinear time series Example 1. Asterisks show the positions of (a small selection of) the particles at each stage. The solid line shows a kernel density estimate of the distributions represented at each stage. 10,000 particles were used in total. Notice that resampling concentrates particles into the region of high probability.

C. The bootstrap filter (after [3])

The bootstrap filter proposed by [3] uses the state transition density f, or "prior kernel" as importance distribution. The importance weight then simplifies to

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} g(y_t|x_t^{(i)}) \ .$$

In the original version of the algorithm resampling is carried out at each and every time step, in which case the term $\omega_{t-1}^{(i)} = 1/N$ is a constant, which may thus be ignored. In more sophisticated schemes, resampling is only carried out when the distribution of the importance weights becomes degenerate, which can be measured by monitoring the changes of the coefficient of variation or the entropy of the weight pattern over time [22], [36], [23]

A distinctive feature of the bootstrap filter is that the incremental weight *does not depend on* the past trajectory of the particles but only on the likelihood of the observation, $g(y_t|x_t)$. The use of the prior kernel is popular because sampling is often straightforward, and computing the incremental weight simply amounts to evaluating the conditional likelihood of the new observation given the updated particle position.

A diagrammatic representation of the bootstrap filter in operation is given in Fig. 5, in which the resampling (selection) step is seen to concentrate particles (asterisks) into the two high probability modes of the density function.

D. How to build better proposals?

Despite its appealing properties, the use of the state transition density f as importance distribution can often lead to poor performance, which is manifested in a lack of robustness with respect to the values taken by the observed sequence, for example when outliers occur in the data (the observation is not informative) or on the contrary when the variance of the observation noise is small (the observation is very informative). This results from a mismatch between the prior predictive distribution and the posterior distribution of the state conditioned on the new measurement. In order to reduce this mismatch a natural option is to propose the new particle position under the following distribution

$$q_t(x_t|x_{t-1}, y_t) = \frac{f(x_t|x_{t-1})g(y_t|x_t)}{\int f(x|x_{t-1})g(y_t|x)dx} \ , \qquad (11)$$

which may be recognised as the conditional distribution of the hidden state x_t given x_{t-1} *and* the current observation y_t. The normalisation constant can be seen to equal the predictive distribution of y_t conditional on x_{t-1}, i.e. $p(y_t|x_{t-1})$. In the sequel, we will refer to this kernel as the *optimal kernel*, following the terminology found in the sequential importance sampling literature. This terminology dates back probably to [37], [38] and is largely adopted by authors such as [18], [39], [23], [20], [26]. The optimal property of this kernel is that the conditional variance of the weights is zero, given the past history of the particles:

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} p(y_t|x_{t-1}) = \omega_{t-1}^{(i)} \int f(x|x_{t-1}^{(i)}) g(y_t|x)dx \ . \qquad (12)$$

The incremental weight above depends only on the previous position of the particle and the new observation. This is the opposite of the situation observed previously for the prior kernel, which depended only upon the new proposed state and the observation. The optimal kernel (11) is attractive because it incorporates information both on the state dynamics and on the current observation: the particles move "blindly" with the prior kernel, whereas when using the optimal kernel the particles tend to cluster in regions where the conditional posterior distribution for the current state has high probability mass. While the optimal kernel is intuitively appealing, and also satisfies an optimality criterion of some sort, it should be noted that it is possible to sample directly from such a kernel and to evaluate the weight integral analytically only in specific classes of model.

Since the optimal kernel is itself intractable in most cases, much effort has been expended in attempting to approximate the optimal kernel. One principal means to achieve this is local linearisation and Gaussian approximation, using techniques inspired by standard nonlinear filter methodology. Here, however, linearisation is carried out *per particle*, and a proper weighting is computed in order to correct for the approximations introduced. Hence standard methodology may be leveraged to good effect without losing the asymptotic consistency of the particle representation. These techniques are developed and extensively reviewed in [23], [24], [6], [7], [12], [40].

E. Auxiliary Sampling

We now consider a more profound revision of the principles exposed so far. Let us first remark that as (8) is valid for any function h, it defines a *weighted empirical distribution on the path space* which we will denote by $\hat{\pi}_{0:t|0:t}$, equivalently writing

$$\hat{\pi}_{0:t|0:t}(dx_{0:t}) = \sum_{i=1}^{N} \omega_t^{(i)} \delta_{x_{0:t}^{(i)}}(dx_{0:t}) \ , \qquad (13)$$

where the notation $\delta_{x_{0:t}^{(i)}}$ denotes the Dirac mass at point $x_{0:t}^{(i)}$. Under suitable technical assumptions, the weighted empirical distribution $\hat{\pi}_{0:t|0:t}$ is a consistent approximation to $\pi_{0:t|0:t}$, i.e. for any function h on the path space

$$\hat{\pi}_{0:t|0:t}(h) \stackrel{\text{def}}{=} \sum_{i=1}^{N} \omega_t^{(i)} h(x_{0:t}^{(i)}) \ ,$$

converges to $\pi_{0:t|0:t}(h)$ as the number N of particles increases to infinity. The degree to which this assertion is correct is discussed in [41], [42], [43], [44], [32], [7] but we only need to know at this stage that the general intuition is justifiable.

The previous methods were aimed principally at improving the proposal distribution for the new states at time t. However, it was realised that further improvements could be achieved by replacing the standard resampling schemes by more sophisticated algorithms. These attempt to favour particles which are more likely to survive *at the next time step*. Such schemes introduce a bias into the filtering density representation which is corrected by associating with surviving

particles an appropriately modified weight. The first exponents of these ideas were probably Pitt and Shephard [45], and the ideas link in closely with the biased sampling approaches proposed by [46].

The formulation given here is equivalent to that given Pitt and Shephard, although we avoid the explicit inclusion of an auxiliary indexing variable by considering a proposal over the entire path of the process up to time t. The starting assumption is that the joint posterior at $t-1$ is well approximated by $\hat{\pi}_{0:t-1|0:t-1}$. Based on this assumption the joint posterior distribution at time t is approximated as

$$\pi_{0:t|0:t}(dx_{0:t}|y_{0:t}) \approx \\ \times \frac{1}{Z}\sum_{i=1}^{N}\omega_{t-1}^{(i)}\delta_{x_{0:t-1}^{(i)}}(dx_{0:t-1})g(y_t|x_t)f(x_t|x_{t-1}^{(i)})dx_t \ , \quad (14)$$

where the normalisation factor Z is given by

$$Z = \sum_{j=1}^{N}\omega_{t-1}^{(j)}\int f(x|x_{t-1}^{(j)})g(y_t|x)dx \ .$$

Now, in exactly the same way as we interpreted (9) as a joint proposal for all states of indices 0 to t, we now consider a general joint proposal for the entire path of the new particles $x_{0:t}^{(i)}$, that is,

$$\begin{aligned} q_t(dx_{0:t}) &= q_{0:t-1}(dx_{0:t-1}|y_{0:t})q_t(dx_t|x_{t-1},y_t) \\ &= \left(\sum_{i=1}^{N}v_{t-1}^{(i)}\delta_{x_{0:t-1}^{(i)}}(dx_{0:t-1})\right) \\ &\quad \times \left(q_t(dx_t|x_{t-1}^{(i)},y_t)\right) \ , \end{aligned}$$

where $\sum_{i=1}^{N}v_{t-1}^{(i)}=1$ and $v_{t-1}^{(i)}>0$. Notice that as before the proposal splits into two parts: a marginal proposal $q_{0:t-1}$ for the past path of the process $x_{0:t-1}$ and a conditional proposal q_t for the new state x_t. Note that the first component is constructed to depend explicitly on data up to time t in order to allow adaptation of the proposal in the light of the new data point y_t (and indeed it may depend on future data points as well if some look-ahead and latency is allowable). The first part of the proposal is a discrete distribution centred upon the old particle paths $\{x_{0:t-1}^{(i)}\}$, but now with probability mass for each component in the proposal distribution designed to be $\{v_{t-1}^{(i)}\}$. The weighting function $v_{t-1}^{(i)}$ can be data dependent, the rationale being that we should preselect particles that are a good fit to the new data point y_t. For example, Pitt and Shephard [45] suggest taking a point value $\mu^{(i)}$ of the state, say the mean or mode of $f(x_t|x_{t-1}^{(i)})$, and computing the weighting function as the likelihood evaluated at this point, i.e. $v_{t-1}^{(i)}=g(y_t|\mu_t^{(i)})$; or if the particles from $t-1$ are weighted, one would choose $v_{t-1}^{(i)}=\omega_{t-1}^{(i)}g(y_t|\mu_t^{(i)})$. The rationale for this is as follows. Ideally, for filtering at time t, one would wish to propose the past paths $x_{0:t-1}$ from their marginal conditional distribution $\pi_{0:t-1|0:t}$. This can be written out and expanded using the particle approximation from $t-1$ as:

$$\begin{aligned} &\pi_{0:t-1|0:t}(dx_{0:t-1}|y_{0:t}) \\ &\quad \propto \int \pi_{0:t-1|0:t-1}(dx_{0:t-1}|y_{0:t-1})f(x_t|x_{t-1})g(y_t|x_t)dx_t \\ &\quad \approx \sum_{i=1}^{N}\omega_{t-1}^{(i)}\delta_{x_{0:t-1}^{(i)}}(dx_{0:t-1})\int f(x_t|x_{t-1}^{(i)})g(y_t|x_t)dx_t \ . \end{aligned}$$

This integral may be approximated by any means available, including Monte Carlo. In [45], it is suggested to use the crude approximation $f(dx_t|x_{t-1}^{(i)}) \approx \delta_{\mu_t^{(i)}}(dx_t)$, in which case we have

$$\pi_{0:t-1|0:t}(dx_{0:t-1}|y_{0:t}) \approx \sum_{i=1}^{N}g(y_t|\mu_t^{(i)})\omega_{t-1}^{(i)}\delta_{x_{0:t-1}^{(i)}}(dx_{0:t-1}) \ ,$$

and hence the choice $v_{t-1}^{(i)}=g(y_t|\mu_t^{(i)})\omega_{t-1}^{(i)}$. Other biasing schemes based upon an unscented approximation to the integral can be found in [47], or on exploration of future data points in [46].

Using this proposal mechanism it is then possible to define a generalised importance ratio (in the Radon-Nikodym sense) between the approximate posterior in (14) and the full path proposal q, given by

$$\tilde{\omega}_t^{(i)} = \frac{\omega_{t-1}^{(i)}}{v_{t-1}^{(i)}} \times \frac{g(y_t|x_t^{(i)})f(x_t^{(i)}|x_{t-1}^{(i)})}{q_t(x_t^{(i)}|x_{t-1}^{(i)},y_t)} \ .$$

Note that compared to the standard SIS sampler we have had to account for the bias introduced in the sampler by a correction to the importance weight equal to $1/v_{t-1}^{(i)}$; the ratio $\omega_{t-1}^{(i)}/v_{t-1}^{(i)}$ is known as the first stage weight. Note that in the original scheme a resampling stage was added to the first stage selection; however, this is unnecessary and introduces further Monte Carlo error into the filter. More general schemes that allow some exploration of future data points by so-called pilot sampling to generate the weighting function have been proposed in, for example [46], while further discussion of the framework can be found in [48]. A summary of the auxiliary particle filter is given in Algorithm 4. We assume that the selection step occurs at each point, although it may be omitted exactly as in the standard particle filter, in which case of course no weight correction is applied.

F. Simulation Example

We now provide brief simulation results for the particle filter, using Example 1, the nonlinear time series model. This is presented purely as an example of the type of results obtainable and their interpretation: others have provided extensive simulation studies in this type of model. A single data set is generated from the model, see Fig. 6. The full particle filter (see Algorithm 3) is run on this data. The prior importance function f is used, and resampling occurs at every time step — this is then the bootstrap version of the particle filter. The number of particles used is fixed over time to $N=10,000$, a large number that may be reduced substantially in practice, depending on the accuracy of inference required. Figs. 7–8 show two time snapshots of the filter output, i.e. estimates of

Algorithm 4 Auxiliary Particle Filter

for $i = 1, \ldots, N$ **do** ▷ Initialisation

Sample $\tilde{x}_0^{(i)} \sim q_0(x_0|y_0)$.

Assign initial importance weights

$$\tilde{\omega}_0^{(i)} = \frac{g(y_0|\tilde{x}_0^{(i)})\pi_0(\tilde{x}_0^{(i)})}{q_0(\tilde{x}_0^{(i)}|y_0)} .$$

end for

for $t = 1, \ldots, T$ **do**

Select N particle indices $j_i \in \{1, \ldots, N\}$ according to weights

$$\{v_{t-1}^{(i)}\}_{1 \le i \le N} .$$

for $i = 1, \ldots, N$ **do**

Set $x_{t-1}^{(i)} = \tilde{x}_{t-1}^{(j_i)}$.

Set first stage weights:

$$u_{t-1}^{(i)} = \frac{\omega_{t-1}^{(j_i)}}{v_{t-1}^{(j_i)}} .$$

end for

for $i = 1, \ldots, N$ **do**

Propagate:

$$\tilde{x}_t^{(i)} \sim q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, y_t) .$$

Compute weight:

$$\tilde{\omega}_t^{(i)} = u_{t-1}^{(i)} \frac{g(y_t|\tilde{x}_t^{(i)})f(\tilde{x}_t^{(i)}|x_{t-1}^{(i)})}{q_t(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, y_t)} .$$

end for

Normalise weights:

$$\omega_t^{(i)} = \tilde{\omega}_t^{(i)} / \sum_{j=1}^{N} \tilde{\omega}_t^{(j)} , \quad i = 1, \ldots, N.$$

end for

$\pi_{t|0:t}$. In these we plot the particle weights (unnormalised) against raw particle values as small dots, i.e we plot the set of $\{\tilde{x}_t^{(i)}, \omega_t^{(i)}\}$ points - note that the dots merge almost into a continuous line in some places as there are so many particles covering important regions. As a dashed line we plot a kernel density estimate obtained from the weighted sample, using a Gaussian kernel having fixed width of 0.5. Notice that the filter is easily able to track multimodality in the distributions over time. Notice also that the highest weighted particles are not necessarily the most probable state estimates: the kernel density estimator places the maximum of the filtering density wherever the weights *and* the local density of particles combine to give the highest probability density. This is an elementary point which is often overlooked by practitioners starting in the field. Finally, to give the whole picture, the kernel density estimates over time are compiled into an intensity image to show the evolution with time of the densities, see Fig. 9. As a comparison we have run the SIS algorithm, i.e. with no resampling incorporated, as in Algorithm 2, under otherwise identical conditions. As expected, this is unable to track the correct state sequence and the particle distributions are highly degenerate, i.e. resampling is an essential ingredient in this type of model - see Fig. 10.

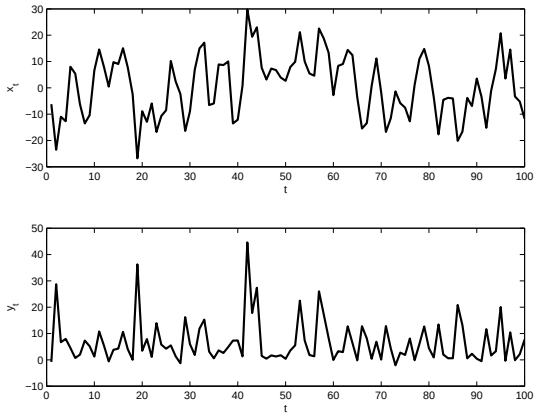

Fig. 6. Data set drawn from the nonlinear time series model of Example 1

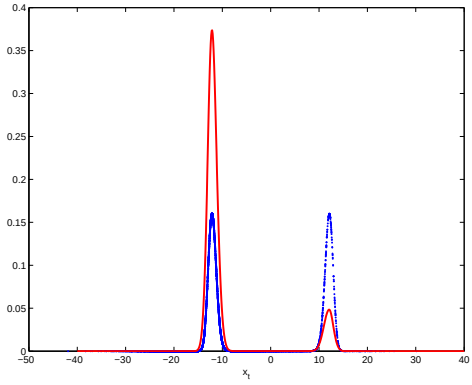

Fig. 7. Particle filter output, $t = 40$. Weighted samples $\{\tilde{x}_{40}^{(i)}, \omega_{40}^{(i)}\}$ (shown as small dots - almost continuous line) and kernel density estimate (dashed)

G. Marginalised Particle Filters

In many practical scenarios, especially those found in the tracking domain, the models are not entirely nonlinear and non-Gaussian. By this we mean that some subset of the state vector is linear and Gaussian, *conditional upon* the other states. In these cases one may use standard linear Gaussian optimal filtering for the linear part, and particle filtering for the nonlinear part. This may be thought of as an optimal Gaussian mixture approximation to the filtering distribution. See [23], [39], [49] for detailed descriptions of this approach to the problem, which is referred to either as the *Rao-Blackwellised* particle filter, or *Mixture Kalman* filter. Recent work [50], [51] has studied in detail the possible classes of model that may be handled by the marginalised filter, and computational complexity issues. The formulation is as follows[1]. First, the state is partitioned into two components, x_t^L and x_t^N, referring respectively to the linear ('L') and nonlinear ('N') components.

[1] [50], [51] present a more general class of models to which the marginalised filter may be applied, but we present a more basic framework for the sake of simplicity here.

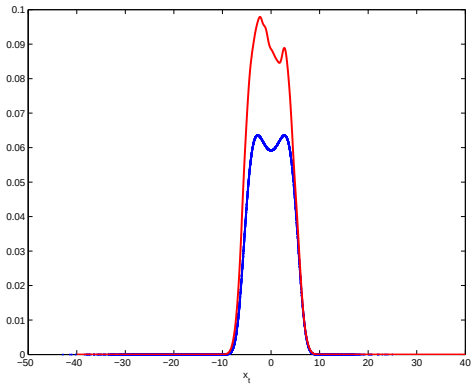

Fig. 8. Particle filter output, $t = 50$. Weighted samples $\{\tilde{x}_{50}^{(i)}, \omega_{50}^{(i)}\}$ (shown as small dots - almost continuous line) and kernel density estimate (dashed)

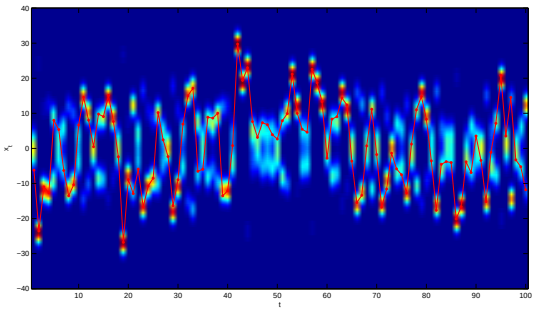

Fig. 9. Full particle filter density output (shown as image intensity plot of kernel density estimates). True state sequence overlaid (solid line with asterisk markers)

The linear part of the model is expressed in the form of a linear Gaussian state-space model as follows, with state-space matrices that may depend upon the nonlinear state x_t^N:

$$x_t^L = A(x_t^N)x_{t-1}^L + u_t^L \ , \tag{15}$$

$$y_t = B(x_t^N)x_t^L + v_t^L \ . \tag{16}$$

Here u_t^L and v_t^L are independent, zero-mean, Gaussian disturbances with covariances C_u and C_v, respectively, and $A()$ and $B()$ are matrices of compatible dimensions that may depend upon the nonlinear state x_t^N. At $t = 0$, the linear part of the model is initialised with $x_0^L \sim \mathcal{N}(\mu_0(x_0^N), P_0(x_0^N))$.

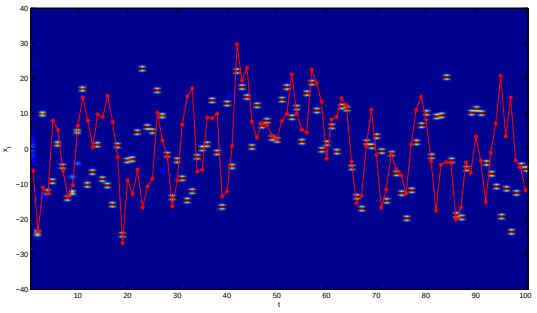

Fig. 10. Full Sequential importance sampling (no resampling) filter density output (shown as image intensity plot of kernel density estimates). True state sequence overlaid (solid line with asterisk markers)

Now the nonlinear part of the state obeys a general dynamical model (which is not necessarily Markovian):

$$x_t^N \sim f(x_t^N | x_{0:t-1}^N), \ \ x_0^N \sim \pi_0(x_0^N) \ . \tag{17}$$

In such a case, conditioning on the nonlinear part of the state $x_{0:t}^N$ and the observations $y_{0:t}$, the linear part of the state is jointly Gaussian and the means and covariances of this Gaussian representation may be obtained by using the classical Kalman filtering recursions [52]. The basic idea is then to *marginalise* the linear part of the state vector to obtain the posterior distribution of the nonlinear part of the state:

$$\pi_{0:t|0:t}(x_{0:t}^N | y_{0:t}) = \int \pi_{0:t|0:t}(x_{0:t}^L, x_{0:t}^N | y_{0:t}) dx_{0:t}^L \ .$$

Particle filtering is then run on the nonlinear state sequence only, with target distribution $\pi_{0:t|0:t}(x_{0:t}^N | y_{0:t})$. The resulting algorithm is almost exactly as before, requiring only a slight modification to the basic particle filter (Algorithm 3) to allow for the fact that the marginalised system is no longer Markovian, since

$$p(y_t | y_{0:t-1}, x_{0:t}^N) \neq p(y_t | x_t^N) \ .$$

Moreover, the dynamical model for the nonlinear part of the state may itself be non-Markovian, see Eq. (17).

Thus, instead of the usual updating rule we have:

- *Prediction*

$$\pi_{0:t|0:t-1}(x_{0:t}^N | y_{0:t-1}) = \pi_{0:t-1|0:t-1}(x_{0:t-1}^N | y_{0:t-1}) f(x_t^N | x_{0:t-1}^N) \ . \tag{18}$$

- *Correction*

$$\pi_{0:t|0:t}(x_{0:t}^N | y_{0:t}) = \frac{p(y_t | y_{0:t-1}, x_{0:t}^N) \pi_{0:t|0:t-1}(x_{0:t}^N | y_{0:t-1})}{\ell_{t|0:t-1}(y_t | y_{0:t-1})} \ , \tag{19}$$

  where as before $\ell_{t|0:t-1}$ is the predictive distribution of y_t given the past observations $y_{0:t-1}$, which is a fixed normalising constant (independent of the state sequence $x_{0:t}^N$).

Note that if $\{(x_{0:t}^{N,(i)}, \omega_t^{(i)})\}_{i=1,\ldots,N}$ denote the particles evolving in the state-space of the nonlinear variables according to the above equations, and their associated importance weights, estimation of the linear part of the state may be done using a *Rao-Blackwellised* estimation scheme [53]: the posterior density for the linear part is obtained as a random Gaussian mixture approximation given by

$$\pi_{t|0:t}(x_t^L | y_{0:t}) \approx \sum_{i=1}^{N} \omega_t^{(i)} p(x_t^L | x_{0:t}^{N,(i)}, y_{0:t}) \ , \tag{20}$$

where the conditional densities $p(x_t^L | x_{0:t}^{N,(i)}, y_{0:t})$ are Gaussian and computed again using Kalman filtering recursions. Eq. (20) replaces the standard point-mass approximation (13) arising in the generic particle filter. The Rao-Blackwellised estimate is usually better in terms of Monte Carlo error than the corresponding scheme that performs standard particle filtering jointly in both nonlinear and linear states. The

computational trade-off is more complex, however, since the marginalised filter can be significantly more time-consuming than the standard filter *per particle*. These trade-offs have been extensively studied by [51] and in many cases the performance/computation trade-off comes out in favour of the marginalised filter.

To give further detail to the approach, we first summarise the Kalman filter itself in this probabilistic setting [34], then we place the whole scheme back in the particle filtering context. As a starting point, assume the distribution $p(x_{t-1}^L|y_{0:t-1}, x_{0:t-1}^N)$ has been obtained. This is a Gaussian, denoted by

$$p(x_{t-1}^L|y_{0:t-1}, x_{0:t-1}^N) = \mathcal{N}(x_{t-1}^L|\mu_{t-1|0:t-1}, C_{t-1|0:t-1}) \ ,$$

where the mean and covariance terms are dependent upon both $y_{0:t-1}$ and $x_{0:t-1}^N$. Now, (15) shows how to update this distribution one step, since x_t^L is just a summation of two transformed independent Gaussian random vectors, $A(x_t^N)x_{t-1}^L$ and u_t^L, which itself must be a Gaussian. Under the standard rules for summation of independent Gaussian random vectors, we obtain the predictive distribution for x_t^L, conditioned upon $y_{0:t-1}$ and $x_{0:t}^N$, as follows:

$$p(x_t^L|y_{0:t-1}, x_{0:t}^N) = \mathcal{N}(x_t^L|\mu_{t|0:t-1}, C_{t|0:t-1}) \ , \tag{21}$$

where

$$\begin{aligned} \mu_{t|0:t-1} &= A(x_t^N)\mu_{t-1|0:t-1} \ , \\ C_{t|0:t-1} &= A(x_t^N)C_{t-1|0:t-1}A(x_t^N)^T + C_u \ . \end{aligned}$$

As a second step in the update, the new data point y_t is incorporated through Bayes' Theorem:

$$\begin{aligned} p(x_t^L|y_{0:t}, x_{0:t}^N) &= \frac{p(x_t^L|y_{0:t-1}, x_{0:t}^N) \times p(y_t|x_t^L, x_t^N)}{p(y_t|y_{0:t-1}, x_{0:t}^N)} \\ &\propto \mathcal{N}(x_t^L|\mu_{t|0:t-1}, C_{t|0:t-1}) \times \mathcal{N}(y_t|B(x_t^N)x_t^L, C_v) \\ &= \mathcal{N}(x_t^L|\mu_{t|0:t}, C_{t|0:t}) \end{aligned} \tag{22}$$

where $\mu_{t|0:t}$ and $C_{t|0:t}$ are obtained by standard rearrangement formulae as

$$\begin{aligned} \mu_{t|0:t} &= \mu_{t|0:t-1} + K_t(y_t - B(x_t^N)\mu_{t|0:t-1}) \ , \\ C_{t|0:t} &= (I - K_t B(x_t^N))C_{t|0:t-1} \ , \\ K_t &= C_{t|0:t-1}B^T(x_t^N)(B(x_t^N)C_{t|0:t-1}B^T(x_t^N) + C_v)^{-1} \ , \end{aligned}$$

and where the term K_t is known as the *Kalman Gain*. In order to complete the analysis for particle filter use, one further term is required, $p(y_t|y_{0:t-1}, x_{0:t}^N)$. This is obtained by the so-called *prediction error decomposition*, which is easily obtained from (21), since y_t is obtained by summing a transformed version of x_t^L, i.e. $B(x_t^N)x_t^L$, with an independent zero-mean Gaussian noise term v_t^L having covariance C_v, leading to:

$$p(y_t|y_{0:t-1}, x_{0:t}^N) = \mathcal{N}(y_t|\mu_{y_t}, C_{y_t}) \ , \tag{23}$$

where

$$\begin{aligned} \mu_{y_t} &= B(x_t^N)\mu_{t|0:t-1} \ , \\ C_{y_t} &= B(x_t^N)C_{t|0:t-1}B^T(x_t^N) + C_v \ . \end{aligned}$$

In order to construct the marginalised particle filter, notice that for any realisation of the nonlinear state sequence $x_{0:t}^N$, and data sequence $y_{0:t}$, one may calculate the value of $p(y_t|y_{0:t-1}, x_{0:t}^N)$ in (23) through sequential application of the formulae (21) and (22). The marginalised particle filter then requires computation and storage of the term $p(y_t|y_{0:t-1}, x_{0:t}^{N,(i)})$ in (23), for *each* particle realisation $x_{0:t}^{N,(i)}$. In the marginalised particle filter the particles are stored as the nonlinear part of the state x_t^N, the associated sufficient statistics for each particle, i.e. $\mu_{t|0:t}$ and $C_{t|0:t}$, and the weight for each particle. We do not give the entire modified algorithm. The only significant change is to the weighting step, which becomes

$$\tilde{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \frac{p(y_t|y_{0:t-1}, \tilde{x}_t^{N,(i)}) f(\tilde{x}_t^{N,(i)}|x_{0:t-1}^{N,(i)})}{q_t(\tilde{x}_t^{N,(i)}|x_{0:t-1}^{N,(i)}, y_{0:t})}$$

As an important aside, we note that the marginalised filter may also be used to good effect when the linear states are unknown but 'static' over time, i.e. $f(dx_t^L|x_{t-1}^L) = \delta_{x_{t-1}^L}(dx_t^L)$ with some Gaussian initial distribution or prior $x_0^L \sim \mathcal{N}(\mu_0(x_0^N), P_0(x_0^N))$, as before. Then the marginalised filter runs exactly as before but we are now able to marginalise, or infer the value of, a static parameter $\theta = x_t^L$. Early versions of such filters are found in the sequential imputations work of [18], for example. This issue is explored more fully, including an example, in the context of other parameter estimation schemes in Section IV.

We have focused here on the linear Gaussian case of the marginalised filter. However, another important class of models is the discrete state-space Hidden Markov model, in which the states are discrete values and switching may occur between one time and the next according to a Markov transition matrix. As for the linear Gaussian case, the discrete state values may be marginalised to form a marginalised particle filter, using the HMM forward algorithm [54] instead of the Kalman filter [23]. For simulations and examples within both frameworks, see [23], [7].

As mentioned before, several generalisations are possible to the basic model. The most basic of these allow the matrices $A()$, $B()$, C_u and C_v to depend on time and on any or all elements of the nonlinear state sequence $x_{0:t}^N$. None of these changes require any modification to the algorithm formulation. Another useful case allows a deterministic function of the nonlinear states to be present in the observation and dynamical equations. These two features combined lead to the following form:

$$\begin{aligned} x_t^L &= A_t(x_{0:t}^N)x_{t-1}^L + c(x_{0:t}^N) + u_t^L \ , \\ y_t &= B_t(x_{0:t}^N)x_t^L + d(x_{0:t}^N) + v_t^L \ , \end{aligned}$$

and again the form of the algorithm is unchanged; see [55] for a good coverage of the most general form of Kalman filters required in these cases.

One other important case involves nonlinear observations that are not a function of the linear state. Then the linear observation equation (16) can be generalised to $y_t \sim g(y_t|x_t^N)$, which is a general observation density. This form is quite useful in tracking examples, where observation functions are often nonlinear (range and bearings, for example, or range-only), but dynamics can be considered as linear to a good approximation [49], [50], [51]. If in addition the nonlinear

state can be expressed in linear Gaussian state-space form with respect to the linear state, i.e:

$$x_t^N = B(x_t^N)x_t^L + c(x_{t-1}^N) + v_t^L \ ,$$
$$x_t^L = A(x_t^N)x_{t-1}^L + u_t^L \ ,$$

then once again the Kalman filter can be run to marginalise the linear state variable. In this case the weight expression becomes:

$$\tilde{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \frac{g(y_t|\tilde{x}_t^{N,(i)})p(\tilde{x}_t^{N,(i)}|x_{0:t-1}^{N,(i)})}{q_t(\tilde{x}_t^{N,(i)}|x_{0:t-1}^{N,(i)}, y_{0:t})} \ ,$$

where now the term $p(\tilde{x}_t^{N,(i)}|x_{0:t-1}^{N,(i)})$ is computed using the Kalman filter. In some cases the linear state transition matrices and observation matrices $A()$ and $B()$ for this Kalman filter are independent of the nonlinear state and the observations; then this form of marginalised particle filter may be computed very efficiently, since the covariance matrices are identical for all particles and thus need only be computed once at each time step.

H. MCMC Adaptations

Another area where improvements can be made over the basic methods is in Markov chain Monte Carlo (MCMC) techniques. The general approach is that one would design an MCMC transition kernel, such as a Gibbs sampler or Metropolis Hastings scheme [53], having $\pi_{0:t|0:t}(x_{0:t}|y_{0:t})$ as its stationary distribution. The MCMC transition kernel is then applied one or more times to each particle $x_{0:t}^{(i)}$, either before or after the resampling step, the intuition being that an adequate particle filter representation at time t can only be improved with the application of MCMC moves having the correct stationary distribution; see especially the *resample-move* procedures [56], [57], [58], [48], [59], and also more recent works on incorporation of MCMC into particle filtering in [60]. MCMC schemes are found to be particularly effective in fixed-lag smoothing approaches and in static parameter estimation, as further discussed in Sections III and IV below. To give a concrete example, consider a fixed-lag smoothing approach with MCMC moves (see also [61]). In this case one designs an MCMC kernel having the fixed-lag conditional distribution as its stationary distribution:

$$\pi_{0:t|0:t}(x_{t-L+1:t}|x_{0:t-L}, y_{0:t}) \ , \quad L > 0 \ .$$

Such a kernel also has by construction the joint posterior as its stationary distribution [62], as required. As a simple example that will work for many models, consider a Gibbs sampling implementation that samples states within the fixed lag window one by one from their conditional distribution:

$$x_{t-l}^{(i)} \sim \pi_{0:t|0:t}(x_{t-l}|x_{0:t\setminus(t-l)}^{(i)}, y_{0:t}) \ , \quad l \in \{0, \ldots, L-1\}.$$

where $x_{0:t\setminus j}$ denotes all elements of $x_{0:t}$ except for element l. Such moves are applied successively, with replacement, to all particles in the current set $i \in \{1, \ldots, N\}$, and for all lags $l \in \{0, \ldots, L-1\}$, for as many iterations as required (usually dictated by the available computing resource). In cases where a random draw cannot be made directly from the conditional distributions, it will be necessary to split the state x_t into smaller sub-components [48], or to apply Metropolis-Hastings moves instead of Gibbs sampling [63], [64]. A common misconception with this type of scheme is that a full MCMC-style *burn-in* period[2] is required for each time step and for each particle. This is not the case, since we are initialising nominally from a 'converged' particle set, and so any MCMC moves will remain converged and require no burn-in (although in practice the schemes are often adopted to improve on a poor particle representation and to introduce variability between replicated particles following the selection step). Note that we have not intended to give a tutorial in this review on general MCMC methods, which are a whole research discipline in themselves, and for this the reader is referred on to the textbooks [65], [53], [7].

III. Smoothing

In this section we review methodology for Monte Carlo smoothing based upon particle filters. We note that smoothing is particularly relevant in complex dynamical systems, since filtering alone will often yield only fairly uninformative state estimates, while the 'lookahead' allowed by smoothing enables much more accurate estimates to be achieved retrospectively.

The first thing to notice is that the basic 'filtering' version of the particle filter (7) actually provides us with an approximation of the joint smoothing distribution at no extra cost, since the equations are defined for the whole path of the process from time 0 up to time t. Thus the stored particle trajectories $\{x_{0:t}^{(i)}\}$ and their associated weights $\{\omega_t^{(i)}\}$ can be considered as a weighted sample from the joint *smoothing* distribution $\pi_{0:t|0:t}(x_{0:t}|y_{0:t})$. From these joint draws one may readily obtain fixed lag or fixed interval smoothed samples by simply extracting the required components from the sampled particles and retaining the same weights; for example, if $\{(x_{0:t}^{(i)}, \omega_t^{(i)})\}$ is a weighted approximation to $\pi_{0:t|0:t}(x_{0:t}|y_{0:t})$ then it automatically follows that, for some smoothing lag L, $\{(x_{t-L}^{(i)}, \omega_t^{(i)})\}$ is a weighted approximation to $\pi_{t-L|0:t}(x_{t-L}|y_{0:t})$. Similarly, if we are interested in studying dependencies over time of state variables these can be obtained by extracting sub-sequences from the path particle representation, e.g. for $M > L$, $\{(x_{t-M+1:t-L}^{(i)}, \omega_t^{(i)})\}$ is a weighted approximation to $\pi_{t-M+1:t-L|0:t}(x_{t-M+1:t-L}|y_{0:t})$, where in this case we are interested in a smoothed subsequence of length $M - L$ extracted from the state sequence.

While these appealingly simple schemes can be successful for certain models and small lags L and M, it rapidly becomes apparent that resampling procedures will make this a very depleted and potentially inaccurate representation of the required smoothing distributions. This situation is schematically represented on Figure 11 which shows that while the diversity of the particles is satisfactory for the current time index, successive resamplings imply that for time-lags that are back in the past, the number of particle positions that are indeed

[2] In usual MCMC applications, the initial iterations of the chain are most often discarded in an attempt to reduce the bias caused by the fact that the chain is started from an arbitrary point (rather than from a point drawn from the stationary distribution).

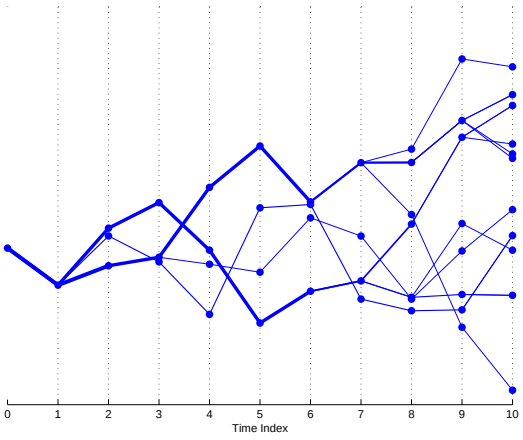

Fig. 11. Typical plot of the particle trajectories after a few time steps; the width of the lines is proportional to the number of current particles which share a particular ancestor path.

different decreases, eventually reaching a point where all current particles share a common ancestor.

There are various ways in which one can improve upon the performance of the basic scheme. We first consider the use of backward smoothing recursions, which can be thought of as the natural extension of the Kalman backward smoothing recursions to nonlinear and non-Gaussian state-space models.

We first note that the joint smoothing distribution may be factorised as follows

$$\pi_{0:T|0:T}(x_{0:T}|y_{0:T}) = \pi_{T|0:T}(x_T|y_{0:T}) \prod_{t=0}^{T-1} p(x_t|x_{t+1:T}, y_{0:T}) \tag{24}$$

$$= \pi_{T|0:T}(x_T|y_{0:T}) \prod_{t=0}^{T-1} p(x_t|x_{t+1}, y_{0:t}) \tag{25}$$

where the term in the product can be expressed as

$$p(x_t|x_{t+1}, y_{0:T}) = \frac{\pi_{t|0:t}(x_t|y_{0:t}) f(x_{t+1}|x_t)}{\int \pi_{t|0:t}(x_t|y_{0:t}) f(x_{t+1}|x_t) dx_t} \tag{26}$$

$$\propto \pi_{t|0:t}(x_t|y_{0:t}) f(x_{t+1}|x_t) \ . \tag{27}$$

These formulae then form the basis of a sequence-based smoother using the weighted sample generated in the forward pass of the SMC procedure, see [66], and also [67], [32].

Assume initially that Monte Carlo filtering has already been performed on the entire dataset, leading to an approximate representation of the filtering distribution $\pi_{t|0:t}(x_t|y_{0:t})$ for each time step $t \in \{0, \ldots, T\}$, consisting of weighted particles $\{(x_t^{(i)}, \omega_t^{(i)})\}_{i=1,\ldots,N}$.

Using this weighted sample representation, it is straightforward to construct a particle approximation to $p(x_t|x_{t+1}, y_{0:T})$ from (27) as follows:

$$p(dx_t|x_{t+1}, y_{0:T}) \approx \sum_{i=1}^{N} \rho_t^{(i)}(x_{t+1}) \delta_{x_t^{(i)}}(dx_t) \ , \tag{28}$$

where the modified weights are defined as

$$\rho_t^{(i)}(x_{t+1}) \stackrel{\text{def}}{=} \frac{\omega_t^{(i)} f(x_{t+1}|x_t^{(i)})}{\sum_{j=1}^{N} \omega_t^{(j)} f(x_{t+1}|x_t^{(j)})} \ . \tag{29}$$

This revised particle-based distribution can now be used to generate states successively in the reverse-time direction, conditioning upon future states, using the sampling importance resampling idea. Specifically, given a random sample $\widetilde{x}_{t+1:T}$ drawn approximately from $\pi_{t+1:T|0:T}$, take one step back in time and sample $\widetilde{x}_t$ from the particle approximation (28) to $p(dx_t|\widetilde{x}_{t+1}, y_{0:T})$. The pair $(\widetilde{x}_t, \widetilde{x}_{t+1:T})$ is then approximately a random realization from $\pi_{t:T|0:T}$. Repeating this process sequentially back over time produces the general particle smoother outlined in Algorithm 5.

Algorithm 5 Particle Smoother

for $t = 0$ to T **do** ▷ Forward Pass Filter
Run Particle filter, storing at each time step the particles and weights $\{x_t^{(i)}, \omega_t^{(i)}\}_{1 \le i \le N}$.
end for
Choose $\widetilde{x}_T = x_T^{(i)}$ with probability $\omega_T^{(i)}$.
for $t = T-1$ to 0 **do** ▷ Backward Pass Smoother
Calculate $\rho_t^{(i)} \propto \omega_t^{(i)} f(\widetilde{x}_{t+1}|x_t^{(i)})$, for $i = 1, \ldots, N$; and normalise the modified weights.
Choose $\widetilde{x}_t = x_t^{(i)}$ with probability $\rho_t^{(i)}$.
end for

Further independent realizations are obtained by repeating this procedure as many times as required. The computational complexity for each random realisation is $O(NT)$, so the procedure is quite expensive if many realisations are required. Developments to these basic techniques that consider the Rao-Blackwellised setting can be found in [68], see Section II-G.

To illustrate this smoothing technique, consider the nonlinear time series model of Example 1. Smoothing is carried out using the above particle smoother, applying 10,000 repeated draws from the smoothing density. A simple bootstrap particle filter was run through the data initially, itself with 10,000 particles, and the weighted particles $\{(x_t^{(i)}, \omega_t^{(i)})\}_{1 \le i \le N}$ were stored at each time step, exactly as in the simulations for this model presented in the section on particle filtering. Smoothing then follows exactly as in the above algorithm statement. A small random selection of the smoothed trajectories drawn from $\pi_{0:100|0:100}(x_{0:100}|y_{0:100})$ is shown in Fig. 12. Note some clear evidence of multimodality in the smoothing distribution can be seen, as shown by the separated paths of the process around $t = 46$ and $t = 82$. We can also show the posterior distribution via grey-scale histograms of the particles, see Fig. 13. Finally, see Figs. 14 and 15 for visualisation of an estimated bivariate marginal, $\pi_{3:4|0:100}(x_{3:4}|y_{0:100})$, using 2-dimensional scatter plots and kernel density estimates, again showing evidence of multimodality and strong non-Gaussianity that will not be well captured by more standard methods.

This algorithm is quite generic in that it allows joint random draws from arbitrary groupings of state variables over time. See also [67] for related methods that generate smoothed

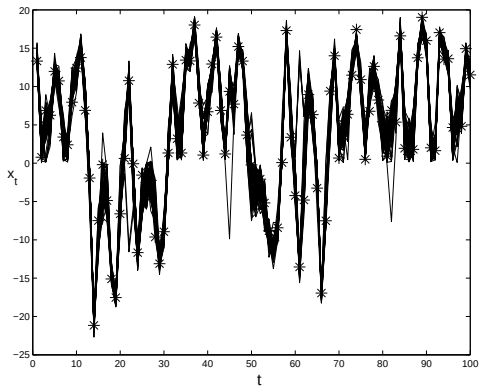

Fig. 12. Smoothing trajectories approximating $\pi_{0:100|0:100}(x_{0:100}|y_{0:100})$. True simulated states shown as '*' (from [66]).

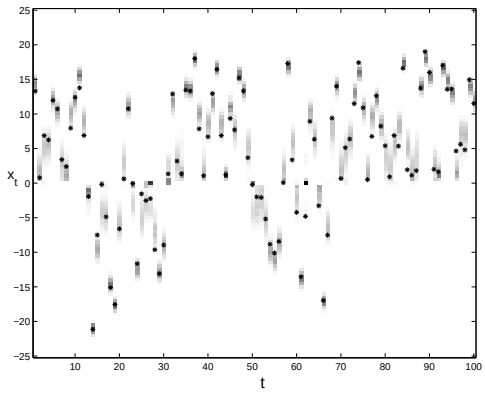

Fig. 13. Histogram estimates of smoothing densities, $\pi_{t|0:100}(x_t|y_{0:100})$, shown as gray scale intensities in vertical direction. True simulated states shown as '*' (from [66]).

sample paths by rejection sampling ideas. Sometimes however one is specifically interested in the marginal smoothing distributions, i.e. $\pi_{t|0:T}$ for some $t < T$. There are several specialised methods available for this, based on the following backward recursion (see [69]) over a fixed interval 0 to T:

$$\begin{aligned}
&\pi_{t|0:T}(x_t|y_{0:T}) \\
&= \pi_{t|0:t}(x_t|y_{0:t}) \int \frac{\pi_{t+1|0:T}(x_{t+1}|y_{0:T}) f(x_{t+1}|x_t)}{\int \pi_{t|0:t}(x|y_{0:t}) f(x_{t+1}|x) dx} dx_{t+1} \\
&= \int \pi_{t+1|0:T}(x_{t+1}|y_{0:T}) p(x_t|x_{t+1}, y_{0:T}) dx_{t+1} \ , \qquad (30)
\end{aligned}$$

where $p(x_t|x_{t+1}, y_{0:T})$ simplifies as before in (26). In [23], [70] marginal smoothing is achieved by a direct Monte Carlo implementation of (30). One recursively obtains particle estimates of the marginal smoothing distribution at the next time point, i.e. $\pi_{t+1|0:T}$ and combines these with the particle filtering estimate of $\pi_{t|0:t}$ in (30). A complication compared with the sequence-based smoothers of [66] is that one cannot in these schemes ignore the denominator term in (26), that is, $\pi_{t+1|0:t}(x_{t+1}|y_{0:t}) = \int \pi_{t|0:t}(x_t|y_{0:t}) f(x_{t+1}|x_t) dx_t$, as a normalising constant, and instead a Monte Carlo estimate must also be made for this term.

If we approximate the smoothing distribution $\pi_{t+1|0:T}$ using

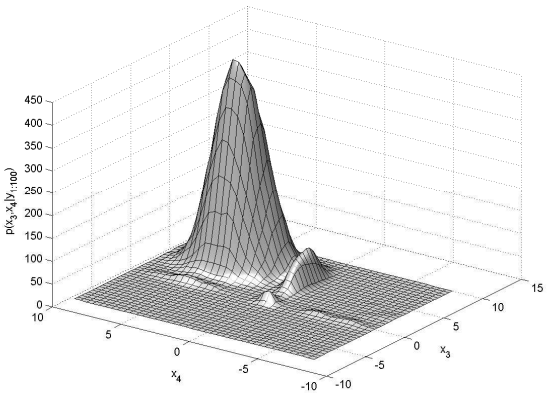

Fig. 14. Kernel density estimate for $\pi_{3:4|0:100}(x_{3:4}|y_{0:100})$ (from [66]).

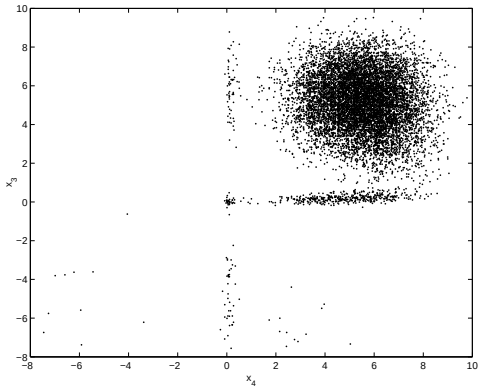

Fig. 15. Scatter plot of points drawn from $\pi_{3:4|0:100}(x_{3:4}|y_{0:100})$ (from [66]).

the weighted sample $\{(x_{t+1}^{(i)}, \omega_{t+1|0:T}^{(i)})\}_{1 \le i \le N}$, i.e.,

$$\pi_{t+1|0:T}(dx_{t+1}|y_{0:T}) \approx \sum_{i=1}^{N} \omega_{t+1|0:T}^{(i)} \delta_{x_{t+1}^{(i)}}(dx_{t+1}) \ ,$$

then we may substitute this and the approximation of the filtering distribution from time t into (30) and (26) to obtain

$$\pi_{t|0:T}(dx_t|y_{0:T}) \approx \sum_{i=1}^{N} \omega_{t|0:T}^{(i)} \delta_{x_t^{(i)}}(dx_t) \ ,$$

where the new weight is recursively updated according to

$$\omega_{t|0:T}^{(i)} = \omega_t^{(i)} \left(\sum_{j=1}^{N} \frac{\omega_{t+1|0:T}^{(j)} \ f(x_{t+1}^{(j)}|x_t^{(i)})}{\sum_{k=1}^{N} f(x_{t+1}^{(j)}|x_t^{(k)}) \omega_t^{(k)}} \right) \ .$$

Note that this procedure inherently requires of the order of $O(N^2 T)$ operations, and hence, is very expensive to compute as the number of particles becomes large.

Other forms of marginal smoothing can be obtained using the so-called two-filter formula, see [1], [71], although it should be noted that it is not always straightforward to initialise or implement the required backward filtering pass.

The required two-filter factorisation is:

$$\begin{aligned}
\pi_{t|0:T}(x_t|y_{0:T}) &= \frac{\pi_{t|0:t}(x_t|y_{0:t}) p(y_{t+1:T}|y_{0:t}, x_t)}{p(y_{t+1:T}|y_{0:t})} \\
&\propto \pi_{t|0:t}(x_t|y_{0:t}) p(y_{t+1:T}|x_t) \ .
\end{aligned}$$

Note that this requires the approximation from the usual forward filter plus a backward 'anti-causal prediction' function $p(y_{t+1:T}|x_t)$[3]. See also [74] for further developments in this area. Some restrictive assumptions in the two-filter smoothers have been removed, and schemes for making them computationally more tractable have been introduced in [72], [73]. Note that these two-filter methods focus on approximation of smoothed marginals $\pi_{t|0:T}(x_t|y_{0:T})$ rather than the full sequence $\pi_{0:T|0:T}(x_{0:T}|y_{0:T})$ as in [66].

In some applications, a Maximum *a posteriori* (MAP) estimate is required for the state sequence rather than samples from the posterior distribution, i.e. one is interested in

$$\operatorname*{argmax}_{x_{0:T}} \pi_{0:T|0:T}(x_{0:T}|y_{0:T}) = \operatorname*{argmax}_{x_{0:T}} \pi_0(x_0) \prod_{t=1}^{T} f(x_t|x_{t-1}) \prod_{t=0}^{T} g(y_t|x_t) \ .$$

This can be obtained in several ways from the particle filter output. A common misconception is that the MAP estimate may be found by simply choosing the particle trajectory with largest weight $\omega_T^{(i)}$. This, however, is not correct as the weights depend upon the target distribution $\pi_{0:T|0:T}$ *and* the proposal distribution. A suitable on-line procedure is given in [75]. In this the particle representation is considered as a randomised adaptive discrete grid approximation to the target distribution. Since we can evaluate the transition probabilities between any two states at adjacent times (via $f(x_t|x_{t-1})$) and also the observation probabilities $g(y_t|x_t)$, the discrete approximation may be interpreted as a Hidden Markov Model with N states. Thus the classical Viterbi algorithm [76] may be employed to find a particle estimate of the MAP sequence at any given time. Specifically, the Viterbi algorithm here finds the exact maximiser of the target distribution $\pi_{0:T|0:T}$, *subject to* the constraint that individual states lie on the discrete particle grid, $x_t \in \{x_t^{(i)}\}_{1\leq i\leq N}$, for $t \in \{0,\ldots,T\}$. The procedure is summarised in Algorithm 6.

Note that this algorithm is genuinely sequential in that $\widehat{x}_{0:t}$ approximates the MAP estimator for each and every t that the above algorithm operates, but that the algorithm is again $O(N^2)$ at each time step. The methods of [73] have also been applied to speeding up of this expensive algorithm.

Algorithm 6 Particle MAP Sequence Estimator

Run particle filter for $t = 0$ to obtain particle locations $\{x_0^{(i)}, i = 1,\ldots,N\}$
for $i = 1$ to N **do**
  $\alpha_0^{(i)} = \pi_0(x_0^{(i)})g(y_0|x_0^{(i)})$.
  $z_0^{(i)} = x_0^{(i)}$.
end for
$j_0^{\max} = \operatorname{argmax}_j \alpha_0^{(j)}$.
$\widehat{x}_0 = x_0^{(j_0^{\max})}$.
for $t = 1$ to T **do**
  Run particle filter for time t to obtain particle locations $\{x_t^{(i)}\}_{1\leq i\leq N}$.
  for $i = 1$ to N **do**
    $\alpha_t^{(i)} = \max_j \alpha_{t-1}^{(j)} f(x_t^{(i)}|x_{t-1}^{(j)})g(y_t|x_t^{(i)})$.
    $j_t^{(i)} = \operatorname{argmax}_j \alpha_{t-1}^{(j)} f(x_t^{(i)}|x_{t-1}^{(j)})g(y_t|x_t^{(i)})$.
    $z_{0:t}^{(i)} = (z_{0:t-1}^{(j_t^{\max})}, x_t^{(i)})$.
  end for
  $j_t^{\max} = \operatorname{argmax}_j \alpha_t^{(i)}$.
  $\widehat{x}_{0:t} = z_{0:t}^{(j_t^{\max})}$.
end for

IV. Parameter estimation

We now consider the practically important problem of calibrating system parameters to observations, otherwise known as 'parameter estimation'. In this section we thus assume that both the state transition density and the conditional likelihood function depend not only upon the dynamic state x_t, but also on a static parameter vector θ, which will be stressed by use of the notations $f(x_t|x_{t-1},\theta)$ and $g(y_t|x_t,\theta)$.

Depending on the requirements of a given application, calibration of model parameters can be carried in two very different modes. If the calibration data is available in a batch beforehand, the estimation of the parameters will generally be done prior to the state inference (filtering or smoothing) task. We refer to this situation as batch-mode or off-line estimation. On the other hand, in some cases the parameters have to be estimated sequentially without the data being stored, which we refer to as on-line estimation.

A. Batch methods

In the batch setting, the parameters can be estimated with non-sequential Monte Carlo methods, such as Markov Chain Monte Carlo [53]. It has now been recognised however that the use of sequential Monte Carlo methods offers some significant advantages over these non-sequential methods in certain cases [77], [78], [7]. A first point to note is that running a sequential Monte Carlo method for a given value θ of the parameter does itself provide a simple way of evaluating the data likelihood

$$\ell_{0:T}(y_{0:T}|\theta) = \int p(y_{0:T}, x_{0:T}|\theta) dx_{0:T}$$

by use of the following decomposition:

$$\ell_{0:T}(y_{0:T}|\theta) = \ell_0(y_0|\theta) \prod_{t=1}^{T-1} \ell_{t+1|0:t}(y_{t+1}|y_{0:t},\theta), \qquad (31)$$

where the individual predictive likelihood terms are defined as:

$$\ell_{t+1|0:t}(y_{t+1}|y_{0:t}) = \int p(y_{t+1}, x_{t+1}|y_{0:t}\theta) dx_{t+1}.$$

[3]It should be noted that the backward function is not a probability distribution (it is not, and in some cases it may not be, normalised), see [72], [7], [73] for further discussion of this isssue.

These terms may be easily estimated from the weighted particles $\{(x_t^{(i,\theta)}, \omega_t^{(i,\theta)})\}_{1\leq i\leq N}$ as

$$\ell_{t+1|0:t}(y_{t+1}|y_{0:t},\theta) = \iint g(y_{t+1}|x_{t+1},\theta) f(x_{t+1}|x_t,\theta) \pi_{t|0:t}(x_t|y_{0:t},\theta) dx_t dx_{t+1} \quad (32)$$

$$\approx \sum_{i=1}^{N} \omega_t^{(i,\theta)} \int g(y_{t+1}|x_{t+1},\theta) f(x_{t+1}|x_t^{(i,\theta)},\theta) dx_{t+1} \,. \quad (33)$$

The superscript θ highlights the fact that both the weights $\omega_t^{(i,\theta)}$ and particle positions $x_t^{(i,\theta)}$ depend on the parameter value θ used to construct the weighted sample approximation $\{(x_{0:t}^{(i,\theta)}, \omega_t^{(i,\theta)})\}_{1\leq i\leq N}$ of the filtering distribution. The integral in (33) may be evaluated within the sequential importance sampling framework, using the new particle positions $x_{t+1}^{(i,\theta)}$. For instance, when using the bootstrap filter discussed in Section II-C, the new particle positions $\tilde{x}_{t+1}^{(i)}$ are drawn from the mixture distribution $\sum_{i=1}^{N} \omega_t^{(i,\theta)} f(x|x_t^{(i,\theta)},\theta)$ and the associated unnormalised importance weights write $\tilde{\omega}_t^{(i,\theta)} = g(y_{t+1}|\tilde{x}_{t+1}^{(i)},\theta)$; in this case, the predictive likelihood approximation simplifies to

$$\ell_{t+1|0:t}(y_{t+1}|y_{0:t},\theta) \approx \sum_{i=1}^{N} \tilde{\omega}_{t+1}^{(i,\theta)} \,.$$

In models where the dimension of the parameter vector θ is small, a first natural solution for parameter estimation consists in using directly the particle approximation to the likelihood $\ell_{0:T}(y_{0:T}|\theta)$ (or rather its logarithm), for instance evaluated on a grid of values of θ. All of [40], [70], [79] discuss the practical and theoretical aspects of this approach, and in particular ways in which the Monte Carlo variance of the log-likelihood evaluation may be controlled.

When the model dimension gets large however, optimising $\ell_{0:T}(y_{0:T}|\theta)$ through a grid-based approximation of its values becomes computationally cumbersome, leading to a necessity for more efficient optimisation strategies. A natural option consists in using iterative optimisation algorithms, such as Gauss-Newton or the steepest ascent algorithm (and variants of it) or the EM (Expectation-Maximisation) algorithm [80], [81], [82], [7]. From a practical perspective, these two options imply similar computations as both the evaluation of the gradient of the log-likelihood function $\nabla \ell_{0:T}(y_{0:T}|\theta)$ or the E-step of the EM algorithm require computation of quantities in the form

$$\tau_{T|0:T}(y_{0:T},\theta) = \mathrm{E}\left[\sum_{t=0}^{T-1} s_t(x_t, x_{t+1}) \middle| y_{0:T}, \theta \right] \,, \quad (34)$$

where s_0 to s_T are vector-valued functions which may implicitly also depend on the observations and the parameter. The full derivation of the EM or gradient equations would require lengthier developments (see, e.g., [83] and [7]) and we simply note that in the EM approach, the appropriate functions are of the form $s_t(x_t, x_{t+1}) = \log f(x_{t+1}|x_t,\theta') + \log g(y_{t+1}|x_{t+1},\theta')$ for $t \geq 1$ and $s_0(x_0) = \log g(y_0|x_0,\theta')$ (assuming that the prior distribution π_0 of the initial state does not depend on the parameter); while for gradient based methods, the proper choice of functions in (34) is $s_t(x_t, x_{t+1}) = \nabla \log f(x_{t+1}|x_t,\theta) + \nabla \log g(y_{t+1}|x_{t+1},\theta)$ (for $t \geq 1$) and $s_0(x_0) = \nabla \log g(y_0|x_0,\theta)$ (using the so called *Fisher identity* [84])[4]. The most natural sequential Monte Carlo approximation to (34) is given by:

$$\hat{\tau}_{T|0:T}(y_{0:T},\theta) = \sum_{i=1}^{N} \omega_T^{(i,\theta)} \sum_{t=0}^{T-1} s_t(x_t^{(i,\theta)}, x_{t+1}^{(i,\theta)}) \,, \quad (35)$$

which obviously admits a simple recursive form that can be evaluated without storing the whole particle path but keeping track only of $\sum_{t=0}^{T-1} s_t(x_t^{(i,\theta)}, x_{t+1}^{(i,\theta)})$, for $i = 1,\ldots,N$, in addition to the current particle positions and weights. This approximation has been derived by [87], [88], [89], [90] using different arguments (see also [89], [83], [91] for alternative proposals). Such approximations have been used with reasonable successes either using Monte Carlo versions of the EM algorithm [7] or stochastic gradient procedures [89]. There are however some empirical and theoretical evidences that, when the number of observations T becomes large, the number N of particles should be increased to ensure the convergence of the optimisation procedure [7], [92]. This observation is closely related to the unsatisfactory behaviour of the basic particle filter when used to approximate smoothing distributions, as illustrated by Figure 11. It has been observed in practice, that the mean squared error between $\tau_{T|0:T}$ and $\hat{\tau}_{T|0:T}$ can be reduced, sometimes very significantly, by replacing (35) by an approximation based on *fixed-lag smoothing* [7], [93]. Recent theoretical analyses confirm that the Monte Carlo error of the fixed-lag approximation to (35) can be controlled uniformly in T (in a suitable sense), under mild assumptions on the number N of particles and on the lag used in the smoothing procedure [94].

In [92], the degeneracy of the joint smoothing distribution is addressed using a technique originally introduced in [95], which consists in splitting observations into blocks of equal sizes and defining a proxy of the log-likelihood of the full observations by summing the log-likelihood of these individual adjacent blocks. Because the size of the block is fixed, the accuracy of the likelihood estimator over each individual block does not depend on the number of observations T, making the procedure usable even if the sample size is very large; the downside is that choosing an appropriate size for the individual blocks introduces an additional parameter in the design of the procedure, which is not always easy to set.

B. On-line methods

The methods discussed above have on-line variants as discussed, for instance, in [89], [88], [83], [92]. The most obvious options consist in embedding the previously discussed SMC-based gradient approximations in a stochastic approximation framework; see [96] and the references therein.

[4] As a side comment, note that it is possible to rewrite (34) in a form which is suitable for recursive implementation, although it does involve updating an auxiliary quantity in addition to $\tau_{t|0:t}$ itself and $\pi_{t|0:t}$, following the approach first described in [85] (see also [86], [7]). When the quantity of interest is the gradient of the log-likelihood, the obtained recursions are fully equivalent to the so-called *sensitivity equations* which may be obtained by formally differentiating with respect to the parameter θ the logarithm of (31) and (32) [7]. This recursive rewriting however is mostly useful in models where exact computations are feasible.

For Bayesian dynamic models, however, the most natural option consists in treating the unknown parameter θ, using the state-space representation, as a component of the state which has no dynamic evolution, also referred to as a *static parameter*. Hence, we can reformulate our initial objectives as trying to simulate from the joint posterior distribution of the unobservable states and parameters $\pi_{0:t|0:t}(x_{0:t}, \theta|y_{0:t})$. Unfortunately, the direct use of particle filtering techniques described so far is bound to fail in this particular case since the absence of evolution for θ implies that the exploration of the parameter space is limited to the first time index: at subsequent times the initial parameter samples will only be reweighted or resampled but will not be moved around. A pragmatic solution consists in running the sequential Monte Carlo filter using an artificial, hopefully negligible, dynamic equation on the parameter θ (typically a random walk-like dynamic with a small variance); see [3], [97], [98]. This approach can also be related to kernel estimate ideas where the target filtering and smoothing distributions are smoothed using a kernel with a small bandwidth [99], [100].

The idea put forward in [101], [56], [59] is based on using Markov chain Monte Carlo (MCMC) moves, as briefly discussed in Section II-H, in order to maintain the diversity of the samples in the parameter space. Here the stationary distribution for the MCMC will be the full joint posterior distribution of states and parameters, $\pi_{0:t|0:t}(x_{0:t}, \theta|y_{0:t})$, and a natural choice of algorithm structure might be to apply Metropolis-within-Gibbs sampling steps separately to $\pi_{0:t|0:t}(\theta|x_{0:t}, y_{0:t})$ and $\pi_{0:t|0:t}(x_{0:t}|\theta, y_{0:t})$. Note, however, that in general models this will not be feasible for large datasets, since sampling from $\pi_{0:t|0:t}(\theta|x_{0:t}, y_{0:t})$ may involve recomputing statistics based on the entire path $x_{0:t}$ and $y_{0:t}$. In many models of interest, however, this will not be necessary, since the influence of the path $x_{0:t}$ and $y_{0:t}$ may be summarised by low-dimensional *sufficient statistics*. To give a simple example of this situation, consider the nonlinear time series model of Example 1 where the observation equation takes the form $y_t = \theta b(x_t) + v_t$ where $v_t \sim \mathcal{N}(0, \sigma_v^2)$, $b(x) = x^2$ and θ is a scalar parameter, which is here assumed unknown for illustration purposes. It is easily checked that the conditional distribution of the observation and the state variables given the parameter θ is proportional to

$$p(\theta|x_{0:T}, y_{0:T}) \propto \mathcal{N}\left(\theta \left| \frac{\sum_{t=0}^{T} y_t b(x_t)}{\sum_{t=0}^{T} b^2(x_t)}, \frac{\sum_{t=0}^{T} b^2(x_t)}{\sigma_v^2} \right.\right)$$

Hence if θ is equipped with a prior distribution $p(\theta)$ chosen in the *conjugate family* [102], which here will be any Gaussian distribution, the posterior distribution $p(\theta|x_{0:T}, y_{0:T})$ is known and depends upon the observation and the state variables through only two low-dimensional sufficient statistics, $\sum_{t=0}^{T} y_t f(x_t)$ and $\sum_{t=0}^{T} f^2(x_t)$. Note that the argument used here turns out to be very similar to the situation encountered when approximating the behaviour of gradient-based or EM methods — see (34) and associated discussion. In such a case, it is possible devise a particle filter which simulates from the posterior distribution of the states, with the parameter θ, directly regenerated from its full conditional distribution $p(\theta|x_{0:t}, y_{0:t})$ using a single Gibbs sampling step. For example, we may place this within the setting of the bootstrap filter of Section II-C, denoting by S_t the, possibly vector-valued, sufficient statistic at time t and by s the function such that $S_t = S_{t-1} + s(x_t, y_t)$.

Note that in some models featuring conditionally Gaussian distributions, an alternative version of this algorithm would marginalise θ directly and run a fully marginalised particle filter on just $x_{0:t}$, as in Section II-G, see also [18] for models with static parameters of this kind.

Algorithm 7 Bootstrap Filter with parameter regeneration

for $i = 1, \ldots, N$ **do** ▷ Initialisation
Sample $\theta_0^{(i)} \sim p(\theta^{(i)})$ and $\tilde{x}_0^{(i)} \sim \pi_0(x_0|\theta_0^{(i)})$.
Compute statistics $\tilde{S}_0^{(i)} = s(\tilde{x}_0^{(i)}, y_0)$.
Assign initial importance weights

$$\tilde{\omega}_0^{(i)} = g(y_0|\tilde{x}_0^{(i)}, \theta_0^{(i)}) \ .$$

end for
for $t = 1, \ldots, T$ **do**
Select N particle indices ▷ Resampling $j_i \in \{1, \ldots, N\}$ according to weights

$$\{\omega_{t-1}^{(j)}\}_{1 \leq j \leq N} \ .$$

Set $x_{t-1}^{(i)} = \tilde{x}_{t-1}^{(j_i)}$, $\theta_{t-1}^{(i)} = \theta_{t-1}^{(j_i)}$ and $S_{t-1}^{(i)} = \tilde{S}_{t-1}^{j_i}$, $i = 1, \ldots, N$.
for $i = 1, \ldots, N$ **do** ▷ Propagation and weighting
Propagate

$$\theta_t^{(i)} \sim p(\theta_t^{(i)}|S_{t-1}^{(i)}) \ ,$$
$$\tilde{x}_t^{(i)} \sim f(\tilde{x}_t^{(i)}|x_{t-1}^{(i)}, \theta_t^{(i)}) \ .$$

Update statistics $\tilde{S}_t^{(i)} = s(\tilde{x}_t^{(i)}, y_0) + S_{t-1}^{(i)}$.
Compute weight

$$\tilde{\omega}_t^{(i)} = g(y_t|\tilde{x}_t^{(i)}, \theta_t^{(i)}) \ .$$

end for
Normalise weights

$$\omega_t^{(i)} = \tilde{\omega}_t^{(i)} \Big/ \sum_{j=1}^{N} \tilde{\omega}_t^{(j)} \ , \quad i = 1, \ldots, N.$$

end for

At any time point, one may estimate by $\sum_{i=1}^{N} \omega_t^{(i)} h(\tilde{x}_t^{(i)})$ the expectation $\mathrm{E}[h(X_t)|Y_{0:t}]$, where the unknown parameter θ has been marginalised out. Similarly, $\sum_{i=1}^{N} \omega_t^{(i)} \theta_t^{(i)}$ provides an approximation to expectation of θ given $Y_{0:t}$, that is, the minimum mean square estimate of the parameter θ given the observations up to time t. As an alternative, we may also consider instead $\sum_{i=1}^{N} \omega_t^{(i)} p(\theta|\tilde{S}_t^i)$ with provides a smooth approximation to the complete parameter posterior as well as $\sum_{i=1}^{N} \omega_t^{(i)} \mathrm{E}(\theta|\tilde{S}_t^i)$ which has reduced variance, by virtue of the Rao-Blackwell principle. Other more sophisticated examples of this approach are discussed in [103], [104], [105]. Note that as in the case of batch estimation discussed in the previous section, the successive resamplings performed on the accumulated statistics $\tilde{S}_t^{(i)}$ may lead to a sample

impoverishment phenomenon and ultimately compromise the long-term stability of the method [92]. Here again, it is likely that the forgetting ideas discussed at the end of Section IV-A can be put in use to robustify the basic algorithms described above.

V. Non-filtering Uses of Sequential Monte Carlo

Thus far, sequential Monte Carlo has been presented as a technique that is intrinsically related to the filtering and smoothing problems in state-space models. We will now consider the technique from a more general standpoint and review some recent contributions where SMC methods are used for other inference tasks, which are not necessarily intrinsically sequential (see also Section IV-A). In particular, recent population-based sampling algorithms provide methods for parameter estimation in high dimensional batch processing problems where MCMC would typically have been thought of as the method of choice. i As a starting point, note that the basic structure for applying SMC approaches is given by (5)–(6), which we may rewrite in the following more compact form

$$\pi_{0:t}(x_{0:t}) = c_t^{-1}\pi_{0:t-1}(x_{0:t-1})k_t(x_{t-1}, x_t) \ , \qquad (36)$$

where $\pi_{0:l}$ is a l-dimensional probability density function, k_l is an unnormalised transition density function (i.e. $k_l(x, x') \geq 0$ and $\int k_l(x, x')dx' = C(x) < \infty$ but where $C(x)$ may differ from unity), and finally, c_l is the normalising constant defined by

$$c_l = \int \cdots \int \pi_{0:l-1}(x_{0:l-1})k_l(x_{l-1}, x_l)dx_{0:l} \ , \qquad (37)$$

which we may rewrite as $c_l = \iint \pi_{l-1}(x)k_l(x, x')dxdx'$ upon defining by $\pi_k(x_k) = \iint \pi_{0:k}(x_{0:k})dx_{0:k-1}$ the marginal of $\pi_{0:k}$. Equation (36) is referred to by [44] as a (discrete-time) Feynman-Kac system of probability distributions. This structure is encountered in various contexts outside of the standard filtering and smoothing applications, notably in statistical physics [106], [107]. Note that, as with standard filtering, in some cases only the marginals of $\pi_{0:l}$ are of interest and (36) takes the simpler form

$$\pi_t(x_t) = c_t^{-1} \int \pi_{t-1}(x_{t-1})k_t(x_{t-1}, x_t)dx_{t-1} \ , \qquad (38)$$

where $\pi_l(x_l)$ are the one-dimensional marginals.

An example of (38) which is of particular interest occurs when considering the successive posterior distributions of the parameter in a Bayesian model, as more and more observations are incorporated at successive times t. We have already touched upon this topic in the context of parameter estimation for state-space models in Section IV-B, where the matter is complicated due to the Markov dependence in the unobservable states. Both [77] and [108], [109] consider the case where independent observations $y_0, \ldots, y_t$ with common marginal likelihood $\ell(y|\theta)$ are used to estimate the parameter θ. It is then easy to check that the ratio of the posterior corresponding to different observation lengths satisfy

$$\frac{\pi(\theta|y_{0:t+k})}{\pi(\theta|y_{0:t})} \propto \prod_{l=1}^{k} \ell(y_{t+l}|\theta) \ ,$$

which is a (very specific) instance of (38). The methods proposed by [77], [108], [109] combine pure iterated importance sampling steps with techniques that are more characteristic of SMC such as resampling, resample-move proposals [56], or kernel smoothing [99], [100]. It is argued in [77], [108], [109] that the resulting algorithms can be far less computationally demanding than complete-sample Markov Chain Monte Carlo approaches when dealing with large datasets.

A simple example of (36) occurs when considering a target of the form

$$\pi_{0:t}(x_{0:t}) = \prod_{l=1}^{t} \pi(x_l), \qquad (39)$$

i.e. when considering repeated samplings from a fixed distribution $\pi_t = \pi$. We can now envisage schemes which iteratively propose new particles with target distribution $\pi(x_t)$, based on particles from earlier distributions $\pi(x_l)$, $l < t$, in order to refine the approximation to $\pi(x_t)$ as t increases. By analogy with (10), if at iteration l the new particle positions $x_l^{(i)}$ are proposed from $q_l(\cdot|x_{l-1}^{(i)})$, then the corresponding importance weights are given by

$$\omega_{l-1}^{(i)} \times \frac{\pi(x_l^{(i)})}{q_l(x_l^{(i)}|x_{l-1}^{(i)})} \ . \qquad (40)$$

This strategy, termed *Population Monte Carlo* by [110], is mostly of interest when dynamically adapting the form of the importance transition q_l between iterations, as indicated above (see also [2] for an early related approach). Here, the target distribution $\pi(x_l)$ is fixed but one determines, from the output of the simulations, the best possible form of the importance density, given some optimality criterion (such as minimising the Monte Carlo variance). Hence adaptive population Monte Carlo offers an alternative to adaptive MCMC algorithms as proposed by [111] and others. The advantages of population Monte Carlo in this context are twofold: first, the possible computational speedup achievable by parallelising the computations; and second, there are less stringent technical requirements on the adaptation scheme since, for large enough population sizes, the fact that the whole procedure indeed targets π is guaranteed by (40), which implies that weighted averages (Monte Carlo expectations) are unbiased estimates of the expectation under π. Provably efficient rules for adapting mixture importance sampling densities — i.e. transitions of the form $q(x'|x) = \sum_{j=1}^{m} \alpha_j q_j(x'|x)$, where q_j are fixed and $\alpha_1, \ldots \alpha_m$ are the parameters of the proposal to be adapted are given in [112], [113].

Also of interest are cases where the target density is of the form $\pi_t(x) \propto \pi^{\gamma_t}(x)$, where γ_t are positive numbers. Using γ_t strictly smaller than one (and generally converging to one) flattens the target π and is often advocated as a solution for simulating from highly multi-modal distributions, in a process called *simulated tempering* [53]. Conversely, *simulated annealing* which consists in letting γ_t tends to infinity at a sufficiently slow rate is a well-known method for finding the global maximiser(s) of π [114], [101]. Other examples occur in the simulation of rare events where the successive targets π_t correspond to distributions under which

the event of interest is less likely than under the original distribution π ([115] use a classic exponential tilting to achieve this goal, whereas the construction of [116] makes a more thorough use of the Markov property).

In this context, [78] point out that the choice of the product target $\pi_{0:t}(x_{0:t}) = \prod_{l=1}^{t} \pi_l(x_l)$ is mostly arbitrary. Instead, it is possible to *postulate* the existence of a *time-reversed transition density* r_t such that $\int \cdots \int \pi_{0:t}(x_{0:t}) dx_{0:t-2} = r_t(x_{t-1}|x_t)\pi_t(x_t)$, see also [117]. This implies that, if the particles $x_{t-1}^{(i)}$ at iteration t are perturbed using a draw from $q_t(x_t|x_{t-1}^{(i)})$, the importance weights become

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \times \frac{r_t(x_{t-1}^{(i)}|x_t^{(i)})\pi_t(x_t^{(i)})}{\pi_{t-1}(x_{t-1}^{(i)})q_t(x_t^{(i)}|x_{t-1}^{(i)})} ,$$

where both the choice of the importance density q_t and the time-reversed density r_t are user-defined. This scheme, termed the *Sequential Monte Carlo Sampler* by [78], offers much more flexibility in the type of importance density q_t that may be used and, in addition, r_t can, to some extent, be selected so as to reduce the variance of the simulation. The theoretical analysis of the resulting algorithm in [78] is more complicated than for the adaptive population Monte Carlo methods mentioned above since it is not based on repeated applications of basic importance sampling (with resampling) steps, hence lacking the simple unbiasedness property. Several applications of this approach are presented in [118], [119], [120], see also [121] for its application in variable dimension settings.

VI. Conclusion and Discussion

In this article we have reviewed a range of existing core topics in sequential Monte Carlo methodology, and described some of the more recent and emerging techniques. In particular we see the expansion of SMC methods into realms more routinely handled by MCMC or other batch-based inference methods, both for static parameter estimation in dynamical models and for more general inference about high-dimensional distributions. Our coverage is aimed at the methods themselves, so we have not provided a full list of application references, of which there are now many hundreds, nor have we given any details of theoretical analysis, which is now a mature and sophisticated topic. A primary resource for new papers in SMC methods is the SMC Methods Homepage, hosted on the website of The Signal Processing and Communications Group in the University of Cambridge, see `www-sigproc.eng.cam.ac.uk/smc/`. There are several emerging areas which we have not been able to cover, either for reasons of space or because the topics are too new to have generated publications as yet. Amongst these we identify particularly particle methods for random finite set models, see [122], and particle methods for continuous time diffusion models ([123] provides the basic theory for this development in the batch (MCMC) setting). For a snapshot of current emerging work see the proceedings of two recent conferences relevant to the topic: the Workshop on Sequential Monte Carlo Methods: filtering and other applications (Oxford, UK, July 2006), Proceedings to appear in European Series in Applied and Industrial Mathematics (ESAIM), under the auspices of Société de Mathématiques Appliquées et Industrielles (SMAI); and the IEEE Nonlinear Statistical Signal Processing Workshop: Classical, Unscented and Particle Filtering Methods (Cambridge, UK, September 2006).

References

[1] G. Kitagawa, "Monte-Carlo filter and smoother for non-Gaussian nonlinear state space models," *J. Comput. Graph. Statist.*, vol. 1, pp. 1–25, 1996.

[2] M. West, "Mixture models, Monte Carlo, Bayesian updating and dynamic models," *Computing Science Statistics*, vol. 24, pp. 325–333, 1993.

[3] N. Gordon, D. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F, Radar Signal Process.*, vol. 140, pp. 107–113, 1993.

[4] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, 1979.

[5] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Prentice-Hall, 2000.

[6] B. Ristic, M. Arulampalam, and A. Gordon, *Beyond Kalman Filters: Particle Filters for Target Tracking*. Artech House, 2004.

[7] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. Springer, 2005.

[8] A. Jazwinski, *Stochastic processes and filtering theory*. New York: Academic Press, 1970.

[9] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximation," *IEEE Trans. Automat. Control*, vol. 17, no. 4, pp. 439–448, 1972.

[10] R. Kulhavy, "Recursive nonlinear estimation: a geometric approach," *Automatica*, vol. 26, no. 3, pp. 545–555, 1990.

[11] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997.

[12] R. Van der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The unscented particle filter," in *Adv. Neural Inf. Process. Syst.*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2000, vol. 13.

[13] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. Automat. Control*, vol. 45, pp. 910–927, 2000.

[14] J. Handschin and D. Mayne, "Monte Carlo techniques to estimate the conditionnal expectation in multi-stage non-linear filtering," in *Int. J. Control*, vol. 9, 1969, pp. 547–559.

[15] J. Handschin, "Monte Carlo techniques for prediction and filtering of non-linear stochastic processes," *Automatica*, vol. 6, pp. 555–563, 1970.

[16] D. B. Rubin, "A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when the fraction of missing information is modest: the SIR algorithm (discussion of Tanner and Wong)," *J. Am. Statist. Assoc.*, vol. 82, pp. 543–546, 1987.

[17] A. Blake and M. Isard, *Active Contours*. Springer, 1998.

[18] J. Liu and R. Chen, "Blind deconvolution via sequential imputations," *J. Roy. Statist. Soc. Ser. B*, vol. 430, pp. 567–576, 1995.

[19] P. Del Moral, "Nonlinear filtering: interacting particle solution," *Markov Process. Related Fields*, vol. 2, pp. 555–579, 1996.

[20] A. Doucet, N. De Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001.

[21] J. Liu, *Monte Carlo Strategies in Scientific Computing*. New York: Springer, 2001.

[22] J. Liu and R. Chen, "Sequential Monte-Carlo methods for dynamic systems," *J. Roy. Statist. Soc. Ser. B*, vol. 93, pp. 1032–1044, 1998.

[23] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte-Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 10, pp. 197–208, 2000.

[24] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on line non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 241–254, 2002.

[25] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Process. Mag.*, vol. 20, no. 5, pp. 19–38, 2003.

[26] H. Tanizaki, "Nonlinear and non-Gaussian state-space modeling with Monte-Carlo techniques: a survey and comparative study," in *Handbook of Statistics 21. Stochastic processes: Modelling and Simulation*, D. N. Shanbhag and C. R. Rao, Eds. Elsevier, 2003, pp. 871–929.

[27] C. Andrieu, A. Doucet, and C. P. Robert, "Computational advances for and from Bayesian analysis," *Statist. Sci.*, vol. 19, no. 1, pp. 118–127, 2004.

[28] J. Geweke, "Bayesian inference in econometric models using Monte-Carlo integration," *Econometrica*, vol. 57, no. 6, pp. 1317–1339, 1989.

[29] D. B. Rubin, "Using the SIR algorithm to simulate posterior distribution," in *Bayesian Statistics 3*, J. M. Bernardo, M. DeGroot, D. Lindley, and A. Smith, Eds. Clarendon Press, 1988, pp. 395–402.

[30] J. Carpenter, P. Clifford, and P. Fearnhead, "An improved particle filter for non-linear problems," *IEE Proc., Radar Sonar Navigation*, vol. 146, pp. 2–7, 1999.

[31] P. Fearnhead and P. Clifford, "On-line inference for hidden Markov models via particle filters," *J. Roy. Statist. Soc. Ser. B*, vol. 65, pp. 887–899, 2003.

[32] H. R. Künsch, "Recursive Monte-Carlo filters: algorithms and theoretical analysis," *Ann. Statist.*, vol. 33, no. 5, pp. 1983–2021, 2005.

[33] R. Douc, O. Cappé, and E. Moulines, "Comparison of resampling schemes for particle filtering," in *4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Zagreb, Croatia, September 2005, arXiv: cs.CE/0507025.

[34] Y. C. Ho and R. C. K. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Trans. Automat. Control*, vol. 9, no. 4, pp. 333–339, 1964.

[35] P. Del Moral and J. Jacod, "Interacting particle filtering with discrete-time observations: Asymptotic behaviour in the Gaussian case," in *Stochastics in Finite and Infinite Dimensions: In Honor of Gopinath Kallianpur*, T. Hida, R. L. Karandikar, H. Kunita, B. S. Rajput, S. Watanabe, and J. Xiong, Eds. Boston, MA: Birkhäuser, 2001, pp. 101–122.

[36] A. Kong, J. S. Liu, and W. Wong, "Sequential imputation and Bayesian missing data problems," *J. Am. Statist. Assoc.*, vol. 89, no. 278-288, pp. 590–599, 1994.

[37] V. Zaritskii, V. Svetnik, and L. Shimelevich, "Monte-Carlo techniques in problems of optimal data processing," *Autom. Remote Control*, vol. 12, pp. 2015–2022, 1975.

[38] H. Akashi and H. Kumamoto, "Random sampling approach to state estimation in switching environment," *Automatica*, vol. 13, pp. 429–434, 1977.

[39] R. Chen and J. S. Liu, "Mixture Kalman filter," *J. Roy. Statist. Soc. Ser. B*, vol. 62, no. 3, pp. 493–508, 2000.

[40] N. Shephard and M. Pitt, "Likelihood analysis of non-Gaussian measurement time series," *Biometrika*, vol. 84, no. 3, pp. 653–667, 1997, erratum in volume 91, 249–250, 2004.

[41] P. Del Moral, "Measure-valued processes and interacting particle systems. Application to nonlinear filtering problems," *Ann. Appl. Probab.*, vol. 8, pp. 69–95, 1998.

[42] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 736–746, 2002.

[43] N. Chopin, "Central limit theorem for sequential monte carlo methods and its application to bayesian inference," *Ann. Statist.*, vol. 32, no. 6, pp. 2385–2411, 2004.

[44] P. Del Moral, *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. Springer, 2004.

[45] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Am. Statist. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.

[46] J. L. Zhang and J. S. Liu, "A new sequential importance sampling method and its application to the two-dimensional hydrophobic-hydrophilic model," *J. Chem. Physics*, vol. 117, no. 7, 2002.

[47] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1762–1770, 2003.

[48] S. Godsill and T. Clapp, "Improvement strategies for monte carlo particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. De Freitas, and N. Gordon, Eds. Springer, 2001.

[49] C. Andrieu and A. Doucet, "Particle filtering for partially observed Gaussian state space models," *J. Roy. Statist. Soc. Ser. B*, vol. 64, no. 4, pp. 827–836, 2002.

[50] R. Karlsson, T. Schön, and F. Gustafsson, "Complexity analysis of the marginalized particle filter," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4408–4411, 2005.

[51] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2279–2289, 2005.

[52] R. E. Kalman and R. Bucy, "New results in linear filtering and prediction theory," *J. Basic Eng., Trans. ASME, Series D*, vol. 83, no. 3, pp. 95–108, 1961.

[53] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. New York: Springer, 2004.

[54] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.

[55] A. C. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1989.

[56] W. R. Gilks and C. Berzuini, "Following a moving target—Monte Carlo inference for dynamic Bayesian models," *J. Roy. Statist. Soc. Ser. B*, vol. 63, no. 1, pp. 127–146, 2001.

[57] C. Berzuini and W. R. Gilks, "Resample-move filtering with cross-model jumps," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. De Freitas, and N. Gordon, Eds. Springer, 2001.

[58] S. N. MacEachern, M. Clyde, and J. Liu, "Sequential importance sampling for nonparametric bayes models: The next generation," *Can. J. Statist.*, vol. 27, pp. 251–267, 1999.

[59] P. Fearnhead, "Markov chain Monte Carlo, sufficient statistics and particle filter," *J. Comput. Graph. Statist.*, vol. 11, no. 4, pp. 848–862, 2002.

[60] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805–1918, 2005.

[61] T. C. Clapp and S. J. Godsill, "Fixed-lag smoothing using sequential importance sampling," in *Bayesian Statistics VI*, J. Bernardo, J. Berger, A. Dawid, and A. Smith, Eds. Oxford University Press, 1999, pp. 743–752.

[62] L. Tierney, "Markov chains for exploring posterior disiributions (with discussion)," *Ann. Statist.*, vol. 22, no. 4, pp. 1701–1762, 1994.

[63] A. Doucet, A. Logothetis, and V. Krishnamurthy, "Stochastic sampling algorithms for state estimation of jump Markov linear systems," *IEEE Trans. Automat. Control*, vol. 45, no. 2, pp. 188–202, 2000.

[64] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 49, pp. 613–624, 2001.

[65] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, Eds., *Markov Chain Monte Carlo in Practice*, ser. Interdisciplinary Statistics Series. Chapman & Hall, 1996.

[66] S. J. Godsill, A. Doucet, and M. West, "Monte carlo smoothing for non-linear time series," *J. Am. Statist. Assoc.*, vol. 50, pp. 438–449, 2004.

[67] H. R. Künsch, "State space and hidden markov models." in *Complex Stochastic Systems*, O. E. Barndorff-Nielsen, D. R. Cox, and C. Klueppelberg, Eds. Boca raton: CRC Publisher, 2001, pp. 109–173.

[68] W. Fong, S. Godsill, A. Doucet, and M. West, "Monte carlo smoothing with application to audio signal enhancement," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 438–449, 2002.

[69] G. Kitagawa, "Non-Gaussian state space modeling of nonstationary time series," *J. Am. Statist. Assoc.*, vol. 82, no. 400, pp. 1023–1063, 1987.

[70] M. Hürzeler and H. R. Künsch, "Monte Carlo approximations for general state-space models," *J. Comput. Graph. Statist.*, vol. 7, pp. 175–193, 1998.

[71] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *Int. J Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[72] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state-space models," University of Cambridge, Department of Engineering, Tech. Rep. TR-CUED-F-INFENG 498, 2004.

[73] M. Klaas, M. Briers, N. De Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: If i had a million particles," in *23rd Int. Conf. Machine Learning (ICML)*, Pittsburgh, Pennsylvania, June 25-29 2006.

[74] H. Tanizaki, "Nonlinear and non-Gaussian state space modeling using sampling techniques," *Ann. Inst. Statist. Math.*, vol. 53, no. 1, pp. 63–81, 2001.

[75] S. J. Godsill, A. Doucet, and M. West, "Maximum *a posteriori* sequence estimation using Monte Carlo particle filters," *Ann. Inst. Stat. Math.*, vol. 53, no. 1, pp. 82–96, Mar. 2001.

[76] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[77] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, pp. 539–552, 2002.

[78] P. Del Moral, A. Doucet, and A. Jasra, "Sequential monte carlo samplers," *J. Roy. Statist. Soc. Ser. B*, vol. 68, no. 3, p. 411, 2006.

[79] J. Olsson and T. Rydén, "Asymptotic properties of the bootstrap particle filter maximum likelihood estimator for state space models," Lund University, Tech. Rep. LUTFMS-5052-2005, 2005.

[80] R. Shumway and D. Stoffer, "An approach to time series smoothing and forecasting using the em algorithm," *J. Time Ser. Anal.*, vol. 3, no. 4, pp. 253–264, 1982.

[81] F. Campillo and F. Le Gland, "MLE for patially observed diffusions: Direct maximization vs. the EM algorithm," *Stoch. Proc. App.*, vol. 33, pp. 245–274, 1989.

[82] M. Segal and E. Weinstein, "A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems," *IEEE Trans. Inform. Theory*, vol. 35, pp. 682–687, 1989.

[83] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadic, "Particle methods for change detection, system identification, and control," *IEEE Proc.*, vol. 92, no. 3, pp. 423–438, 2004.

[84] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38 (with discussion), 1977.

[85] O. Zeitouni and A. Dembo, "Exact filters for the estimation of the number of transitions of finite-state continuous-time Markov processes," *IEEE Trans. Inform. Theory*, vol. 34, no. 4, July 1988.

[86] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*. New York: Springer, 1995.

[87] O. Cappé, "Recursive computation of smoothed functionals of hidden Markovian processes using a particle approximation," *Monte Carlo Methods Appl.*, vol. 7, no. 1–2, pp. 81–92, 2001.

[88] F. Cérou, F. Le Gland, and N. Newton, "Stochastic particle methods for linear tangent filtering equations," in *Optimal Control and PDE's - Innovations and Applications, in Honor of Alain Bensoussan's 60th Anniversary*, J.-L. Menaldi, E. Rofman, and A. Sulem, Eds. Amsterdam: IOS Press, 2001, pp. 231–240.

[89] A. Doucet and V. B. Tadić, "Parameter estimation in general state-space models using particle methods," *Ann. Inst. Statist. Math.*, vol. 55, no. 2, pp. 409–422, 2003.

[90] J. Fichou, F. Le Gland, and L. Mevel, "Particle based methods for parameter estimation and tracking: Numerical experiments," INRIA, Tech. Rep. PI-1604, 2004.

[91] G. Poyiadjis, A. Doucet, and S. S. Singh, "Particle methods for optimal filter derivative: application to parameter estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 18-23 March 2005, pp. v/925–v/928.

[92] C. Andrieu, A. Doucet, and V. B. Tadic, "Online simulation-based methods for parameter estimation in non linear non gaussian state-space models," in *Proc. IEEE Conf. Decis. Control*, 2005.

[93] O. Cappé and E. Moulines, "On the use of particle filtering for maximum likelihood parameter estimation," in *European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, September 2005.

[94] J. Olsson, O. Cappé, R. Douc, and E. Moulines, "Sequential Monte Carlo smoothing with application to parameter estimation in non-linear state space models," Lund University, Tech. Rep., 2006, arXiv:math.ST/0609514.

[95] T. Rydén, "Consistent and asymptotically normal parameter estimates for hidden Markov models," *Ann. Statist.*, vol. 22, no. 4, pp. 1884–1895, 1994.

[96] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer, 1997.

[97] G. Kitagawa, "A self-organizing state-space model," *J. Am. Statist. Assoc.*, vol. 93, no. 443, pp. 1203–1215, 1998.

[98] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, N. D. F. A. Doucet and N. Gordon, Eds. Springer, 2001.

[99] P. Stavropoulos and D. M. Titterington, "Improved particle filters and smoothing," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. De Freitas, and N. Gordon, Eds. Springer, 2001.

[100] C. Musso, N. Oudjane, and F. Le Gland, "Improving regularized particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. De Freitas, and N. Gordon, Eds. Springer, 2001.

[101] R. M. Neal, "Annealed importance sampling," *Stat. Comput.*, vol. 11, no. 2, pp. 125–139, 2001.

[102] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. New York: Chapman, 1995.

[103] G. Storvik, "Particle filters for state-space models with the presence of unknown static parameters," *IEEE Trans. Signal Process.*, pp. 281–289, 2002.

[104] J. Stroud, N. Polson, and P. Müller, "Practical filtering for stochastic volatility models," in *State Space and Unobserved Component Models*, A. Harvey, S. J. Koopman, and N. Shephard, Eds. Cambridge University Press, 2003.

[105] A. Papavasiliou, "A uniformly convergent adaptive particle filter," *J. Appl. Probab.*, vol. 42, no. 4, pp. 1053–1068, 2005.

[106] Y. Iba, "Population-based Monte Carlo algorithms," *Trans. Japanese Soc. Artificial Intell.*, vol. 16, no. 2, pp. 279–286, 2000.

[107] ——, "Extended ensemble monte carlo," *nt. J. Mod. Phys. C*, vol. 12, no. 5, pp. 623–656, 2001.

[108] G. Ridgeway and D. Madigan, "A sequential Monte Carlo method for Bayesian analysis of massive datasets," *Data Mining and Knowledge Discovery*, vol. 7, no. 3, pp. 301–319, 2003.

[109] S. Balakrishnan and D. Madigan, "A one-pass sequential Monte Carlo method for Bayesian analysis of massive datasets," *Bayesian Analysis*, vol. 1, no. 2, pp. 345–362, 2006.

[110] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert, "Population Monte Carlo," *J. Comput. Graph. Statist.*, vol. 13, no. 4, pp. 907–929, 2004.

[111] H. Haario, E. Saksman, and J. Tamminen, "Adaptive proposal distribution for random walk Metropolis algorithm," *Computational Statistics*, vol. 14, pp. 375–395, 1999.

[112] R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert, "Convergence of adaptive mixtures of importance sampling schemes," *Ann. Statist.*, vol. 35, no. 1, 2007, to appear.

[113] ——, "Minimum variance importance sampling via population monte carlo," CEREMADE, Tech. Rep., 2005.

[114] P. J. V. Laarhoven and E. H. L. Arts, *Simulated Annealing: Theory and Applications*. Reidel Publisher, 1987.

[115] P. Del Moral and J. Garnier, "Genealogical particle analysis of rare events," *Ann. Appl. Probab.*, vol. 15, no. 4, pp. 2496–2534, 2005.

[116] F. Cerou, P. Del Moral, F. Le Gland, and P. Lezaud, "Limit theorems for multilevel splitting algorithms in the simulation of rare events," in *Proceedings of the 37th Winter Simulation Conference*, Orlando, Florida, 2005.

[117] R. M. Neal, "Markov chain Monte Carlo methods based on 'slicing' the density function," University of Toronto, Tech. Rep., 1997.

[118] A. Jasra, D. A. Stephens, and C. C. Holmes, "On population-based simulation for static inference," Department of Mathematics, Imperial College, Tech. Rep., 2005.

[119] A. Johansen, A. Doucet, and M. Davy, "Maximum likelihood parameter estimation for latent variable models using sequential Monte Carlo," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2006.

[120] A. M. Johansen, P. Del Moral, and A. Doucet, "Sequential Monte Carlo samplers for rare events," in *Proceedings of the 6th International Workshop on Rare Event Simulation*, Bamberg, Germany, Oct. 2006.

[121] J. Vermaak, S. J. Godsill, and A. Doucet, "Sequential Bayesian kernel regression," in *Adv. Neural Inf. Process. Syst.* MIT Press, 2003.

[122] B.-N. Vo, S. S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multi-target filtering with random finite sets," *IEEE Aerospace and Electronic Systems*, 2007, to appear.

[123] A. Beskos, O. Papaspiliopoulos, G. O. Roberts, and P. Fearnhead, "Exact and efficient likelihood-based estimation for discretely observed diffusions processes," *J. Roy. Statist. Soc. Ser. B*, vol. 68, no. 2, pp. 1–29, 2006.