Exact Approximation of Rao-Blackwellised Particle Filters ⋆

Adam M. Johansen * **Nick Whiteley** ** **Arnaud Doucet** ***

* *University of Warwick, Department of Statistics, Coventry, CV4 7AL, UK (e-mail: a.m.johansen@warwick.ac.uk)*
** *University of Bristol, School of Mathematics, Bristol, BS1 8TW, UK (e-mail: nick.whiteley@bristol.ac.uk)*
*** *University of Oxford, Department of Statistics, 1 South Parks Road Oxford, OX1 3TG, UK (e-mail: doucet@stats.ox.ac.uk)*

Abstract: Particle methods are a category of Monte Carlo algorithms that have become popular for performing inference in non-linear non-Gaussian state-space models. The class of "Rao-Blackwellised" particle filters exploits the analytic marginalisation that is possible for some state-space models to reduce the variance of the Monte Carlo estimates. Despite being applicable to only a restricted class of state-space models, such as conditionally linear Gaussian models, these algorithms have found numerous applications. In scenarios where no such analytical integration is possible, it has recently been proposed in Chen et al. [2011] to use "local" particle filters to carry out this integration numerically. We propose here an alternative approach also relying on "local" particle filters which is more broadly applicable and has attractive theoretical properties. Proof-of-concept simulation results are presented.

Keywords: Dynamic Systems; Monte Carlo Method; Optimal Filtering; Target Tracking; Target Tracking Filters

1. INTRODUCTION

Let $\{(X_n, Z_n)\}_{n\geq 1} \in (\mathcal{X} \times \mathcal{Z})^{\mathbb{N}}$ be an unobserved homogeneous Markov process characterized by its initial density

$$(X_1, Z_1) \sim \mu(\cdot) \tag{1}$$

and transition probability density

$$(X_n, Z_n) \,|(X_{n-1} = x_{n-1}, Z_{n-1} = z_{n-1}) \sim f(\cdot|x_{n-1}, z_{n-1}) \tag{2}$$

w.r.t to a dominating measure, e.g. Lebesgue, denoted abusively $dx_n dz_n$. The observations $\{Y_n\}_{n\geq 1} \in \mathcal{Y}^{\mathbb{N}}$ are assumed to be conditionally independent given $\{(X_n, Z_n)\}_{n\geq 1}$, and their common marginal probability density is given by

$$Y_n|\,(X_n = x_n, Z_n = z_n) \sim g(\cdot|\,x_n, z_n) \tag{3}$$

w.r.t. to a dominating measure denoted dy_n.

Hereafter, for any generic sequence $\{u_n\}$, $u_{i:j}$ will denote $(u_i, u_{i+1}, \ldots, u_j)$. From the model definition, we have the following joint density

$$p(x_{1:n}, z_{1:n}, y_{1:n}) = \mu(x_1, z_1)\, g(y_1|\,x_1, z_1) \prod_{k=2}^{n} f(x_k, z_k|x_{k-1}, z_{k-1})\, g(y_k|\,x_k, z_k). \tag{4}$$

In this context, we are interested in the sequence of posterior probability densities $\{p(x_{1:n}, z_{1:n}|\,y_{1:n})\}_{n\geq 1}$ which satisfies

$$p(x_{1:n}, z_{1:n}|\,y_{1:n}) = p(x_{1:n}|\,y_{1:n})\, p(z_{1:n}|\,x_{1:n}, y_{1:n})$$

where, for $n \geq 2$, we have

$$p(x_{1:n}|\,y_{1:n}) = \frac{p(x_n, y_n|\,x_{1:n-1}, y_{1:n-1})\, p(x_{1:n-1}|\,y_{1:n-1})}{p(y_n|\,y_{1:n-1})} \tag{5}$$

and

$$p(z_{1:n}|\,x_{1:n}, y_{1:n}) = p(z_{1:n-1}|\,x_{1:n-1}, y_{1:n-1}) \times \frac{g(y_n|\,x_n, z_n)\, f(x_n, z_n|x_{n-1}, z_{n-1})}{p(x_n, y_n|\,x_{1:n-1}, y_{1:n-1})}. \tag{6}$$

For non-linear non-Gaussian state-space models, we do not generally have a closed-form expression for these densities and it is usual to rely on Monte Carlo methods.

Standard particle filters (PF) approximate the associated sequence of probability distributions with weighted empirical distribution associated with a set of N random samples $\left\{X^i_{1:n}, Z^i_{1:n}\right\}_{i=1}^N$ termed particles. When it is possible to obtain closed form expressions for $p(z_{1:n}|\,x_{1:n}, y_{1:n})$ and $p(x_n, y_n|\,x_{1:n-1}, y_{1:n-1})$, but $p(x_{1:n}|\,y_{1:n})$ is unavailable, Rao-Blackwellised particle filters (RBPF) exploit the available structure by approximating only the marginal $p(x_{1:n}|\,y_{1:n})$ through Monte Carlo methods. The Monte Carlo approximation being performed in a space of lower dimension, for common marginal proposals and sample sizes the asymptotic variance of the Rao-Blackwellised particle estimates never exceeds that of standard particle estimates [Chopin, 2004, Theorem 3]. The conditionally linear-Gaussian state-space model [Chen and Liu, 2000, Doucet et al., 2000, 2001, Fearnhead and Clifford, 2003] in which $f(x_n, z_n|x_{n-1}, z_{n-1})$ factorises as

$$f(x_n|x_{n-1})\, \mathcal{N}(z_n; A(x_n)\, z_{n-1}, \Sigma(x_n))$$

and

$$g(y_n|\,x_n, z_n) = \mathcal{N}(y_n; C(x_n)\, z_n, \Xi(x_n))$$

⋆ AMJ was partially supported by EPSRC Grant EP/I017984/1.

is one popular application, for which

$$p(x_n, y_n | x_{1:n-1}, y_{1:n-1}) = f(x_n|x_{n-1})\, p(y_n | x_{1:n}, y_{1:n-1}).$$

Both $p(z_{1:n} | y_{1:n}, x_{1:n})$ and $p(y_n | x_{1:n}, y_{1:n-1})$ are Gaussian densities whose statistics can be computed using Kalman recursions. Another common application is the partially observed Gaussian state-space model [Andrieu and Doucet, 2002, Schön et al., 2005]; in which $f(x_n, z_n|x_{n-1}, z_{n-1}) = \mathcal{N}(z_n; Az_{n-1}, \Sigma)\mathcal{N}(x_n; Cz_n, \Xi)$ and $g(y_n | x_n, z_n) = g(y_n | x_n)$ in which case

$$p(x_n, y_n | x_{1:n-1}, y_{1:n-1}) = g(y_n | x_n)\, p(x_n | x_{1:n-1}).$$

Then $p(z_{1:n} | y_{1:n}, x_{1:n}) = p(z_{1:n} | x_{1:n})$ and $p(x_n | x_{1:n-1})$ are also Gaussian densities whose statistics can be computed using Kalman recursions.

We consider here the scenario in which it is not possible to obtain a closed-form expression for $p(z_{1:n} | x_{1:n}, y_{1:n})$ and $p(x_n, y_n | x_{1:n-1}, y_{1:n-1})$. Hence it is not possible to implement the Rao-Blackwellised particle filter. In this case a strategy one can adopt is to approximate the Rao-Blackwellised particle filter by a hierarchical Monte Carlo algorithm. At the upper level of the hierarchy, we approximate $p(x_{1:n} | y_{1:n})$ using particles $\left\{X^i_{1:n}\right\}_{i=1}^N$ and for each higher level-particle $X^i_{1:n}$ we approximate $p\left(z_{1:n} | X^i_{1:n}, y_{1:n}\right)$ and $p\left(X^i_n, y_n | X^i_{1:n-1}, y_{1:n-1}\right)$ using a "local" particle filter using lower-level particles $\left\{Z^{i,j}_{1:n}\right\}_{j=1}^M$.

A related approach has been proposed in Chen et al. [2011] but their algorithm approximates the ideal algorithm differently and requires that the conditional prior $f(z_n | x_{n-1}, x_n, z_{n-1})$ is known analytically and has cost of $\mathcal{O}(NM^2)$ (the algorithm below requires only the more tractable joint transition and has cost $\mathcal{O}(NM)$). Additionally, it is only guaranteed to provide a consistent estimate of the target distribution $p(x_{1:n}, z_{1:n} | y_{1:n})$ when both $N \to \infty$ *and* $M \to \infty$.

In this paper, our main contribution is to present a new Approximate Rao-Blackwellised particle filter which is applicable to the general class of models defined by (1)-(2)-(3). Our algorithm is "exact" in the sense that it does provide a consistent approximation of $p(x_{1:n}, z_{1:n} | y_{1:n})$ as $N \to \infty$ whatever number, $M \geq 1$, of particles is used by the local particle filters. The proof of this result relies on an extension of the construction presented in Andrieu et al. [2010]. We also propose various methodological extensions. Related ideas have been proposed more recently in scenarios where a static parameter, θ, is the object inferred by the top level algorithm, rather than the Markov process $\{X_n\}_{n\geq 1}$ [Chopin et al., 2011, Fulop and Li, 2011].

An obvious question at this stage is: why implement a Monte Carlo approximation of a Rao-Blackwellised Particle Filter? A Monte Carlo approximation of an algorithm which differs from the standard particle filter in that it admits some analytic integration is not the most obvious construction, it is true. All of the arguments employed to motivate the approach of Chen et al. [2011] could be employed here — in particular, the parallelisation of these algorithms is extremely straightforward and very efficient implementation is possible on parallel architectures by exploiting the fact that a very high proportion of the execution cost is associated with parallelizable steps (cf. Amdahl [1967]). The benefits of parallel implementation of *appropriate* Sequential Monte Carlo algorithms has recently been demonstrated by Lee et al. [2010]. The precise approach which we propose admits a simple importance sampling interpretation which provides an immediate formal justification for the proposed approach (which does not rely on asymptotic arguments beyond those used to justify the standard particle filter).

We have a further motivation for the approach which we propose: it allows for a hierarchically segmented implementation in which dedicated, efficient particle filters of different types are employed at each level. This also allows computational effort to be divided between different parts of the filtering problem, concentrating resources in the more difficult areas. We touch on some specific instances of this type of algorithm in Section 3, in particular when the state space is a product of discrete and continuous components (see Section 3.1). Note that we present here a case in which there are two levels: a top-level and a local-level; in principle a larger number of levels could be employed to allow for a finer factorisation of the state-space. Such a strategy may prove useful when the state space or associated dynamics are complex but can be factorised in such a way that efficient (conditional) particle filters can be implemented.

2. EXACT APPROXIMATION OF RAO-BLACKWELLISED PARTICLE FILTERS

2.1 Rao-Blackwellised Particle Filter

Algorithm 1 Rao-Blackwellised Particle Filter

At time $n = 1$

- Sample $X^i_1 \sim q^x(\cdot | y_1)$ then compute $p\left(z_1 | X^i_1, y_1\right)$ and $p\left(X^i_1, y_1\right)$.
- Compute and normalise the weights

$$w^x_1\left(X^i_1\right) := \frac{p\left(X^i_1, y_1\right)}{q^x\left(X^i_1 | y_1\right)},\ W^{x,i}_1 := \frac{w^x_1\left(X^i_1\right)}{\sum_{k=1}^N w^x_1\left(X^k_1\right)}. \tag{7}$$

At times $n \geq 2$

- Resample $\left\{W^{x,i}_n, \left(X^i_{1:n-1}, p(z_{1:n-1}|X^i_{1:n-1}, y_{1:n-1})\right)\right\}_i$ to obtain $\left\{\frac{1}{N}, \left(\widetilde{X}^i_{1:n-1}, p(z_{1:n-1}|\widetilde{X}^i_{1:n-1}, y_{1:n-1})\right)\right\}_i$.
- Sample $X^i_n \sim q^x(\cdot|\widetilde{X}^i_{1:n-1}, y_{1:n})$ and set $X^i_{1:n} := (\widetilde{X}^i_{1:n-1}, X^i_n)$.
- Compute $p\left(z_{1:n} | X^i_{1:n}, y_{1:n}\right)$ and $p\left(X^i_n, y_n | \widetilde{X}^i_{1:n-1}, y_{1:n-1}\right)$ based on $p\left(z_{1:n-1} | \widetilde{X}^i_{1:n-1}, y_{1:n-1}\right)$ using (6).
- Compute and normalise the weights

$$w^x_n\left(X^i_{1:n}\right) := \frac{p\left(X^i_n, y_n | \widetilde{X}^i_{1:n-1}, y_{1:n-1}\right)}{q^x(X^i_n|\widetilde{X}^i_{1:n-1}, y_{1:n})},$$
$$W^{x,i}_n := \frac{w^x_n\left(X^i_{1:n}\right)}{\sum_{k=1}^N w^x_n\left(X^k_{1:n}\right)}. \tag{8}$$

For completeness, we first describe the standard Rao-Blackwellised particle filter in general terms. Our presentation is slightly non-standard but will generalise naturally later. Note that in order to alleviate the notational burden

we adopt the convention that whenever the index i is used we mean "for all $i \in \{1, ..., N\}$,". This algorithm relies on the importance densities $q^x(x_1|y_1)$ at time 1 and $q^x(x_n|x_{1:n-1}, y_{1:n})$ at times $n \geq 2$.

As presented, Algorithm 1 would require the storage, at time n, of the particles $\left\{X_{1:n}^i, p\left(z_{1:n}|X_{1:n}^i, y_{1:n}\right)\right\}_{i=1}^N$. In most applications of Rao-Blackwellised particle filters, we are interested in estimating only the marginal $p(x_n, z_n|y_{1:n})$ and we can limit ourselves to storing $\left\{X_n^i, p\left(z_n|X_{1:n}^i, y_{1:n}\right)\right\}_{i=1}^N$ where $p\left(z_n|X_{1:n}^i, y_{1:n}\right)$ is characterized by a fixed-dimensional sufficient statistic.

2.2 Approximate Rao-Blackwellised Particle Filter

Consider now the scenario in which it is not possible to obtain closed-form expressions for $p(z_{1:n}|x_{1:n}, y_{1:n})$ and $p(x_n, y_n|x_{1:n-1}, y_{1:n-1})$. Hence it is not possible to implement the Rao-Blackwellised particle filter described above. However, conditional upon a particle $X_{1:n}^i$, we can run a "local" particle filter using particles $\left\{Z_{1:n}^{i,j}\right\}_{j=1}^M$ to approximate $p\left(z_{1:n}|X_{1:n}^i, y_{1:n}\right)$ and $p\left(X_n^i, y_n|\widetilde{X}_{1:n-1}^i, y_{1:n-1}\right)$ as originally suggested in Chen et al. [2011]. These local particle filters rely on the importance distributions $q^z(z_1|x_1, y_1)$ at time 1 and $q^z(z_n|x_{1:n}, y_{1:n}, z_{1:n-1})$ at times $n \geq 2$. Local importance weights are expressed in terms of the readily-evaluated joint distributions:

$$p(x_1, y_1, z_1) = g(y_1|x_1, x_1)\,\mu(x_1, z_1) \text{ and}$$
$$p(x_n, y_n, z_n|x_{n-1}, z_{n-1}) = f(x_n, z_n|x_{n-1}, z_{n-1})g(y_n|x_n, z_n).$$

Whenever the indices i, j are used, the statement applies "for all $i \in \{1, \ldots, N\}$ and $j \in \{1, ..., M\}$, respectively".

Algorithm 2 shows the proposed algorithm in detail. As presented, it would require that, at time n, the particles $\left\{X_{1:n}^i\right\}_{i=1}^N$ and $\left\{Z_{1:n}^{i,j}\right\}_{i=1,j=1}^{N,M}$ all be available in memory. In most applications, we are interested in estimating only the marginal $p(x_n, z_n|y_{1:n})$ and we can limit ourselves to storing $\left\{X_n^i\right\}_{i=1}^N$ and $\left\{Z_n^{i,j}\right\}_{i=1,j=1}^{N,M}$. In this scenario, we can think of $\left\{Z_n^{i,j}\right\}_{j=1}^M$ as the finite-dimensional "sufficient" statistics associated with $p\left(z_n|X_{1:n}^i, y_{1:n}\right)$.

2.3 Validity of The ARBPF Algorithm

The validity of the approximate Rao-Blackwellised particle filter (ARBPF) is established by showing that it can be reinterpreted as a standard particle algorithm on an extended space targetting a sequence of distributions, the n^{th} of which admits the marginal posterior $p(x_{1:n}|y_{1:n})$ as a marginal. Furthermore, it is straightforward to obtain an estimate of the joint posterior $p(x_{1:n}, z_{1:n}|y_{1:n})$. The arguments used here are based on a slight extension of the construction proposed in Andrieu et al. [2010]; see also Chopin et al. [2011].

Preliminaries We require notation and an interpretation of the algorithm which allows us to keep track of the full collection of random variables used during the running of the algorithm. This is a formal representation of precisely

Algorithm 2 Approximate Rao-Blackwellised PF

At time $n = 1$

- Sample $X_1^i \sim q^x(\cdot|y_1)$.
- Sample $Z_1^{i,j} \sim q^z\left(\cdot|X_1^i, y_1\right)$.
- Compute and normalise the local weights

$$w_1^z\left(X_1^i, Z_1^{i,j}\right) := \frac{p(X_1^i, y_1, Z_1^{i,j})}{q^z\left(Z_1^{i,j}\middle|X_1^i, y_1\right)}$$

$$W_1^{z,i,j} := \frac{w_1^z\left(X_1^i, Z_1^{i,j}\right)}{\sum_{k=1}^N w_1^z\left(X_1^i, Z_1^{i,k}\right)},$$

$$\text{define } \widehat{p}(X_1^i, y_1) := \frac{1}{M}\sum_{j=1}^M w_1^z\left(X_1^i, Z_1^{i,j}\right).$$

- Compute and normalise the top-level weights

$$w_1^x\left(X_1^i\right) := \frac{\widehat{p}(X_1^i, y_1)}{q^x\left(X_1^i|y_1\right)},\ W_1^{x,i} := \frac{w_1^x\left(X_1^i\right)}{\sum_{k=1}^N w_1^x\left(X_1^k\right)}. \tag{9}$$

At times $n \geq 2$

- Resample

$$\left\{W_{n-1}^{x,i}, \left(X_{1:n-1}^i, \left\{W_{n-1}^{z,i,j}, Z_{1:n-1}^{i,j}\right\}_j\right)\right\}_i$$

to obtain

$$\left\{\frac{1}{N}, \left(\widetilde{X}_{1:n-1}^i, \left\{\overline{W}_{n-1}^{z,i,j}, \overline{Z}_{1:n-1}^{i,j}\right\}_j\right)\right\}_i.$$

- Resample $\{\overline{W}_{n-1}^{z,i,j}, \overline{Z}_{1:n-1}^{i,j}\}_j$ to obtain $\{\frac{1}{M}, \widetilde{Z}_{1:n-1}^{i,j}\}_j$.
- Sample $X_n^i \sim q^x(\cdot|\widetilde{X}_{1:n-1}^i, y_{1:n})$; set $X_{1:n}^i := (\widetilde{X}_{1:n-1}^i, X_n^i)$.
- Sample $Z_n^{i,j} \sim q^z\left(\cdot|X_{1:n}^i, y_{1:n}, \widetilde{Z}_{1:n-1}^{i,j}\right)$; set $Z_{1:n}^{i,j} := (\widetilde{Z}_{1:n-1}^{i,j}, Z_n^{i,j})$.
- Compute and normalise the local weights

$$w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,j}\right) := \frac{p\left(X_n^i, y_n, Z_n^{i,j}\middle|\widetilde{X}_{n-1}^i, \widetilde{Z}_{n-1}^{i,j}\right)}{q^z\left(Z_n^{i,j}\middle|X_{1:n}^i, y_{1:n}, \widetilde{Z}_{1:n-1}^{i,j}\right)},$$

$$\widehat{p}(X_n^i, y_n|\widetilde{X}_{1:n-1}^i, y_{1:n-1}) := \frac{1}{M}\sum_{j=1}^M w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,j}\right),$$

$$W_n^{z,i,j} := \frac{w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,j}\right)}{\sum_{k=1}^M w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,k}\right)}.$$

- Compute and normalise the top-level weights

$$w_n^x\left(X_{1:n}^i\right) := \frac{\widehat{p}(X_n^i, y_n|\widetilde{X}_{1:n-1}^i, y_{1:n-1})}{q^x(X_n^i|\widetilde{X}_{1:n-1}^i, y_{1:n})},$$
$$W_n^{x,i} := \frac{w_n^x\left(X_{1:n}^i\right)}{\sum_{k=1}^N w_n^x\left(X_{1:n}^k\right)}. \tag{10}$$

This algorithm reduces to a standard particle filter when $M = 1$ and approaches the RBPF when $M \to \infty$.

the algorithm described in the previous section which is not required in the implementation of ARBPFS.

To describe the full set of variables used in the execution of the particle filter it is useful to view resampling as a process by which each of the particles, at the beginning of iteration n, draws a parent indicator from an appropriate distribution. The parents of the top-level particles at time n can be denoted $A_{n-1}^{x,i}$ with i running from 1 up to N; whilst, for any given i, we have M parent indicators for the local particles at this generation, denoted $A_{n-1}^{z,A_{n-1}^{x,i},j}$ where the multiple indexing arises because the parents of the local particles are attached to the *parent* of the associated top-level particle and leads to a consistent interpretation of the two sets of indices. In this description, the variable $A_{n-1}^{x,i}$ represents the index of the 'parent' at time $n-1$ of particle $X_{1:n}^i$ whereas $A_{n-1}^{z,A_{n-1}^{x,i},j}$ represents the index of the 'parents' of particle $Z_{1:n}^{i,j}$; i.e. particle $Z_{1:n}^{i,j}$ has two parents, a high-level 'parent' corresponding to the local particle filter $A_{n-1}^{x,i}$ associated to $X_{1:n}^i$ it descends from and a low-level 'parent' which is a particle within this local particle filter. In the case of multinomial resampling, we can view the top-level resampling as drawing

$$(A_{n-1}^{x,i})_{i=1}^N \sim \mathcal{M}\left(N, W_{n-1}^{x,1:N}\right),$$

and the low-level resampling admits a similar representation, for each $i \in \{1,\ldots,N\}$:

$$(A_{n-1}^{z,A_{n-1}^{x,i},j})_{j=1}^M \sim \mathcal{M}\left(M, W_{n-1}^{z,A_{n-1}^{x,i},1:M}\right)$$

where $\mathcal{M}(\cdot|\, n, \mathbf{p})$ denotes the multinomial distribution of parameters n and $\mathbf{p} = (p_1, ..., p_J)$.

To connect this with the algorithmic description, we may identify:

$$\widetilde{X}_{n-1}^i = X_{n-1}^{A_{n-1}^{x,i}} \qquad \widetilde{Z}_{n-1}^{i,j} = Z_{n-1}^{A_{n-1}^{x,i},A_{n-1}^{z,A_{n-1}^{x,i},j}}$$

This provides a slightly finer description than is required to justify the algorithm, but such a representation is required in order to employ the algorithm within a Particle MCMC framework (see section 3.2).

Remark: Other resampling schemes, including those in which resampling is performed at times which depend on the particle configuration (the arguments of Del Moral et al. [2012] apply directly in the present setting), may be used and justified by similar arguments to those used here; the use of multinomial resampling slightly simplifies the notation and clarifies the core argument but is far from essential. Note that by resampling the top level particles and then resampling *within* the local particle filters, it is possible to attach a genealogical tree structure to the algorithm: the top-level filter evolves through time as a collection of random trees just as the standard particle filter does; each root-leaf path through this collection of random trees has a collection of random trees associated with it describing the evolution of the local particle filters. The local random trees for any collection of top-level particles which coincide for part of their evolution also coincide for that part of their temporal evolution.

Marginal Target Distribution We first establish the form of the target distribution using a simple importance sampling argument which matches the intuition behind the development of the algorithm, before considering a more subtle construction which gives a more precise characterisation of the algorithm.

We initially ignore resampling of the top-level particles $\left\{X_n^i\right\}_{i=1}^N$ and consider the resulting sequential importance sampling (SIS) algorithm. In this case, the importance sampling (proposal) distribution at time n is given by

$$\begin{aligned} &q_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| y_{1:n}\right) \\ &= q_n\left(x_{1:n} | y_{1:n}\right) q_n\left(a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| x_{1:n}, y_{1:n}\right) \end{aligned} \tag{11}$$

where

$$q_n\left(x_{1:n} | y_{1:n}\right) = q^x\left(x_1 | y_1\right) \prod_{m=2}^n q^x\left(x_m | x_{1:m-1}, y_{1:m}\right)$$

and, if we assume that *multinomial resampling* is used within the local particle filters, then the conditional distribution $q_n\left(a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| x_{1:n}, y_{1:n}\right)$ may be expanded as:

$$\prod_{j=1}^M \left\{ q^z\left(z_1^j \middle| x_1, y_1\right) \prod_{m=2}^n W_{m-1}^{z,a_{m-1}^{z,j}} q^z\left(z_m^j \middle| x_{1:m}, y_{1:m}, z_{1:m-1}^{a_{m-1}^{z,j}}\right) \right\}.$$

We have suppressed the i-index in this description as it arises directly from drawing N iid samples from this distribution. The importance weight function associated with this distribution is the product of the incremental weights (9)-(10), that is

$$\begin{aligned} &w_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M}\right) = \\ &\qquad \frac{\widehat{p}(x_1, y_1)}{q^x\left(x_1 | y_1\right)} \prod_{m=2}^n \frac{\widehat{p}\left(x_m, y_m | x_{1:m-1}, y_{1:m-1}\right)}{q^x(x_m | x_{1:m-1}, y_{1:m})}, \end{aligned} \tag{12}$$

in the absence of top-level resampling. The target distribution at time n corresponds to the product of the proposal distribution and the importance weight function:

$$\begin{aligned} &\pi_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M}\right) \\ &\propto w_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M}\right) q_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| y_{1:n}\right) \\ &\propto \widehat{p}(x_{1:n}, y_{1:n}) q_n\left(a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| x_{1:n}, y_{1:n}\right) \end{aligned}$$

where

$$\widehat{p}(x_{1:n}, y_{1:n}) := \widehat{p}(x_1, y_1) \prod_{m=2}^n \widehat{p}\left(x_m, y_m | x_{1:m-1}, y_{1:m-1}\right)$$

It is now well-known that the marginal likelihood estimate provided by a particle filter is unbiased [Del Moral, 2004]; i.e.

$$\begin{aligned} &\sum_{a_{1:n-1}^{z,1:M}} \int \widehat{p}(x_{1:n}, y_{1:n})\; q_n\left(a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M} \middle| y_{1:n}, x_{1:n}\right) dz_{1:n}^{1:M} \\ &= p\left(x_{1:n}, y_{1:n}\right). \end{aligned}$$

It follows straightforwardly that the marginal target is:

$$\begin{aligned} \pi_n\left(x_{1:n}\right) &= \sum_{a_{1:n-1}^{z,1:M}} \int \pi_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M}\right) dz_{1:n}^{1:M} \\ &= p\left(x_{1:n} | y_{1:n}\right). \end{aligned}$$

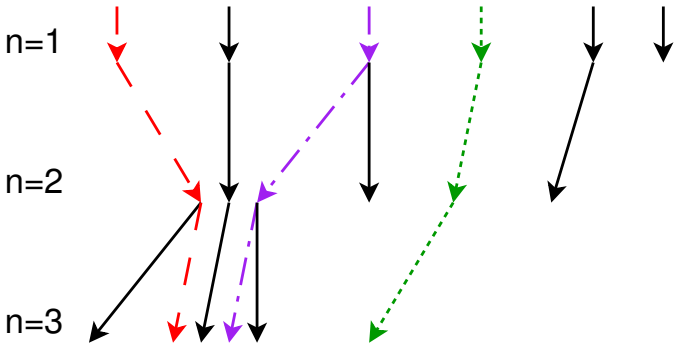

Fig. 1. Ancestral lines: the b-formulation, $n = 3$. In this cartoon representation, the particle indices coincide with their position in the horizontal ordering and now trajectories cross. We have $b_3^2 = 2$, $b_2^2 = 1$ and $b_1^2 = 1$ for the dashed trajectory; $b_3^4 = 4$, $b_2^4 = 3$ and $b_1^4 = 3$ for the dash-dotted trajectory and $b_3^6 = 6$, $b_2^6 = 5$ and $b_1^6 = 4$ in the case of the dotted path.

Joint Target Distribution It is possible to obtain a more precise result by considering explicitly the joint artificial target distribution. After manipulations similar to those of Andrieu et al. [2010], Chopin et al. [2011], we obtain

$$\pi_n\left(x_{1:n}, a_{1:n-1}^{z,1:M}, z_{1:n}^{1:M}\right) = \frac{p(x_{1:n}|y_{1:n})}{M^{n-1}} \times \frac{1}{M}\sum_{j=1}^{M} p\left(z_{1:n}^j \middle| x_{1:n}, y_{1:n}\right) \left\{\prod_{k=1,k\neq b_1^j}^{M} q^z\left(z_1^k \middle| x_1, y_1\right)\right\} \times \prod_{m=2}^{n}\left\{\prod_{k=1,k\neq b_m^j}^{M} W_{m-1}^{a_{m-1}^{z,k}} q^z\left(z_m^k \middle| x_{1:m}, y_{1:m}, z_{1:m-1}^{a_{m-1}^{z,k}}\right)\right\}$$

where $\left(b_1^j, ..., b_n^j\right)$ is the ancestral lineage of $z_{1:n}^j$ which is implicitly dependent upon the current iteration, n, and is defined recursively: $b_n^j = j$ and $b_k^j = a_k^{z,b_{k+1}^j}$; that is $z_{1:n}^j = \left(z_1^{b_1^j}, z_2^{b_2^j}, ..., z_n^{b_n^j}\right)$. This notation is simpler than it may at first seem and is illustrated in Figure 1. It is possible to simply modify this target to introduce an additional discrete random variable $L \in \{1, ..., M\}$ such that

$$\pi_n\left(x_{1:n}, a_{1:n-1}^{1:M}, l, z_{1:n}^{1:M}\right) = \frac{p\left(x_{1:n}, z_{1:n}^l \middle| y_{1:n}\right)}{M^n}\left\{\prod_{k=1,k\neq b_1^j}^{M} q^z\left(z_1^k \middle| x_1, y_1\right)\right\} \times \prod_{m=2}^{n}\left\{\prod_{k=1,k\neq b_m^j}^{M} W_{m-1}^{a_{m-1}^{k}} q^z\left(z_m^k \middle| x_{1:m}, y_{1:m}, z_{1:m-1}^{a_{m-1}^{z,k}}\right)\right\}.$$

This provides use with an estimate of the joint distribution of interest $p\left(x_{1:n}, z_{1:n} \middle| y_{1:n}\right)$.

Both of these descriptions of the algorithm without the top level resampling step clearly justify the use of the algorithm *with* resampling by the usual arguments underlying the sequential importance resampling algorithm (see Del Moral [2004] for a detailed study). A precise characterisation of the algorithm with both resampling steps is possible, indeed it follows by applying essentially the same construction described here at the top level of the algorithm. However, the resulting expressions are rather cumbersome and are not necessary in order to implement or understand the basic algorithm.

It should be noted that in addition to allowing standard results to be applied directly, interpreting the algorithm as a simple SIR algorithm on a suitably augmented state also illustrates that in the context of online inference, this hierarchical approach to particle filtering will not require that the size of the local particle filters increases with the length of the state sequence to maintain a given quality of approximation of the filtering distributions, in contrast to the parameter estimation case [Chopin et al., 2011].

3. EXTENSIONS AND VARIATIONS

For clarity of presentation we have employed only a very simple particle filter in the above description of the algorithm. It should be understood that the vast majority of extensions to the particle filter which have appeared in the literature could be very easily included in ARBPFs. Incorporating MCMC moves to improve sample diversity Gilks and Berzuini [2001] is straightforward (indeed particle MCMC-type moves could be employed to rejuvenate the subsidiary particle filters if required).

In particular, the precise structure of the top-level particle filter does not enter into the construction used to justify the algorithm and indeed any particle filtering algorithm could be used at this level; at the lower level we require only that we are able to represent the algorithm in the appropriate form. In this section we sketch a number of particularly interesting directions which warrant further study (and are currently under investigation).

3.1 Exact Approximation of Discrete Particle Filters

One practically important scenario is that in which $\mathcal{X}$, the space X_n takes values in, is a finite set. In this case, a very efficient discrete particle can be implemented at the top-level by using the algorithm of Fearnhead and Clifford [2003] within the ARBPF framework described above. As our construction does not depend upon any particular properties of the top-level algorithm this leads directly to a valid algorithm.

This approach to particle filtering for small discrete state spaces has been found to be highly efficient but it has not, until now, been obvious how to combine it with other strategies in order to deal with mixed continuous-discrete state spaces (for example, switching models with complex, $\{Z_n\}$-dynamics in a continuous state space that cannot be analytically marginalised which depend upon the current value of a regime-indicator, X_n, that is well-modelled by a finite-space Markov chain). The possibility of this combination is one of the most promising avenues of research arising from the ARBPF construction.

3.2 Approximate Rao-Blackwellised Particle Filters for Particle Markov chain Monte Carlo Methods

As the ARBPF can be viewed as an algorithm for generating a large collection of random variables (describing all of the trajectories generated during the course of the running of the algorithm and all of the ancestral lines associated with the resampling of all particles, including those which

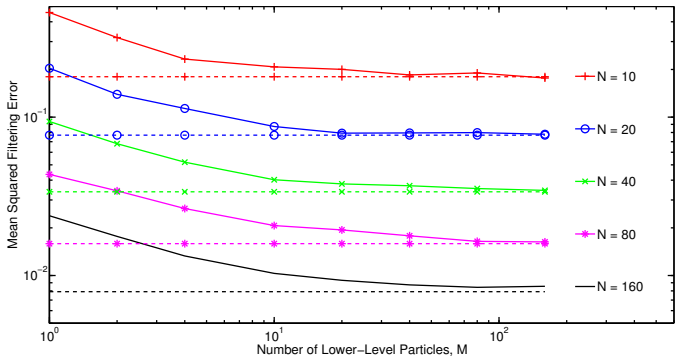

Fig. 2. Monte Carlo Mean Squared Error of ARBPFs and RBPFs. Dashed lines show RBPF errors for each N; solid lines ARBPF errors as a function of M.

have been eliminated before the current time) it is possible to use arguments similar to those of Andrieu et al. [2010] to embed ARBPF proposals within a PMCMC framework, using the trajectories obtained from the PF algorithm as proposals within an MCMC framework.

Further extensions are also possible: It is naturally possible to combine Particle MCMC with the ABRPF algorithm in settings with partially discrete state spaces, embedding the efficient discrete particle filter within the PMCMC algorithm by employing a construction of the sort considered by Whiteley et al. [2010]. It may also be fruitful to embed ARBPFs within the SMC^2 approach [Chopin et al., 2011].

3.3 Other Settings

Similar techniques can also be employed in a number of other settings. In addition to those which have already been discussed in the literature, the use of "local" particle filters to provide approximations of proposal distributions within a block-sampling framework is effective and can be justified as a standard Sequential Monte Carlo algorithm defined upon an extended space by employing further extensions of the auxiliary variable construction used here [Johansen and Doucet, 2012].

4. EXAMPLE

A simple example demonstrates that the performance of the ABRPF rapidly approaches that of the corresponding RBPF (the ideal algorithm it approximates) as M increases. We compare the Monte Carlo approximations to the Kalman filter to quantify the Monte Carlo error.

We use a simulated sequence of 100 observations from the model defined by the densities:

$$\begin{aligned}\mu(x_1, z_1) =& \mathcal{N}\left((x_1, z_1)^{\mathrm{T}}; \mathbf{0}, I_2\right)\\ f(x_n, z_n | x_{n-1}, z_{n-1}) =& \mathcal{N}\left((x_n, z_n)^{\mathrm{T}}; (x_{n-1}, z_{n-1})^{\mathrm{T}}, I_2\right)\\ g(y_n | x_n, z_n) =& \mathcal{N}\left(y_n; (x_n, z_n)^{\mathrm{T}}, I_2\right)\end{aligned}$$

where I_2 denotes the 2×2 identity matrix. We consider the time-averaged squared deviation between the posterior mean filtering estimates of the first coordinate provided by the ARBPF/RBPF and the mean of the optimal filter (i.e. the error arising from the Monte Carlo approximation). We consider only the "bootstrap" ARBPF/RBPF (i.e. using the joint/marginal prior as the proposal) with systematic resampling employed (at both levels in the ARBPF case) every iteration. Figure 2 shows values averaged over 100 replicate of each filter as well as over time. Even small values of M can yield good performance. This illustrates the validity of the algorithm and lends some support to its use in parallel settings; optimal implementations for real problems are under consideration.

REFERENCES

G. Amdahl. Validity of the single processor approach to achieving large-scale computing capabilities. In *Proc. of AFIPS*, volume 30, pages 483–485. ACM, 1967.

C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *J. R. Stat. Soc. Ser. B*, 64(4):827–836, 2002.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo. *J. R. Stat. Soc. Ser. B*, 72(3):269–342, 2010.

R. Chen and J. S. Liu. Mixture Kalman filters. *J. R. Stat. Soc. Ser. B*, 62(3):493–508, 2000.

T. Chen, T. Schön, H. Ohlsson, and L. Ljung. Decentralized particle filter with arbitrary state decomposition. *IEEE Trans. Sig. Proc.*, 59(2):465–478, 2011.

N. Chopin. Central limit theorem for sequential Monte Carlo methods and its applications to Bayesian inference. *Ann. Stat.*, 32(6):2385–2411, 2004.

N. Chopin, P. Jacob, and O. Papaspiliopoulos. SMC^2. *ArXiv Mathematics e-prints*, 1101.1528, 2011.

P. Del Moral. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Probability and Its Applications. Springer Verlag, New York, 2004.

P. Del Moral, A. Doucet, and A. Jasra. On adaptive resampling procedures for sequential Monte Carlo methods. *Bernoulli*, 18(1):252–278, 2012

A. Doucet, S. Godsill, and C. Andrieu. On sequential simulation-based methods for Bayesian filtering. *Stat. and Comput.*, 10(3):197–208, 2000.

A. Doucet, N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Trans. Sig. Proc.*, 49:613–624, 2001.

P. Fearnhead and P. Clifford. On-line inference for hidden Markov models via particle filters. *J. R. Stat. Soc. Ser. B*, 65(4):887–899, 2003.

A. Fulop and J. Li. Robust and efficient learning: A marginalize resample-move approach. Techn. rep., ESSEC Busines School, 2011.

W. R. Gilks and C. Berzuini. Following a moving target – Monte Carlo inference for dynamic Bayesian models. *J. R. Stat. Soc. Ser. B*, 63:127–146, 2001.

A. M. Johansen and A. Doucet Mitigating Path Degeneracy: A Local Particle Filter Approach *CRiSM Working Paper*, in preparation, 2012 .

A. Lee, C. Yau, M. B. Giles, A. Doucet, and C. C. Holmes. On the utility of graphics card to perform massively parallel simulation with advanced Monte Carlo methods. *J. Comput. Graph. Stat.*, 19(4):769–789, 2010.

T. Schön, F. Gustafsson, and P. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Trans. Sig. Proc.*, 53:2279–2289, 2005.

N. Whiteley, C. Andrieu, and A. Doucet. Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods. Res. Rep. 10:04, University of Bristol, 2010.