Dept. of Math. University of Oslo
Statistical Research Report No 1
ISSN 0806–3842 October 2013

Particle move-reweighting strategies for online inference

R. Marques[1], G. Storvik[1]

Abstract

Sequential Monte Carlo (SMC) methods are one of the most important computational tool to deal with intractability in complex statistical models. In those techniques, the distribution of interest is approximated by a set of properly weighted samples. One problem with SMC algorithms is the weights degeneracy: either the weights have huge variability or high correlations between the particles. Updating the particles by a few MCMC steps has been suggested as an improvement in this case (the resample-move algorithm). The general setup is to first resample the particles in such a way that all particles are given equal weight. Thereafter the MCMC steps are applied in order to make the identical samples diverge. In this work we consider an alternative strategy where the order of MCMC updates and the resampling steps are switched, i.e. MCMC updates are performed first. The main advantage with such an approach is that by performing MCMC updates, the weights can be updated simultaneously, making them less variable. We illustrate through simulation studies how our methodology can give improved results for online Bayesian inference in general state space models.

Keywords: Approximate inference; Particle filter; MCMC moves; reweighting schemes; Sequential Monte Carlo methods; State space models

[☆]email of correspondence: geirs@math.uio.no
[1]University of Oslo and Statistics for Innovation Centre, Oslo, Norway.

1. Introduction

In many real-world applications, observations arrive sequentially in time, in which cases state space models provide flexible representations for stochastic dynamical systems. Such models are applied in a wide range of fields, e.g. financial econometrics, ecology, geo(neuro)-science, engineering or machine learning [12, 21, 23].

Sequential Monte Carlo (SMC) methods are an efficient class of algorithms to sample from probability distributions of interest. Their respective algorithms –called *particle filters* (PF)– are exhaustively used to carry out inference of intractable state space models. By employing sequential Monte Carlo (SMC) methods, the target distribution is approximated by a set of weighted samples, generated sequentially from some proposal distributions [see 12, 3, 16, 14, and the references therein].

A well-known problem in SMC methods is *weight degeneracy*, also called sample impoverishment. This problem is related to the increasing variance of the particle weights over time [13, 12]. Therefore, after a few iterations there are only a small number of particles left which adequately characterize the target distribution. Consequently, the Monte Carlo estimators will gradually loose their good statistical proprieties.

Since the seminal paper on SMC methods by [20], the introduction of the resample stage poses an inexpensive alternative to avoid the collapse of particle filter algorithms due to weight degeneracy. The idea of the resample step consists in sampling the particles randomly by duplicating the ones with high weights and removing those with low weights. Afterwards, the weights take equal values. Under mixing conditions, SMC methods with resampling provide estimates of *marginal* distributions whose variance is uniformly bounded with time [9, 7, 14]. However, the successive use of the resample procedure can affect significantly the distinct number of particles, and consequently SMC methods tend to loose sample diversity when considering *simultaneous* distributions or variables with slow mixing.

In order to rejuvenate the particles after the resample step, MCMC kernels were successfully introduced to build proposal distributions in particle filter algorithms [19]. Basically, after the resample stage, the addition of MCMC moves allows the particles to move through the sample space and rejuvenate. Gilks and Berzuini [19] and Chopin [7] provide a formal justification for such schemes, including theoretical results focused on consistency and asymptotic normality. Later, [11] and [26] present alternative strategies to reduce weight degeneracy in particle filter algorithms. Other approaches, such as dynamic parameter estimation using sufficient statistics [15, 41] can also be considered as particle filters with MCMC moves. A restriction for these approaches is however that resampling has to be made before a move, restricting the possibility for adaptive resampling.

In this paper, we propose a flexible strategy that allows for MCMC moves without the need of a preliminary resampling step. Following the MCMC moves, we *update* the particle weight taking into account the diversification step and that MCMC moves give particles closer to the target distribution. The effect of this is that resample stages can be delayed. With this strategy, we are able to diversify the particles via an MCMC move and, at the same time, reduce the weight degeneracy. The validity of this approach is based on the commonly used trick of working on an artificial extended distribution having the target distribution as marginal combined with the use of backwards kernels, introduced in [10] for static problems and considered in more general settings in [42]. Due to the flexibility in choices of backwards kernels, many updating schemes for particle weights can be considered. We will discuss different alternatives.

This paper is organized as follows. Section 2 gives a brief overview of the Sequential Monte Carlo methods and the particle filter algorithm with the MCMC move step. Section 3 presents the move-reweighting approach as an efficient way to combine diversification with updating of the weights.

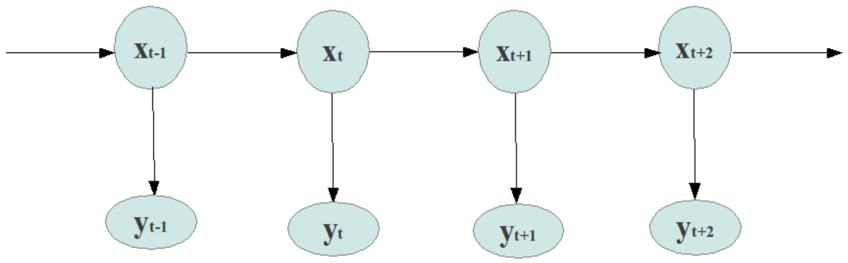

Figure 1: State-space model described by graphical structure.

Section 4 demonstrates two applications of our methodology for sequential Bayesian inference: a non-Gaussian likelihood model with time varying parameter, and a non-linear Gaussian stochastic system. Finally, section 5 closes with a discussion.

2. Sequential Inference in State Space Models

2.1. Preliminaries: notation and model description

Let $\{\mathbf{x}_t\}_{t\in\mathbb{N}}$ and $\{\mathbf{y}_t\}_{t\in\mathbb{N}}$ be discrete-time stochastic processes in which the latent process $\mathbf{x}_t$ is indirectly observed through the measurement data $\mathbf{y}_t$. As traditionally used in the literature, the generic state-space models are expressed in terms of a dynamic process model in combination with an observation model:

$$\begin{aligned} \mathbf{x}_0 &\sim \pi(\mathbf{x}_0); && \text{Initial Distribution} \\ \mathbf{x}_t|\mathbf{x}_{1:t-1} &\sim \pi(\mathbf{x}_t|\mathbf{x}_{t-1}); && \text{Prior Latent Model} \\ \mathbf{y}_t|\mathbf{y}_{1:t-1},\ \mathbf{x}_{1:t} &\sim \pi(\mathbf{y}_t|\mathbf{x}_t), && \text{Observation Model} \end{aligned}$$

where here and in the following $\pi(\cdot)$ is used generically for distributions based on the assumed state-space model with $\cdot$ specifying which variables that are in question. In general the distributions will depend on some parameters $\boldsymbol{\theta}$. These will be assumed parameters known and are therefore not included in the notation. We will also use the notation $\mathbf{x}_{1:t} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t) \in \mathcal{X}_t$ and $\mathbf{y}_{1:t} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t) \in \mathcal{Y}_t$ to denote the first t individuals of the sequence of latent and observed variables. For simplicity of notation we assume each $\mathbf{x}_t$ have a common sample space $\mathcal{X} \subset \mathbb{R}^{p_x}$ and we denote $\mathcal{X}_t = \prod_{k=1}^t \mathcal{X}$ and similarly $\mathcal{Y}_t = \prod_{k=1}^t \mathcal{Y}$ for $\mathcal{Y} \subset \mathbb{R}^{p_y}$. Figure 1 describes the dynamic hierarchical structure between the hidden Markov process and the observations using a graphical configuration.

Given the data $\mathbf{y}_{1:t}$ up to time t, inference focuses on the posterior distribution

$$p(\mathbf{x}_{1:t}) \equiv \pi(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \propto \pi(\mathbf{x}_1)\pi(\mathbf{y}_1|\mathbf{x}_1) \prod_{k=2}^{t} \pi(\mathbf{x}_k|\mathbf{x}_{k-1})\pi(\mathbf{y}_k|\mathbf{x}_k). \tag{1}$$

When $p(\mathbf{x}_{1:t})$ is intractable, Monte Carlo sampling methods can be applied to carry out an approximate inference. Algorithms based on MCMC schemes are traditional stochastic techniques to sample from high-dimensional distributions. In many cases, including sequential inference, these methods do not perform well when strong temporal correlations are present [33, 6, 10]. Within statistical Monte Carlo techniques, SMC methods (and also combined with MCMC sampling) have become a powerful tool to perform static and dynamic inference in complex or high-dimension models [see 16, 1, 8, and the references therein].

2.2. Sequential Monte Carlo methods

SMC methods are a broad class of Monte Carlo integration methods based on importance sampling techniques which decompose the target distribution in a sequence of low dimensional distributions. For each time t, SMC methods represent the posterior $\pi(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$ in (1) using a finite collection of properly weighted samples $\{(\mathbf{x}_t^i, w_t^i), i = 1, \dots, N\}$ with $N \gg 1$, according to the following definition [30, 32]:

Definition 1. *A set of weighted random samples* $\{(\mathbf{x}^i, w^i), i = 1, \dots, N\}$ *is called proper (or properly weighted) with respect to* p *if, for any square integrable function* g*,*

$$E_q[g(\mathbf{x}^i)w^i] = cE_p[g(\mathbf{x}^i)], \quad \textit{for} \quad i = 1, \dots, N,$$

for some normalizing constant c *common to all the* N *samples generated form* q*.*

The population of samples are called particles, and in the case where $\mathbf{x} = \mathbf{x}_{1:t}$ they are typically generated sequentially from some low-dimensional conditional distributions

$$q(\mathbf{x}_{1:t}) = q(\mathbf{x}_1) \prod_{k=2}^{t} q(\mathbf{x}_k|\mathbf{x}_{k-1}).$$

where $q(\cdot)$ is generically used for proposal distributions. Assuming that $q(\mathbf{x}_{1:t}) > 0$ for all $\mathbf{x}_{1:t}$ with $p(\mathbf{x}_{1:t}) > 0$, then based on sequential importance sampling ideas, particle weights are defined as

$$w_t^i \equiv w_t(\mathbf{x}_{1:t}^i) = \frac{p(\mathbf{x}_{1:t}^i)}{q(\mathbf{x}_{1:t}^i)} \propto w_{t-1}^i \frac{\pi(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)\pi(\mathbf{y}_t|\mathbf{x}_t^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}$$

allowing for recursive computation. Estimation is usually based on normalized weights, making the proportionality constant unnecessary to compute. For time $t \geqslant 1$ when new data arrives, the particle filter algorithm provides an online estimate of $E_p[g(\mathbf{x}_{1:t})|\mathbf{y}_{1:t}]$ through

$$\sum_{i=1}^{N} w_t^i g(\mathbf{x}_{1:t}^i) / \sum_{i=1}^{N} w_t^i$$

which is consistent with probability one [12, 7]. In addition, particle filters provide, as a natural byproduct, an unbiased estimator for the marginal likelihood [9, 34]. Such likelihood estimates have been used in different contexts, with emphasis on parameter learning, model selection and building particle proposals for sequential inference [see e.g. 34, 1, 37, 8, 43, 39].

Taking the prior $\pi(\mathbf{x}_t|\mathbf{x}_{t-1})$ as the proposal distribution (the Bootstrap filter by [20]), the incremental weights boils down to evaluating the likelihood for the observation. Despite that the Bootstrap filter is easy to implement, better proposals can give substantial improvements in the efficiency of SMC methods. [13] show that the optimal proposal is the conditional distribution given

Algorithm 1 Ordinary SMC for Filtering

```
**At** $t = 1$
  Sample $\mathbf{x}_1^i \sim q(\mathbf{x}_1^i|\mathbf{y}_1)$ for $i = 1, ..., N$.
  Compute and normalize the weights

    $w_1^i = \pi(\mathbf{x}_1^i)\pi(\mathbf{y}_1|\mathbf{x}_1^i)/q(\mathbf{x}_1^i|\mathbf{y}_1).$

**for** $t = 2, 3, ...$ **do**
  Propagate $\mathbf{x}_t^i \sim q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_t)$ for $i = 1, ..., N$.
  Compute and normalize the weights

    $w_t^i = w_{t-1}^i \frac{\pi(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)\pi(\mathbf{y}_t|\mathbf{x}_t^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_t)}.$

  **if** Effective sample size is small **then**
    Resample to obtain $N$ new equally-weighted particles.
**end for**
```

the previous state and the current observation, i.e., $q^{\mathsf{opt}}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) = \pi(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$. However, the optimal choice is not available in many applications, making it necessary with some approximation (eg. deterministic sampling methods or local linearisation proposals). Alternative choices of q are presented in [5] and [35].

An efficient algorithm should be able to produce particle weights with low variance. Even though the use of clever proposals close to q^{opt} tends to decrease the variance, the weight degeneracy problem will be unavoidable. The next proposition [see proof in 28] deals with how dimension increase influences the variance of the weights.

Proposition 1. *Let $\pi(\mathbf{v}_1, \mathbf{v}_2)$ and $q(\mathbf{v}_1, \mathbf{v}_2)$ be two probability densities, where the support of π is a subset of the support of q. Then,*

$$\mathsf{Var}_q\left[\frac{\pi(\mathbf{v}_1, \mathbf{v}_2)}{q(\mathbf{v}_1, \mathbf{v}_2)}\right] \geq \mathsf{Var}_{q_1}\left[\frac{\pi_1(\mathbf{v}_1)}{q_1(\mathbf{v}_1)}\right]$$

where $\pi_1(\mathbf{v}_1) = \int \pi(\mathbf{v}_1, \mathbf{v}_2)d\mathbf{v}_2$ and $q_1(\mathbf{v}_1) = \int q(\mathbf{v}_1, \mathbf{v}_2)d\mathbf{v}_2$ are marginal densities.

A special case is to define $\mathbf{v}_1 = \mathbf{x}_{1:t-1}$ and $\mathbf{v}_2 = \mathbf{x}_t$ resulting in that

$$\mathsf{Var}[w_t(\mathbf{x}_{1:t})] \geq \mathsf{Var}[w_{t-1}(\mathbf{x}_{1:t-1})]$$

for all $t \geq 2$. In practice, as t increases, only a few particles (in the limit, only one) dominate the entire sample implying an extremely poor representation of the target distribution.

In order to attenuate the degeneracy problem, [20] suggested to add resampling steps, commonly named *sequential importance resampling* (SIR). Resampling is not needed at all time steps, only when samples are too degenerate. The need for resampling may be evaluated by computing the quality of samples using the effective sample size ([24]; [29]; [4]).

Algorithm 1 provides a generic version of the ordinary SMC for filtering, assuming the static parameters are known.

2.3. SMC with MCMC moves

Although the SIR algorithm is able to control the variance of weights and is easy to implement, successive use of the resampling stage tends to impoverish the sample by reducing the distinct number of particles. In particular, when the target distribution differs significantly each time, the variance of the Monte Carlo estimator may be affected negatively ([19]; [2]). Consequently, the efficiency of the SMC algorithms may be poor, even when clever proposals are considered.

[19] proposed to add MCMC moves after the resampling step to reduce the sample impoverishment. The approach is called the resample-move (RM) algorithm, and the main idea is to create a greater diversity in the sample by rejuvenating the particles via a combination of sequential importance resampling and MCMC sampling steps. In this approach, a Markov kernel $K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t})$ with $p(\mathbf{x}_{1:t})$ as the stationary distribution is designed to draw samples after the resample steps.

Different Markovian kernels can be used at each time-step. Some features of the kernel K are explored in [19]. In particular, the Markov kernel does not need to be ergodic since the particles target the correct distribution before the move step. For the same reason, no burn-in is required, and the number of Markov chain iterations for each particle are taken as an option. In case many Markov chain updates are required, the use of parallel programming can compensate for the extra computational complexity due to MCMC moves. A simple version of the RM algorithm at time t is as follows:

(a) Run the particle filter with resampling to obtain N new equally weighted particles;

(b) Move $\mathbf{x}^\star_{1:t}$ according to MCMC kernel K invariant with respect to $p(\mathbf{x}_{1:t})$;

(c) Approximate $E_p[g(\mathbf{x}_{1:t})|\mathbf{y}_{1:t}]$ by $N^{-1}\sum_i^N g(\mathbf{x}^{\star i}_{1:t})$.

In any version of particle filter algorithms, the RM approach can be easily implemented with a computational cost of $\mathcal{O}(tNM)$ at each time step, where M is the number of MCMC steps. A possibility is to choose s such that at each time step we only move $\mathbf{x}_{t-s+1:t}$ in which case the computational cost reduces to $\mathcal{O}(sNM)$ [see, for instance 36]. Further, sophisticated MCMC sampling algorithms [38, 25] can be applied for increasing the statistical performance.

3. Reweighting schemes

The current section introduces our approach to reweight the particles *after* a move. In the suggested methodology, we perform the move before (or without) a resample stage, followed by a *weight updating.* Many types of updates can give properly weighted samples, and by clever choices of such updates we can obtain less variability of the particle weights. This can subsequently lead to delay of resampling and better performance of the full algorithm.

3.1. Move by MCMC kernels

Assume $\mathbf{x}_{1:t} \sim q(\mathbf{x}_{1:t})$ is followed by a move $\mathbf{x}^\star_{1:t} \sim K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t})$ where K is invariant wrt $p(\mathbf{x}_{1:t})$. As suggested by [10] and [42] define an extended target distribution as $\overline{p}(\mathbf{x}^\star_{1:t}, \mathbf{x}_{1:t}) \equiv p(\mathbf{x}^\star_{1:t})h(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t})$, where $h(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t})$ is an artificial density/backward kernel that integrates to one in $\mathcal{X}_t$. Following ordinary importance sample theory working on the enlarged space, we obtain the following result:

Proposition 2. *Let $\{(\mathbf{x}_{1:t}^i, w_t^i), i = 1, \ldots, N\}$ be a properly weighted sample with respect to $p(\mathbf{x}_{1:t})$ where the particles are generated from $q(\mathbf{x}_{1:t})$. Assume for each i we make a move $\mathbf{x}_{1:t}^i \to \mathbf{x}_{1:t}^{\star i}$ by a transition kernel K that is invariant wrt $p(\mathbf{x}_{1:t})$ and update the weights by*

$$w_t^{\star i} \equiv w_t^\star(\mathbf{x}_{1:t}^i; \mathbf{x}_{1:t}^{\star i}) = w_t^i \times r_t^i \tag{2}$$

where

$$r_t^i \equiv r_t(\mathbf{x}_{1:t}^i; \mathbf{x}_{1:t}^{\star i}) = \frac{p(\mathbf{x}_{1:t}^{\star i}) h(\mathbf{x}_{1:t}^i | \mathbf{x}_{1:t}^{\star i})}{p(\mathbf{x}_{1:t}^i) K(\mathbf{x}_{1:t}^{\star i} | \mathbf{x}_{1:t}^i)},$$

and $h(\mathbf{x}_{1:t}|\mathbf{x}_{1:t}^\star)$ is a density such that $\{(\mathbf{x}_{1:t}^\star, \mathbf{x}_{1:t}) : p(\mathbf{x}_{1:t}^\star) h_t(\mathbf{x}_{1:t}|\mathbf{x}_{1:t}^\star) > 0\}$ is a subset of $\{(\mathbf{x}_{1:t}^\star, \mathbf{x}_{1:t}) : q(\mathbf{x}_{1:t}^\star) K(\mathbf{x}_{1:t}|\mathbf{x}_{1:t}^\star) > 0\}$. Then

$$\{(\mathbf{x}_{1:t}^{\star i}, w_t^{\star i}), i = 1, \ldots, N\} \text{ is proper with respect to } p(\mathbf{x}_{1:t}).$$

Also, define $q^\star(\mathbf{x}_{1:t}^\star) = \int_{\mathcal{X}_t} q(\mathbf{x}_{1:t}) K(\mathbf{x}_{1:t}^\star|\mathbf{x}_{1:t}) d\mathbf{x}_{1:t}$. Then

$$E[w_t^\star(\mathbf{x}_{1:t}; \mathbf{x}_{1:t}^\star)|\mathbf{x}_{1:t}^\star] = \frac{p(\mathbf{x}_{1:t}^\star)}{q^\star(\mathbf{x}_{1:t}^\star)} \equiv w_t^{\mathsf{opt}}(\mathbf{x}_{1:t}^\star),$$

and

$$\mathsf{Var}[w_t^\star(\mathbf{x}_{1:t}; \mathbf{x}_{1:t}^\star)] \geqslant \mathsf{Var}[w_t^{\mathsf{opt}}(\mathbf{x}_{1:t}^\star)].$$

For a proof, see appendix Appendix A.

Proposition 2 gives us some useful elements. First, by adding the MCMC move and updating the particle weights, we have that, for any density h, $w_t^\star$ is a proper weight for $\mathbf{x}_{1:t}^\star$ with respect to $p(\mathbf{x}_{1:t})$. Even though we are working with an extended space, the MCMC move does not modify the unbiased property. The main point of this sampling scheme is to highlight the multiple choices of proper weight functions $w_t^\star(\mathbf{x}_{1:t}; \mathbf{x}_{1:t}^\star)$ for any given kernel K in that there is a great flexibility in the choice of h. In subsection 3.2, we describe some of these possibilities. Note that proposition 1 can give the impression that by extending set of variables from $\mathbf{x}_{1:t}$ to $(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}^\star)$ the variance of the weights will increase. However, within this framework $\mathbf{x}_{1:t}$ is no longer the variable of interest and only service as an auxiliary variable for generation of $\mathbf{x}_{1:t}^\star$. An important point is then that the marginal target distribution for $\mathbf{x}_{1:t}$ can deviate from $p(\mathbf{x}_{1:t})$ and thereby Proposition 1 is no longer relevant.

Second, in case we wish to compute the conditional expectation of $g(\mathbf{x}_{1:t})$, this quantity can be consistently estimated by

$$\sum_{i=1}^N w_t^{\star i} g(\mathbf{x}_{1:t}^{\star i}) / \sum_{i=1}^N w_t^{\star i}. \tag{3}$$

Finally, despite in most of the cases we are not able to evaluate $q^\star$, the updated particle weights are unbiased *estimates* of the optimal weights w_t^{opt}.

The stochastic term r_t can be seen as a dynamic correction term that will tend to rebalance the ordinary particle weights after an MCMC move. By updating the ordinary particle weights, we

Algorithm 2 Move-reweighting for Filtering with MCMC kernels

```
At time $t$ do
    Propagations as in the ordinary SMC
    Move to $\mathbf{x}_t^{\star i}$ via a $p$-invariant MCMC kernel, $K(\mathbf{x}_t^{\star i}|\mathbf{x}_t^i)$.
    Update and normalize the weights $w_t^{\star i} \propto w_t^i \times r_t^i$ where
        $r_t^i = \frac{p(\mathbf{x}_{1:t}^{\star i}) h(\mathbf{x}_{1:t}^i|\mathbf{x}_{1:t}^{\star i})}{p(\mathbf{x}_{1:t}^i) K(\mathbf{x}_{1:t}^{\star i}|\mathbf{x}_{1:t}^i)}$,
if Effective sample size is small then
    Resample to obtain $N$ new equally-weighted particles.
```

introduce an additional chance to reduce the cases where the distribution of the weights is extremely skewed. Since via an MCMC move, $\mathbf{x}_{1:t}^\star$ is closer to the target distribution than $\mathbf{x}_{1:t}$, the updated weights should also be closer to $1/N$. For clever choices of h we can obtain that $\mathsf{Var}[w_t^\star] < \mathsf{Var}[w_t]$ allowing us to reduce the weight degeneracy.

The move-reweighing scheme is a simple combination of SMC and MCMC sampling where computational complexity is comparable to the RM algorithm. Similar to the RM algorithm, parallel sampling may be applied to increase the computational performance. Since $\mathbf{x}_{1:t}$ are auxiliary particles, we only need to store $\mathbf{x}_{1:t}^{\star 1:N}$ and their respective proper weights to provide consistent estimation. As for the RM algorithm, any MCMC schemes can be applied. For example, see the appendix in [42] for details on how to update the weights when proposals are generated via Metropolis-Hastings moves. Note that, after updating the particle weights the degeneracy problem may still be present, and resample stages may be required in most practical situations. By clever choices of move steps and weight updating schemes, the degeneracy problem can however be reduces, delaying the need for resampling.

Algorithm 2 describes our algorithm in this case. Similar to the RM algorithm, moves will typically only involve the last s components of $\mathbf{x}_{1:t}$ in which case the weight updating reduces to

$$r_t^i = \frac{p(\mathbf{x}_{t-s+1:t}^{\star i}|\mathbf{x}_{1:t-s}^i) h(\mathbf{x}_{t-s+1:t}^i|\mathbf{x}_{1:t-s}^i, \mathbf{x}_{t-s+1:t}^{\star i})}{p(\mathbf{x}_{t-s+1:t}^i|\mathbf{x}_{1:t-s}^i) K(\mathbf{x}_{t-s+1:t}^{\star i}|\mathbf{x}_{1:t}^i)}$$

in which case the computational complexity at each iteration reduces to $\mathcal{O}(sNM)$.

3.2. Choices of the h functions

The choices of the conditional densities h should, in general, be balanced in two strategies: (i) to control the variability of the particle weights; (ii) to find an artificial density that will lead to weights that are easy to evaluate. Additionally, by the importance sampling principles, h must have a support that is small enough.

Some special cases of h and their respective particle weights are interesting to be discussed. To explore their performance, we will see what kind of weights we obtain in three special cases:

c.1 If $q(\mathbf{x}_{1:t}) = p(\mathbf{x}_{1:t})$, we are drawing from the right distribution in the first place. Updating the particles with an p invariant kernel K also gives the right distribution on the updated/moved particle. We would therefore like the weights to be equal to 1 in this case.

c.2 If $K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t}) = p(\mathbf{x}^\star_{1:t})$, the move gets us directly to the right distribution, so also in this case, we would like the weights to be equal to 1.

c.3 If $K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t}) = q^\star(\mathbf{x}^\star_{1:t})$, that is $\mathbf{x}_{1:t}$ is not used in the update, we would like the weights to be equal to $p(\mathbf{x}^\star_{1:t})/q^\star(\mathbf{x}^\star_{1:t})$.

By Proposition 2, the optimal weight we can obtain at time t is $w_t^{\mathsf{opt}}(\mathbf{x}^\star_{1:t};\mathbf{x}_{1:t})$. This corresponds to choosing $h^{\mathsf{opt}}(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t}) = q(\mathbf{x}_{1:t})K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t})/q^\star(\mathbf{x}^\star_{1:t})$ which is a legal density. In practice, however, the quantities involved will not be possible to calculate, making this not a realistic choice.

Another option is

$$h^1(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t}) \equiv \frac{p(\mathbf{x}_{1:t})K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t})}{p(\mathbf{x}^\star_{1:t})} \Rightarrow w_t^\star(h^1) = w_t. \tag{4}$$

This shows that when MCMC moves are applied, using the original weights still gives a properly weighted sample. An important special case is when resampling is applied before move, so that $w_t^i = 1/N$ for all i. In this case we obtain the RM algorithm. This Monte Carlo scheme was originally proposed, outside the context of particle methods, by [31] and it is also referred as generalized importance sampling in [38]. Note however that although easy to apply, these weights will typically be different from the optimal ones, indicating that better choices are possible. In the *c*.1 case, $w_t^\star(h^1) = w_t = 1$, giving a sensible result. In case *c*.2, however, $w_t^\star(h^1) = w_t = p(\mathbf{x}_{1:t})/q(\mathbf{x}_{1:t})$, not at all taking into account that we in this case are using a good (perfect) kernel for the move. A similar situation occur for case *c*.3.

We can also define the h function by the Markovian transition kernel:

$$h^2(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t}) \equiv K(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t}) \Rightarrow w_t^\star(h^2) = \frac{p(\mathbf{x}^\star_{1:t})K(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t})}{q(\mathbf{x}_{1:t})K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t})}. \tag{5}$$

Note that if K satisfies detailed balance, this choice coincides with h^1 and $w_t^\star(h^2) = w_t$ and the properties are similar to that choice.

Yet another possibility is to choose h as the marginal proposal distribution

$$h^3(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t}) \equiv q(\mathbf{x}_{1:t}) \Rightarrow w_t^\star(h^3) = \frac{p(\mathbf{x}^\star_{1:t})}{K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t})}. \tag{6}$$

Both in cases *c*.2 and *c*.3, we obtain the ideal weights, but in case *c*.1 we obtain $w_t^\star(h^3) = \frac{p(\mathbf{x}^\star_{1:t})}{K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t})}$, not taking into account that we started right in this case. This therefore will be a good choice if we have an efficient MCMC kernel, i.e. $K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t}) \approx p(\mathbf{x}^\star_{1:t})$ or a move not depending much on $\mathbf{x}_{1:t}$. These weights will also be easy to compute, assuming the MCMC kernels are easily available.

Finally, any mixture of the previous h's can also be considered. For example, for $\alpha \in [0,1]$ we have

$$\begin{aligned} &h(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t}) \equiv \alpha q(\mathbf{x}_{1:t}) + (1-\alpha)K(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t}) \\ &\Rightarrow w_t^\star(h) = \alpha w_t^\star(h^3) + (1-\alpha)w_t^\star(h^2). \end{aligned} \tag{7}$$

In practice, only parts of $\mathbf{x}_{1:t}$ is moved will typically be moved. Some extra care is needed in such cases. Appendix

Algorithm 3 Move-reweighting for Filtering with arbitrary transitions

```
At time $t$ do
    Propagations as in the ordinary SMC
    Move to $\mathbf{x}_t^{\star i}$ via a transition kernel $Q(\mathbf{x}_t^{\star i}|\mathbf{x}_t^i)$.
    Update and normalize the weights $w_t^{\star i} \propto w_t^i \times r_t^i$ where
        $r_t^i = \frac{p(\mathbf{x}_{1:t}^{\star i})h(\mathbf{x}_{1:t}^i|\mathbf{x}_{1:t}^{\star i})}{p(\mathbf{x}_{1:t}^i)Q(\mathbf{x}_{1:t}^{\star i}|\mathbf{x}_{1:t}^i)}$,
if Effective sample size is small then
    Resample to obtain $N$ new equally-weighted particles.
```

3.3. General moves

In cases where the use of an MCMC kernel is difficult, alternative moves can be considered. For these situations, we first sample particles from a proposal $\mathbf{x}_{1:t} \sim q(\cdot)$, followed by an arbitrary transition kernel, i.e. $\mathbf{x}_{1:t}^\star \sim Q(\cdot|\mathbf{x}_{1:t})$. The following statement is analogous to that of Proposition 2:

Proposition 3. *Let $\{(\mathbf{x}_{1:t}^i, w_t^i), i = 1, \ldots, N\}$ be a properly weighted sample with respect to p, and the particles are generated from q. Assume for each i we move $\mathbf{x}_{1:t}^i \to \mathbf{x}_{1:t}^{\star i}$ by a transition kernel $Q(\cdot|\mathbf{x}_{1:t})$ and update the weights by*

$$w_t^{\star i} \equiv w_t^{\star i}(\mathbf{x}_{1:t}; \mathbf{x}_{1:t}^\star) = w_t^i \times r_t^i \tag{8}$$

where

$$r_t^i \equiv r_t(\mathbf{x}_{1:t}; \mathbf{x}_{1:t}^\star) = \frac{p(\mathbf{x}_{1:t}^{\star i})h(\mathbf{x}_{1:t}^i|\mathbf{x}_{1:t}^{\star i})}{p(\mathbf{x}_{1:t}^i)Q(\mathbf{x}_{1:t}^{\star i}|\mathbf{x}_{1:t}^i)}$$

and h is a density such that $\{(\mathbf{x}_{1:t}^\star, \mathbf{x}_{1:t}) : p(\mathbf{x}_{1:t}^\star)h(\mathbf{x}_{1:t}|\mathbf{x}_{1:t}^\star) > 0\}$ is a subset of $\{(\mathbf{x}_{1:t}^\star, \mathbf{x}_{1:t}) : q(\mathbf{x}_{1:t})Q(\mathbf{x}_{1:t}^\star|\mathbf{x}_{1:t}) > 0\}$. Then

$$\{(\mathbf{x}_{1:t}^{\star i}, w_t^{\star i}(\mathbf{x}_{1:t}; \mathbf{x}_{1:t}^{\star i})), i = 1, \ldots, N\} \text{ is proper with respect to } p.$$

The proof of Proposition 3 is identical to the proof of Proposition 2, only replacing $K(\mathbf{x}_{1:t}^\star|\mathbf{x}_{1:t})$ with $Q(\mathbf{x}_{1:t}^\star|\mathbf{x}_{1:t})$.

When an MCMC kernel is not available, the use of some arbitrary transition kernel q to move and reweight the particle can be attractive. The choice of Q can, for example, be an approximation to an MCMC kernel or a clever proposals taking into account that we have the samples generated from q as auxiliary particles. For instance, we can choose Q as the Langevin proposals, such as $Q(\mathbf{x}_{1:t}^\star|.) = \mathcal{N}_p(\mathbf{x}_{1:t}^\star|\mathbf{x}_{1:t} + \frac{\sigma^2}{2}\nabla p, \sigma^2 I_p)$ where σ^2 is some tuning parameter, and ∇p is the gradient of the unnormalized target distribution.

Algorithm 3 offers an alternative (analogues to algorithm 2) to implement the move-reweighting with arbitrary transitions algorithm for filtering. As for algorithm 2, more general resampling schemes can be considered.

3.4. Summary of particle moves and (re)-weighting strategies

If we have an p-invariant MCMC kernel, K, or some arbitrary transition kernel Q to diversify the particles, we can adduce at least four strategies for (re)-weighting them in an SMC framework:

- $s.1$ Start with an equally weighted sample, move the particles via K and maintain the equal particle weights;
- $s.2$ Start with a properly weighted sample, move the particles via K and maintain the same particle weights;
- $s.3$ Start with a properly weighted sample, move the particles via K and update the particle weights; or
- $s.4$ Start with a properly weighted sample, move the particles via Q and update the particle weights.

Strategy $s.1$ is the RM algorithm mentioned in subsection 2.3, and $s.2$ is its generalization, both provided properly weighted samples with respect to p. Strategies $s.3$ and $s.4$ are cases where, through a diversification step, the weights are updated. Thus, these schemes are promising to make the particle weights less variable and, at the same time, reduce the sample impoverishment.

4. Numerical Illustrations

In this section we illustrate the move-reweighing approach with two examples. The first is a state-space model with a linear Gaussian state process combined with a non-Gaussian likelihood. Here we are able to move (parts of) the particles directly from a Gibbs kernel. The second is a non-linear Gaussian stochastic system, where the rejuvenated particles are reweighted after a move with a general kernel. In both cases, we assume all static parameters to be known, and we consider $t = 1, 2, \ldots, 200$.

As a standard measure for evaluating the sample impoverishment, we calculate the effective sample size [ESS, 27] by

$$\text{ESS}_t = \frac{(\sum_{i=1}^{N} w_t^i)^2}{\sum_{i=1}^{N} (w_t^i)^2}. \tag{9}$$

The performance is also evaluated using the ordinary root mean square error (RMSE). Assuming μ_t is some functional measure of $\pi(\mathbf{x}_t|\mathbf{y}_{1:t})$ and $\hat{\mu}_t$ is the corresponding particle filter estimate, it is defined through

$$\text{RMSE}(\mu_t) = \sqrt{T^{-1} \sum_{t=1}^{T} (\hat{\mu}_t - \mu_t)^2} \tag{10}$$

The functionals that will be considered are posterior means or α-quantiles of the marginal distribution $\pi(x_{t,l}|\mathbf{y}_{1:t})$. In practice μ_t will be unknown but are approximated by running the ordinary SMC algorithm using a huge number of particles ($N = 35 \times 10^4$).

Table 1: Gauss-Poisson model without resampling. Comparison of the ordinary SMC algorithm (**O**, algorithm 1), the move-without-reweighting algorithm (**M**, algorithm 2 using the backward kernel from equation (4)) and the move-reweight algorithm (**MR**, algorithm 2 using the backward kernel from equation (6)), all without resampling. Results are based on one simulation of data based on model (11) with parameters given in (12) and an average of 1000 repetitions of the algorithms. $N = 5000$ particles were used in all cases. Both for ESS and RMSE, an average over time is given, that is $\overline{\text{ESS}} = \frac{1}{T}\sum_{t=1}^{T} \text{ESS}_t$ and $\overline{\text{RMSE}}_l = \frac{1}{T}\sum_{t=1}^{T} \text{RMSE}(\mu_{l,t})$ with $\mu_{l,t} = E[x_{l,t}|\mathbf{y}_{1:t}]$.

Alg	$\overline{\text{ESS}}$(%)	Mean		10% quantile		90% quantile	
		$\overline{\text{RMSE}}_1$	$\overline{\text{RMSE}}_2$	$\overline{\text{RMSE}}_1$	$\overline{\text{RMSE}}_2$	$\overline{\text{RMSE}}_1$	$\overline{\text{RMSE}}_2$
O	0.0284	1.4878	0.1350	3.5655	2.9066	0.2436	0.2157
M	0.0385	0.3682	0.1897	0.3609	0.4270	0.1470	0.0979
MR	4.9200	0.0631	0.0275	0.0642	0.0493	0.0783	0.0557

4.1. Gaussian process with Poisson distributed data

We start illustrating our methodology for a Poisson-Gaussian state space model. Hence, assume $\mathbf{x}_t = (x_{1,t}, x_{2,t})'$ and

$$\begin{aligned} x_{1,t} =& 0.90x_{1,t-1} + \varepsilon_{1,t} \\ x_{2,t} =& 0.20x_{2,t-1} + 0.95x_{1,t} + \varepsilon_{2,t} \\ y_t \sim& \text{Poisson}(\exp(5 + x_{2,t})). \end{aligned} \tag{11}$$

For $\mathbf{x}_1 = (x_{1,1}, x_{2,1})$, we assume to follow the stationary distribution of the process. In our experiments we set

$$\begin{pmatrix}\varepsilon_{1,t}\\ \varepsilon_{1,t}\end{pmatrix} \overset{ind}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \quad \mathbf{\Sigma} = \begin{pmatrix}1 & 0\\ 0 & 0.1\end{pmatrix} \tag{12}$$

Note in particular that observation y_t only depend on $x_{2,t}$. Based on this model, one set of observations $\mathbf{y}_{1:200}$ where generated which will be our data in the following experiments.

Propagations are in this case applied by generating $x_{1,t} \sim \pi(x_{1,t}|\mathbf{x}_{t-1})$ followed by simulating $x_{2,t} \sim q_1(x_{2,t}|x_{1,t}, y_t, \mathbf{x}_{t-1})$ where the latter is the Laplace approximation to $\pi(x_{2,t}|x_{1,t}, y_t, \mathbf{x}_{t-1})$. The propagation is then followed by a move $(x_{2,t}, x_{1,t}) \to (x^\star_{2,t}, x^\star_{1,t})$ where $x^\star_{2,t} = x_{2,t}$ while $x^\star_{1,t} \sim \pi(x_{1,t}|\mathbf{x}_{t-1}, x_{2,t}, y_t) = \pi(x_{1,t}|\mathbf{x}_{t-1}, x_{2,t})$. Hence, we move $x^\star_{1,t}$ directly from the Gibbs kernel which is a Gaussian distribution. In addition, for reweighting the particles we compute $w^\star_t$ using equation (B.2).

A first experiment was carried out with $N = 5000$ *without resampling* in order to explore the degeneracy problem. Table 1 shows the performance measures while Figure 2 displays the behaviour of the effective sample over time. From Table 1 we see that while some small improvements are made when including a move, a much larger improvement is made when such a move is combined with reweighting. Figure 2 shows that the main improvements are made in the first time-points, indicating that degeneracy can be reduced but that resampling will be needed at a later time point.

In a second experiment, the influence of resampling was explored. Using the same simulated data set, we ran the move-reweighing (MR) approach adding multinomial resampling to compare the resample-move (RM) algorithm with the move-reweight alternatives. Table 2 and Figure 3 report the main results using these methods with $N = 50$ and $N = 5000$ particles. First, since the MR approach introduces a mechanism to update the particle weights before the resample stage, the effective sample size can gain a significant improvement as demonstrated in the first column of

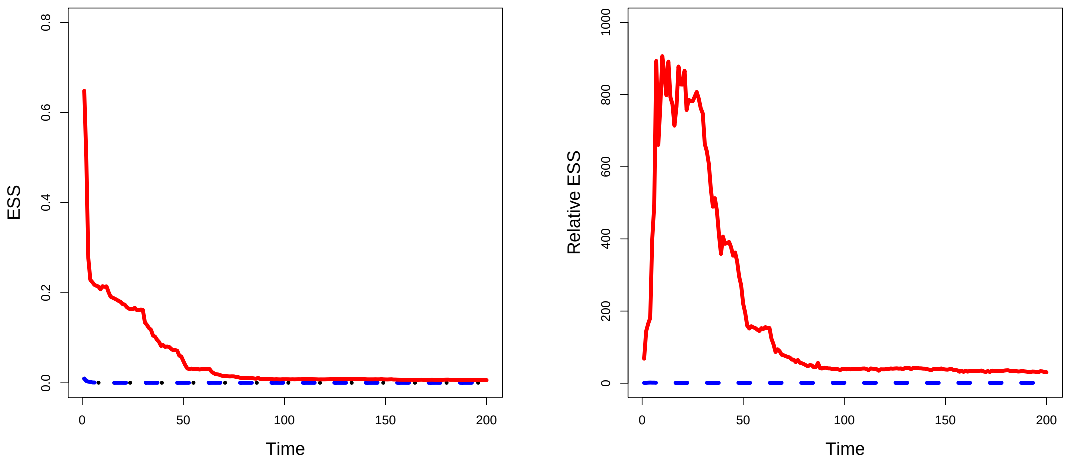

Figure 2: Effective sample size for algorithms described in Table 1. Left panel, ESS, and right panel is the relative ESS scaled by the ordinary ESS. Solid red line: Move-reweighting, dashed blue line: resample-move, dotted black line: ordinary SMC, the last one indistinguishable from the dash line in the left plot and not present in the right plot.

Table 2.In this case, ESS measures the one-step ahead loss in effective sample size after a resampling has been performed (no number is given for MR2 since resampling in this case is stochastic). In practice, this shows that with MR-approaches more particles can be representative to approximate the target distribution. This is also seen in the RMSE number for different targets, in particular for $N = 50$. Closer inspection of the results also revealed that the main benefits were at time points where the data was most informative (high y_t values). For a large number of particles ($N = 5000$), the approaches presented similar results. This mainly indicate that in this case any methods is doing good and that the gain mainly is when fewer particles are used. In summary, good performance depends on a combination of moving the particles to the region of strong support and incorporating this information in the weights.

4.2. A Low-dimensional non-linear dynamical system

Consider now a low-dimensional linear dynamical model

$$\begin{aligned} x_{1,t} =& x_{1,t-1} + hx_{2,t-1} + \sigma_1\varepsilon_{1,t} && (13)\\ x_{2,t} =& x_{2,t-1} + h\alpha(1 - x^2_{1,t-1})x_{2,t-1} - hx_{1,t-1} + \sigma_2\varepsilon_{2,t} && (14)\\ y_t =& x_{2,t} + \sigma_3\varepsilon_{3,t} && (15) \end{aligned}$$

where $\{\varepsilon_{l,t}\}$ are independent white noise processes with standard normal marginals. This model was used in Gao and Zhang [18] as a discretized version of the Van del Pol oscillator described through a second-order differential equation (with h as the discretization step size). We deviate somewhat from Gao and Zhang [18] in the observation model. The static parameter vector is $\theta = (h, \alpha, \sigma_1^2, \sigma_2^2, \sigma_3^2)'$ in which for our simulations we use $h = 0.1, \alpha = 1, \sigma_1^2 = 5\times10^{-2}, \sigma_2^2 = 8\times10^{-3}$ and $\sigma_3^2 = 1 \times 10^{-3}$. We start sampling the states $x_{1,1}$ and $x_{2,1}$ independently from $N(0, 0.01)$.

Table 2: Gauss-Poisson model with resampling. Comparison of the ordinary SMC algorithm with resampling at each time-point (**O**), resample-move (**RM**), move-reweighting with resampling at each time-point (**MR1**) and move-reweighting with resampling if $ESS/N < 50\%$ (**MR2**). Results are based on one simulation of data based on model (11) with parameters given in (12) and an average of 1000 repetitions of the algorithms.

N	Alg	$\overline{\text{ESS}}(\%)$	Mean		10% quantile		90% quantile	
			$\overline{\text{RMSE}}_1$	$\overline{\text{RMSE}}_2$	$\overline{\text{RMSE}}_1$	$\overline{\text{RMSE}}_2$	$\overline{\text{RMSE}}_1$	$\overline{\text{RMSE}}_2$
50	**O**	27.84	0.1451	0.0569	0.3120	0.1569	0.3378	0.0339
	RM	27.86	0.0931	0.0552	0.2620	0.1467	0.3290	0.0269
	MR1	95.64	0.0599	0.0403	0.0612	0.0705	0.0359	0.0169
	MR2	-	0.0548	0.0394	0.0612	0.0684	0.0466	0.0406
5000	**O**	27.13	0.0203	0.0404	0.1924	0.0548	0.2737	0.0163
	RM	27.30	0.0293	0.0165	0.0545	0.0373	0.0281	0.0150
	MR1	94.61	0.0453	0.0320	0.0513	0.0672	0.0353	0.0108
	MR2	-	0.0452	0.0321	0.0638	0.0308	0.0355	0.0513

In this case it is easiest to consider $\pi(\mathbf{x}_{1,t-1}, x_{2,t}|\mathbf{y}_{1:t})$ as the target distribution at time t. Propagation at time t is then performed by simulating $x_{1,t-1} \sim \pi(x_{1,t-1}|\mathbf{x}_{t-2}, x_{2,t-1})$ and $x_{2,t} \sim \pi(x_{2,t}|\mathbf{x}_{t-1}, y_t)$ (two Gaussian distributions). This is followed by a move $x^{\star}_{1,t-1} \sim q(x^{\star}_{1,t-1}|\mathbf{x}_{t-2}, x_{2,t})$, the Laplace approximation of $\pi(x_{1,t-1}|\mathbf{x}_{t-2}, x_{2,t-1}, x_{2,t}, y_t) = \pi(x_{1,t-1}|\mathbf{x}_{t-2}, x_{2,t})$. Note that in this case our move kernel is not invariant with respect to the target distribution, making weight updating according to Proposition 3 necessary to use. In particular, we will assume

$$h(\mathbf{x}_{1,t_1}|\mathbf{x}_{t-2}, x^{*}_{1,t-1}, x_{2,t-1}, x_{2,t}) = \pi(x_{1,t-1}|\mathbf{x}_{t-2})$$

in which case

$$w^{\star}_t = w_{t-1}\frac{\pi(x^{*}_{1,t-1}|\mathbf{x}_{t-2})\pi(x_{2,t}|x^{*}_{1,t-1}, x_{2,t-1})\pi(y_t|x_{2,t})}{\pi(x_{2,t}|\mathbf{x}_{t-1}, y_t)q(x^{*}_{1,t-1}|\mathbf{x}_{t-2}, x_{1,t-1}, x_{2,t-1}, x_{2,t})}$$

In this case, the resample-move approach is not possible to use, so only a comparison between ordinary SMC and move-reweighting is performed in this case.

Even the particles are not rejuvenated via an MCMC kernel, the empirical results suggested again that by reweighing the particles we can gain some improvement in the SMC performance. In this case, the main improvement is in the $x_{1,t}$ variable which is the one that is moved. Moving by an arbitrary kernel can also be propitious to avoid the cases where few extreme weights dominate the entire sample resulting a high variance or lack of accuracy measured by the RMSE. Note that in this case, the (one step ahead) effective sample size did not differ much for the two approaches, indicating that the main benefit in this case is the possibility of better samples through the move step.

5. Discussion

The resample-move (RM) algorithm is based on that if a particle is from the right distribution, a move using a kernel invariant to the target distribution will also be from the right distribution. Getting particles from the right distribution is obtained by a preliminary resample step in the RM

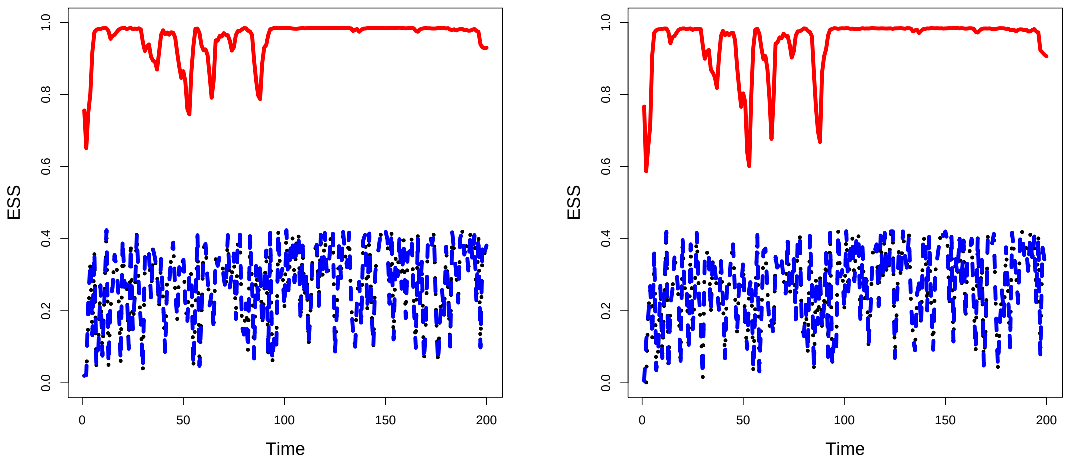

Figure 3: Effective sample for the model and algorithms describe in Table 2. Left panel, ESS obtained by SMC with $N = 50$, and right panel with $N = 5000$. Solid red line: Move-reweighting, dashed blue line: Resample-move and dotted black line: Ordinary SMC.

approach. If particles has corresponding importance weights, keeping these weights after such a move will still give properly weighted particles.

In this paper we have extended the resample-move in different ways. By considering moves as part of simulations in an extended space, alternative weights that still are proper with respect to the target distribution is possible to obtain. This have several benefits:

- Better weights can be obtained, taking into account that when moves are applied another importance distribution is used.
- Resampling is not necessary at each time step, making it possible to combine moves with different resampling strategies.

Table 3: Oscillator model with resampling at each time-point. RMSE for quantiles of the ordinary SMC algorithm (**O**) and move-reweighting (**MR**). Results are based on one simulation of data based on model described in equations (14-15) with an average of 1000 repetitions of the algorithms.

N	Alg	$\overline{\text{ESS}}$(%)	Mean		10% quantile		90% quantile	
			RMSE_1	RMSE_2	RMSE_1	RMSE_2	RMSE_1	RMSE_2
50	**O**	59.91	0.1336	0.0069	0.2174	0.0159	0.1954	0.0057
	MR	61.64	0.1167	0.0067	0.1469	0.0160	0.0807	0.0061
500	**O**	58.73	0.0461	0.0022	0.2012	0.0160	0.1797	0.0051
	MR	59.38	0.0416	0.0025	0.1352	0.0159	0.0752	0.0049
5000	**O**	58.79	0.0171	0.0007	0.2006	0.0161	0.1786	0.0052
	MR	58.88	0.0147	0.0012	0.1303	0.0159	0.0747	0.0048

- Moves using kernels that are not necessary invariant with respect to the target distribution is possible to apply.

Our approach, the move-reweighting algorithm, allows a great flexibility to reweight the samples using a proper weight with respect to the right distribution. We have pointed out that the traditional framework to add MCMC moves within particle filter algorithms ([19]) can be encapsulated as a special case in our approach.

In our simulation studies, we demonstrate that the move-reweighting algorithm can successfully reduce weight degeneracy and, at the same time, increase sample diversity. In the first experiment, we showed clearly how effective sample size has been affected when we update the weights after a MCMC move. In short, the move-reweighting approach can be promising to attenuate the asymmetry in the distribution of the particle weights. As a result, we obtain a satisfactory increase in the effective sample size. In the second experiment we illustrate the use of moves with arbitrary kernels and show that improvements can be obtained compared to ordinary SMC (the resample-move algorithm is not possible to use in this case).

Although we have assumed that static parameters are known, an interesting further research topic is to see how our approach can be combined with different alternatives for sampling from the joint distribution $p(\mathbf{x}_{1:t}, \theta)$ [22, 1, 8]. Furthermore, advanced MCMC move approaches (e.g. adaptive methods by [17]) can also be possible to merge with the reweighting schemes presented in order to construct more efficient algorithms. Lastly, in an attempt to address the problems caused by the curse of dimensionality in SMC methods [40], the move-reweighting strategies can possibly be used for improvements of particle filter algorithms in high-dimension systems.

Acknowledgments

We gratefully acknowledge financial support from CAPES-Brazil and Statistics for Innovation Center, in Norway.

Appendix A. Proofs

Proof of Proposition 2. As g is a p-integrable function,

$$\begin{aligned}
E[g(\mathbf{x}^\star_{1:t})w^\star_t(\mathbf{x}^\star_{1:t})] &= \int_{\mathcal{X}_t\times\mathcal{X}_t} g(\mathbf{x}^\star_{1:t})w^\star_t(\mathbf{x}^\star_{1:t})q(\mathbf{x}_{1:t})K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t})d\mathbf{x}^\star_{1:t}d\mathbf{x}_{1:t}\\
&= \int_{\mathcal{X}_t\times\mathcal{X}_t} g(\mathbf{x}^\star_{1:t})p(\mathbf{x}^\star_{1:t})h(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t})d\mathbf{x}^\star_{1:t}d\mathbf{x}_{1:t}\\
&= \int_{\mathcal{X}_t} g(\mathbf{x}^\star_{1:t})p(\mathbf{x}^\star_{1:t})\int_{\mathcal{X}_t} h(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t})d\mathbf{x}_{1:t}d\mathbf{x}^\star_{1:t}\\
&= \int_{\mathcal{X}_t} g(\mathbf{x}^\star_{1:t})p(\mathbf{x}^\star_{1:t})d\mathbf{x}^\star_{1:t} = E_p[g(\mathbf{x}^\star_{1:t})].
\end{aligned}$$

Further, using that the conditional sampling distribution for $\mathbf{x}_{1:t}$ given $\mathbf{x}^\star_{1:t}$ is $q(\mathbf{x}_{1:t})K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t})/q^\star(\mathbf{x}^*_{1:t})$ we have

$$\begin{aligned}
E[w^\star_t(\mathbf{x}_{1:t};\mathbf{x}^\star_{1:t})|\mathbf{x}^\star_{1:t}] =& \int_{\mathcal{X}_t} w^\star_t(\mathbf{x}_{1:t};\mathbf{x}^\star_{1:t})\frac{q(\mathbf{x}_{1:t})K(\mathbf{x}^\star_{1:t}|\mathbf{x}_{1:t})}{q^\star(\mathbf{x}^*_{1:t})}d\mathbf{x}_{1:t}\\
=&\frac{p(\mathbf{x}^\star_{1:t})}{q^\star(\mathbf{x}^\star_{1:t})}\int_{\mathcal{X}_t} h(\mathbf{x}_{1:t}|\mathbf{x}^\star_{1:t})d\mathbf{x}_{1:t}\\
=&\frac{p(\mathbf{x}^\star_{1:t})}{q^\star(\mathbf{x}^\star_{1:t})} = w^{\mathsf{opt}}_t(\mathbf{x}^\star_{1:t}).
\end{aligned}$$

Finally, $\mathsf{Var}[w^\star_t(\mathbf{x}_{1:t};\mathbf{x}^\star_{1:t})] \geq \mathsf{Var}[w^{\mathsf{opt}}_t(\mathbf{x}^\star_{1:t})]$ is a direct consequence of Proposition 1 choosing $\mathbf{v}_1 = \mathbf{x}^\star_{1:t}$ and $\mathbf{v}_2 = \mathbf{x}_{1:t}$. □

Appendix B. Moving only parts

Assume now that only parts of $\mathbf{x}_{1:t}$ is moved. Some care is then needed. Write $\mathbf{x}_{1:t} = (\mathbf{x}^m_{1:t}, \mathbf{x}^f_{1:t})$ where $\mathbf{x}^m_{1:t}$ is the part of $\mathbf{x}_{1:t}$ that is moved while $\mathbf{x}^f_{1:t}$ is fixed (after propagation). Then the proper extended distribution to work with is $p(\mathbf{x}^f_{1:t}, \mathbf{x}^{m*}_{1:t})h(\mathbf{x}^m_{1:t}|\mathbf{x}^f_{1:t}, \mathbf{x}^m_{1:t})$ and weights are given by

$$w^*_t(\mathbf{x}_{1:t}, \mathbf{x}^{m*}_{1:t}) =\frac{p(\mathbf{x}^f_{1:t}, \mathbf{x}^{m*}_{1:t})h(\mathbf{x}^m_{1:t}|\mathbf{x}^f_{1:t}, \mathbf{x}^{m*}_{1:t})}{q(\mathbf{x}_{1:t})K(\mathbf{x}^{m*}_{1:t}|\mathbf{x}_{1:t})}.$$

Writing $q(\mathbf{x}_{1:t}) = q(\mathbf{x}^f_{1:t})q(\mathbf{x}^m_{1:t}|\mathbf{x}^f_{1:t})$ and choosing $h(\mathbf{x}^m_{1:t}|\mathbf{x}^f_{1:t}, \mathbf{x}^{m*}_{1:t}) = q(\mathbf{x}^m_{1:t}|\mathbf{x}^f_{1:t})$ we end up with weights

$$w^*_t(\mathbf{x}_{1:t}, \mathbf{x}^{m*}_{1:t}) =\frac{p(\mathbf{x}^f_{1:t}, \mathbf{x}^{m*}_{1:t})}{q(\mathbf{x}^f_{1:t})K(\mathbf{x}^{m*}_{1:t}|\mathbf{x}_{1:t})}.$$

An alternative is to write $q(\mathbf{x}_{1:t}) = q(\mathbf{x}^m_{1:t})q(\mathbf{x}^f_{1:t}|\mathbf{x}^m_{1:t})$ and choosing $h(\mathbf{x}^m_{1:t}|\mathbf{x}^f_{1:t}, \mathbf{x}^{m*}_{1:t}) = q(\mathbf{x}^m_{1:t})$ in which case the weights reduces to

$$w^*_t(\mathbf{x}_{1:t}, \mathbf{x}^{m*}_{1:t}) =\frac{p(\mathbf{x}^f_{1:t}, \mathbf{x}^{m*}_{1:t})}{q(\mathbf{x}^f_{1:t}|\mathbf{x}^m_{1:t})K(\mathbf{x}^{m*}_{1:t}|\mathbf{x}_{1:t})}.$$

This alternative will typically be less efficient, but can be preferable when $q(\mathbf{x}^f_{1:t})$ is difficult to compute.

An important special case is where only (parts of) the last $\mathbf{x}_t$ is moved. In that case, we have

$$\begin{aligned}
w^*_t(\mathbf{x}_{1:t}, \mathbf{x}^{m*}_t) =&\frac{p(\mathbf{x}_{1:t-1}, \mathbf{x}^f_t, \mathbf{x}^{m*}_t)h(\mathbf{x}^m_t|\mathbf{x}_{1:t-1}, \mathbf{x}^f_t, \mathbf{x}^{m*}_t)}{q(\mathbf{x}_{1:t-1})q(\mathbf{x}_t|\mathbf{x}_{1:t-1})K(\mathbf{x}^{m*}_t|\mathbf{x}_{1:t})}\\
\propto& w_{t-1}(\mathbf{x}_{1:t-1})\frac{\pi(\mathbf{x}^f_t, \mathbf{x}^{m*}_t|\mathbf{x}_{1:t-1})\pi(\mathbf{y}_t|\mathbf{x}^f_t, \mathbf{x}^{m*}_t)h(\mathbf{x}^m_t|\mathbf{x}_{1:t-1}, \mathbf{x}^f_t, \mathbf{x}^{m*}_t)}{q(\mathbf{x}_t|\mathbf{x}_{1:t-1})K(\mathbf{x}^{m*}_t|\mathbf{x}_{1:t})}
\end{aligned}$$

where $w_{t-1}(\mathbf{x}_{1:t-1}) = p(\mathbf{x}_{1:t-1})/q(\mathbf{x}_{1:t-1})$ are the weights from the previous time points (actually only an unbiased estimate of w_{t-1} is needed so that moves and reweighting at previous time points can easily be incorporated). Similar choices of h as above can be applied in which cases proper weights are

$$w_t^*(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}^{m*}) = w_{t-1}(\mathbf{x}_{1:t-1}) \frac{\pi(\mathbf{x}_t^f, \mathbf{x}_t^{m*}|\mathbf{x}_{1:t-1})\pi(\mathbf{y}_t|\mathbf{x}_t^f, \mathbf{x}_t^{m*})}{q(\mathbf{x}_t^f|\mathbf{x}_{1:t-1})K(\mathbf{x}_t^{m*}|\mathbf{x}_{1:t})} \tag{B.1}$$

and

$$w_t^*(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}^{m*}) = w_{t-1}(\mathbf{x}_{1:t-1}) \frac{\pi(\mathbf{x}_t^f, \mathbf{x}_t^{m*}|\mathbf{x}_{1:t-1})\pi(\mathbf{y}_t|\mathbf{x}_t^f, \mathbf{x}_t^{m*})}{q(\mathbf{x}_t^f|\mathbf{x}_{1:t-1}, \mathbf{x}_t^m)K(\mathbf{x}_t^{m*}|\mathbf{x}_{1:t})}. \tag{B.2}$$

References

[1] Andrieu, C., A. Doucet, and R. Holenstein (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(3), 269–342.

[2] Berzuini, C. and W. Gilks (2003). Particle filtering methods for dynamic and static Bayesian problems. In *Models and inference in HSSS: Recent developments and perspectives*, pp. 207–227. Oxford University Press.

[3] Cappé, O., S. Godsill, and E. Moulines (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE 95*(5), 899–924.

[4] Cappé, O., E. Moulines, and T. Rydén (2005). *Inference in hidden Markov models.* Springer Verlag.

[5] Chen, R. and J. S. Liu (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 62*, 493–508.

[6] Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika 89*(3), 539–552.

[7] Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics 32*(6), 2385–2411.

[8] Chopin, N., P. E. Jacob, and O. Papaspiliopoulos (2013). SMC^2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 75*(3), 397–426.

[9] Del Moral, P. (2004). *Feynman-Kac Formulae, Genealogical and Interacting Particle Systems with Applications.* New York: Springer-Verlag.

[10] Del Moral, P., A. Doucet, and A. Jasra (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(3), 411–436.

[11] Doucet, A., M. Briers, and S. Sncal (2006). Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics 15*(3), 693–711.

[12] Doucet, A., N. de Freitas, and N. Gordon (2001). *Sequential Monte Carlo methods.* Springer-Verlag.

[13] Doucet, A., S. Godsill, and C. Andrieu (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing 10*(3), 197–208.

[14] Doucet, A. and A. M. Johansen (2009). A tutorial on particle filtering and smoothing: fifteen years later. *Handbook of Nonlinear Filtering 12*, 656–704.

[15] Fearnhead, P. (2002). Markov Chain Monte Carlo, sufficient statistics, and particle filters. *Journal of Computational and Graphical Statistics 11*(4), 848–862.

[16] Fearnhead, P. (2008). Computational methods for complex stochastic systems: a review of some alternatives to MCMC. *Statistics and Computing 18*, 151–171.

[17] Fearnhead, P. and B. Taylor (2013). An adaptive sequential Monte Carlo sampler. *Bayesian Analysis 8*(1), 1–28.

[18] Gao, M. and H. Zhang (2012). Sequential Monte Carlo methods for parameter estimation in nonlinear state-space models. *Computers & Geosciences 44*(0), 70 – 77.

[19] Gilks, W. and C. Berzuini (2001). Following a moving target Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(1), 127–146.

[20] Gordon, N., D. Salmond, and A. Smith (1993, apr). Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F 140*(2), 107 –113.

[21] Green, P., N. Hjort, and S. Richardson (2003). *Highly structured stochastic systems*, Volume 10. Oxford University Press Oxford (United Kingdom).

[22] Kantas, N., A. Doucet, S. Singh, and J. Maciejowski (2009). An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. In *Proceedings of the IFAC Symposium on System Identification (SYSID).*

[23] Kevin, M. (2012). *Machine Learning: a probabilistic perspective.* The MIT press.

[24] Kong, A., J. S. Liu, and W. H. Wong (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American statistical association 89*(425), 278–288.

[25] Liang, F., C. Liu, and R. Carroll (2011). *Advanced Markov chain Monte Carlo methods: learning from past samples*, Volume 714. Wiley.

[26] Lin, M., R. Chen, and J. S. Liu (2013). Lookahead strategies for sequential Monte Carlo. *Statistical Science 28*(1), 69–94.

[27] Liu, J. S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing 6*(2), 113–119.

[28] Liu, J. S. (2001). *Monte Carlo strategies in scientific computing.* Springer.

[29] Liu, J. S. and R. Chen (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association 90*(430), 567–576.

[30] Liu, J. S. and R. Chen (1998). Sequential Monte Carlo methods for dynamic Systems. *Journal of the American Statistical Association 93*(443), 1032–1044.

[31] MacEachern, S. N., M. Clyde, and J. S. Liu (1999). Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics 27*(2), 251–267.

[32] Malefaki, S. and G. Iliopoulos (2008). On convergence of properly weighted samples to the target distribution. *Journal of Statistical Planning and Inference 138*(4), 1210–1225.

[33] Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing 11*(2), 125–139.

[34] Pitt, M., R. Silva, P. Giordani, and R. Kohn (2010). Auxiliary particle filtering within adaptive Metropolis-Hastings sampling. Technical report, Cornell University Library.

[35] Pitt, M. K. and N. Shephard (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association 94*(446), 590–599.

[36] Polson, N. G., J. R. Stroud, and P. Müller (2008). Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(2), 413–428.

[37] Poyiadjis, G., A. Doucet, and S. Singh (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika 98*(1), 65–80.

[38] Robert, C. and G. Casella (2004). *Monte Carlo statistical methods*. Springer Verlag.

[39] Shephard, N. and A. Doucet (2012). Robust inference on parameters via particle filters and sandwich covariance matrices. In *Nuffield College Economics Working Papers*. University of Oxford.

[40] Snyder, C., T. Bengtsson, P. Bickel, and J. Anderson (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review 136*(12), 4629–4640.

[41] Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *Signal Processing, IEEE Transactions on 50*(2), 281–289.

[42] Storvik, G. (2011). On the flexibility of Metropolis–Hastings acceptance probabilities in auxiliary variable proposal generation. *Scandinavian Journal of Statistics 38*(2), 342–358.

[43] Whiteley, N. and A. Lee (2012). Twisted particle filters. Technical report, Cornell University Library.