

Chapter 2

An Overview of Recent Advances in Monte-Carlo Methods for Bayesian Filtering in High-Dimensional Spaces

François Septier and Gareth W. Peters

Abstract Nonlinear non-Gaussian state-space models arise in numerous applications in statistics and signal processing. In this context, one of the most successful and popular approximation techniques is the sequential Monte-Carlo (SMC) algorithm, also known as the particle filter. Nevertheless, this method tends to be inefficient when applied to high-dimensional problems. In this chapter, we present, an overview of recent contributions related to Monte-Carlo methods for sequential simulation from ultra high-dimensional distributions, often arising for instance in Bayesian applications.

2.1 Introduction

In many applications, we are interested in estimating a signal from a sequence of noisy observations. Optimal filtering techniques for general nonlinear and non-Gaussian state-space models are consequently of great interest. Except in a few special cases, including linear and Gaussian state-space models (Kalman filter [26]) and hidden finite-state space Markov chains [7], it is impossible to evaluate this filtering distribution analytically. However, linear systems with Gaussian dynamics are generally inappropriate for the accurate modeling of a dynamical system, since they fail to account for the local nonlinearities in the state space or the dynamic changing nature of the system which is under study. It is therefore increasingly common to consider nonlinear or non-Gaussian dynamical systems. In the case of additive Gaussian errors, one could adopt an Extended Kalman filter (EKF) or in the case of non-Gaussian additive errors, an Unscented Kalman filter (UKF) [25].

F. Septier (✉)

Institut Mines-Télécom/Télécom Lille/CRISTAL UMR 9189, Villeneuve D’ascq, France
e-mail: francois.septier@telecom-lille.fr

G.W. Peters

Department of Statistical Science, University College London, London, UK
e-mail: gareth.peters@ucl.ac.uk

© The Author(s) 2015

G.W. Peters and T. Matsui (eds.), *Theoretical Aspects of Spatial-Temporal Modeling*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55336-6_2

Since the 1990s, sequential Monte Carlo (SMC) approaches have become a powerful methodology to cope with nonlinear and non-Gaussian problems [16]. In comparison with standard approximation methods, such as the EKF, the principal advantage of SMC methods is that they do not rely on any local linearization technique or any crude functional approximation. These particle filtering (PF) methods [23], exploit numerical representation techniques for approximating the filtering probability density function of inherently nonlinear non-Gaussian systems. Using these methods for the empirical characterization of sequences of distributions and the resulting estimators formed based on these empirical estimates can be set arbitrarily close to the optimal solution at the expense of computational complexity.

However, due to their importance sampling-based design, classical SMC methods tend to be inefficient when applied to high-dimensional problems [36, 42]. This issue, known as the curse of dimensionality, has rendered traditional SMC algorithms largely useless in high-dimensional applications such as multiple target tracking, weather prediction, and oceanography. In this chapter, we aim at reviewing recent developments in Monte-Carlo-based techniques that have been specifically designed to deal with high-dimensional systems. The chapter is organized as follows. In Sect. 2.2, we describe the model and the different quantities of interest in dynamic settings. Then, Sect. 2.3 discusses the general principle of SMC methods and their limitations in high-dimensional systems. Several recent developments to improve their performance in this specific setting are then presented. Section 2.4 describes another class of sequential inference algorithms based on the use of Markov chain Monte-Carlo methods (SMCMC) as an alternative to SMC methods. Numerical results are shown in Sect. 2.6. Conclusions are given in Sect. 2.7.

2.2 Problem Formulation

A *hidden Markov model* (HMM) corresponds to a \mathbb{R}^d -valued discrete-time Markov process, $\{X_n\}_{n \geq 1}$ that is not directly observable but we have only access to another \mathbb{R}^{d_y} -valued discrete-time stochastic process, $\{Y_n\}_{n \geq 1}$, which is linked to the *hidden* Markov process of interest. Owing to the Markovian property of the process, the joint distribution of the process $\{X_n\}_{n \geq 1}$,

$$p(x_{1:n}) = \mu(x_1) \prod_{k=1}^n f_k(x_k | x_{k-1}) \quad (2.1)$$

is completely defined by an initial probability density function (pdf) $\mu(x_1)$ and the transition density function at any time k , denoted by $f_k(x_k | x_{k-1})$.

In a HMM, the observed process $\{Y_n\}_{n \geq 1}$ is such that the conditional joint density of $Y_{1:n} = y_{1:n}$ given $X_{1:n} = x_{1:n}$ has the following conditional independence (product) form

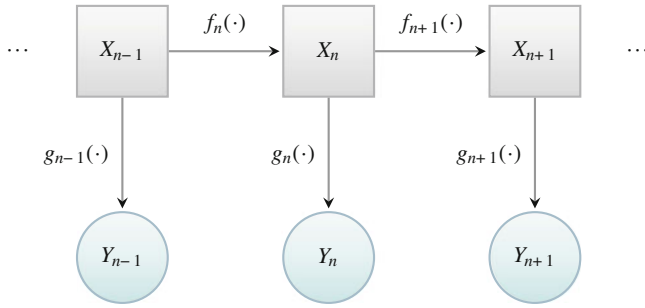


Fig. 2.1 Graphical representation of a hidden Markov model

$$p(y_{1:n}|x_{1:n}) = \prod_{k=1}^n g_k(y_k|x_k). \quad (2.2)$$

The dependence structure of an HMM can be represented by a graphical model shown in Fig. 2.1.

Equations (2.1)–(2.2) define a Bayesian model where (2.1) defines the prior distribution of the “state” process of interest $\{X_n\}_{n \geq 1}$ and (2.2) defines the likelihood function of the conditional observations. One of the most common inference problems, known as *optimal filtering*, which occurs with HMMs is the estimation of the current state value based upon the sequence of observations observed so far. Such inference about X_n given a sequence of the observations $Y_{1:n} = y_{1:n}$ relies upon the posterior distribution,

$$p(x_{1:n}|y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} = \frac{p(x_{1:n})p(y_{1:n}|x_{1:n})}{p(y_{1:n})}. \quad (2.3)$$

This posterior distribution, known also as the *smoothing* distribution, satisfies the following recursion

$$p(x_{1:n}|y_{1:n}) = \frac{g_n(y_n|x_n)f_n(x_n|x_{n-1})}{p(y_n|y_{1:n-1})}p(x_{1:n-1}|y_{1:n-1}), \quad (2.4)$$

where

$$p(y_n|y_{1:n-1}) = \int g_n(y_n|x_n)f_n(x_n|x_{n-1})p(x_{n-1}|y_{1:n-1})dx_{n-1:n}. \quad (2.5)$$

In the literature, this recursion is sometimes presented directly in terms of the marginal posterior distribution, $p(x_n|y_{1:n})$, known as the *filtering* distribution:

$$p(x_n|y_{1:n}) = \frac{g_n(y_n|x_n)p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})}, \quad (2.6)$$

with

$$p(x_n|y_{1:n-1}) = \int f_n(x_n|x_{n-1})p(x_{n-1}|y_{1:n-1})dx_{n-1}. \quad (2.7)$$

However, most sequential Monte-Carlo-based algorithms rely on a numerical approximation of recursion (2.4) instead of (2.6).

2.3 Sequential Monte-Carlo Methods

2.3.1 General Methodology

SMC methods have several variants sometimes appearing under the names of particle filtering or interacting particle systems, e.g. [10, 15, 37], and their theoretical properties have been extensively studied in [8–10, 29].

The general context of a standard SMC method is that one wants to approximate a (often naturally occurring) sequence of target probability density functions (pdf) $\{\pi_n(x_{1:n})\}_{n \geq 1}$ of increasing dimension, i.e. the support of every function in this sequence is defined as $\text{supp}(\pi_n) = \mathbb{R}^{dn}$ and therefore the dimension of its support forms an increasing sequence with n . We may also assume that π_n is only known up to a normalizing constant,

$$\pi_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{Z_n}. \quad (2.8)$$

SMC methods firstly provide an approximation of $\pi_1(x_1)$ and an unbiased estimate of Z_1 , then at the second iteration (“time step” 2) once a new observation is received, an approximation of $\pi_2(x_{1:2})$ is formed as well as an unbiased estimate of Z_2 and this repeats with each distribution in the sequence.

Let us remark at this stage that SMC methods can be used for any sequence of target distributions and therefore the application of SMC to optimal filtering, known as *particle filtering*, is just a special case of this general methodology by choosing $\gamma_n(x_{1:n}) = p(x_{1:n}, y_{1:n})$ and $Z_n = p(y_{1:n})$.

Procedurally, we initialize the algorithm by sampling a set of N particles, $\{X_1^j\}_{j=1}^N$, from the distribution π_1 and set the normalized weights as $W_1^j = 1/N$, for all $j = 1, \dots, N$. If it is not possible to sample directly from π_1 , one should sample from an importance distribution q_1 and calculate its weights according to the importance sampling principle, i.e. $W_1^j \propto \pi_1(X_1^j)/q_1(X_1^j)$. Then, the particles are sequentially propagated thorough each distribution π_t in the sequence via two main processes: mutation and correction (incremental importance weighting). In the first step (mutation), we propagate particles from time $t - 1$ to time t and in the second one (correction) we calculate the new importance weights of the particles.

This method can be seen as a sequence of *importance sampling* steps, where the target distribution at each step n is $\pi_n(x_{1:n})$ and the importance distribution is given by

$$q_n(x_{1:n}) = q_1(x_1) \prod_{k=2}^n q_k(x_k | x_{1:k-1}), \quad (2.9)$$

where $q_k(x_k | x_{1:k-1})$ is the proposal distribution used to propagate particles from time $k - 1$ to k . As a consequence, the unnormalized importance weights are computed recursively by:

$$\begin{aligned} W(x_{1:n}) &= \frac{\gamma_n(x_{1:n})}{q_n(x_{1:n})} \\ &= \frac{\gamma_{n-1}(x_{1:n-1})}{q_{n-1}(x_{1:n-1})} \frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1}) q_n(x_n | x_{1:n-1})} \\ &= W(x_{1:n-1}) \tilde{w}(x_{1:n}), \end{aligned} \quad (2.10)$$

where $\tilde{w}(x_{1:n})$ is known as the *incremental importance weight*. When SMC is applied for the optimal filtering problem with $\gamma_n(x_{1:n}) = p(x_{1:n}, y_{1:n})$, it is straightforward to show by using the recursion of the smoothing distribution in Eq. (2.4) that the incremental importance weight is given by:

$$\tilde{w}(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1}) q_n(x_n | x_{1:n-1})} = \frac{g_n(y_n | x_n) f_n(x_n | x_{n-1})}{q_n(x_n | x_{1:n-1})}. \quad (2.11)$$

At any time n , we obtain an approximation of the target distribution via the empirical measure obtained by the collection of weighted samples, i.e.

$$\hat{\pi}_n(x_{1:n}) = \sum_{j=1}^N W_n^j \delta_{X_{1:n}^j}(dx_{1:n}), \quad (2.12)$$

where W_n^j is the normalized importance weights such that $\sum_{j=1}^N W_n^j = 1$. Moreover, an unbiased estimate of the ratio of two successive normalizing constants is also provided as follows:

$$\frac{\widehat{Z}_n}{\widehat{Z}_{n-1}} = \sum_{j=1}^N W_{n-1}^j \tilde{w}(X_{1:n}^j). \quad (2.13)$$

The algorithm described above is known as the *Sequential Importance Sampling* (SIS) algorithm. However, direct importance sampling on a very large space is rarely efficient as the importance weights exhibit very high variance. As a consequence, SIS will provide estimates whose variance increases exponentially with time n . Indeed, after only a few iterations, all but a few particles will have negligible weights thus leading to the phenomena known as *weight degeneracy*. A well-known criterion to

quantify in an online manner this degeneracy is the *effective sample size* defined as follows:

$$\text{ESS}_n = \frac{1}{\sum_{j=1}^N (W_n^j)^2} \quad (2.14)$$

with $1 \leq \text{ESS}_n \leq N$. In order to overcome this degeneracy problem, a resampling step is thus added in the basic algorithm when the effective sample size drops below some threshold, which as a rough guide is typically in the range of 30–60% of the total number of particles. The purpose of resampling is to reduce this degeneracy by eliminating samples which have low importance weights and duplicating samples with large importance weights [15]. It is quite obvious that when one is interested in the filtering distribution $p(x_n|y_{1:n})$, performing a resampling step at the previous time step will lead to a better level of sample diversity as those particles which were already extremely improbable at time $n - 1$ are likely to have been eliminated and those which remain have a better chance of representing the situation at time n accurately. Unfortunately, when the smoothing distribution is really the quantity of interest, it is more problematic since the resampling mechanism eliminates some trajectories with every iteration, thus leading to problem known as *path or sample degeneracy*. Indeed, resampling will reduce at every iteration the number of distinct samples representing the first time instant of the hidden Markov process. Since in filtering applications, one is generally only interested in the final filtering posterior distribution, this resampling step is widely used in practice at the expense of further diminishing the quality of the path samples. Some strategy that will be discussed in Sect. 2.3.3.1 is generally employed in practice to increase the diversity of the samples.

This SMC algorithm which incorporates a resampling step is often referred to as *Sequential Importance Resampling* (SIR) or *Sequential Importance Sampling and Resampling* (SIS-R). This approach applied for filtering is summarized in Algorithm 1. By assuming that the cost of computing the product of the prior and the likelihood distribution is $\mathcal{O}(d)$ (i.e., a function of the dimension of the hidden state), the cost of the general SMC algorithm is $\mathcal{O}(nNd)$.

2.3.2 Limitations of SMC Methods

In this section, we will discuss the limitations of SMC methods when applied to high-dimensional problems. The main reason why the SIR algorithm performs poorly when the model dimension is high is essentially the same reason why the SIS algorithm behaves badly when the time-horizon is large, and it has to do with the fact that the importance sampling paradigm is typically very inefficient in high-dimensional models. As discussed previously, the SIS algorithm is designed to approximate the smoothing distribution $p(x_{1:n}|y_{1:n})$, weight degeneracy occurs as n increases since the dimension of this target distribution increases with time. Now, if the hidden Markov process is high-dimensional, weight degeneracy will occur as the dimension

Algorithm 1 SMC algorithm for optimal filtering

-
- ```

1: if time $n = 1$ then
2: Sample $X_1^j \sim q_1(x_1), \forall j = 1, \dots, N$
3: Calculate the weights $W_1^j \propto \frac{g_1(Y_1|X_1^j)\mu(X_1^j)}{q_1(X_1^j)}, \forall j = 1, \dots, N$
4: else if time $n \geq 2$ then
5: Sample $X_n^j \sim q_n(x_n|x_{1:n-1})$ and set $X_{1:n}^j := (X_{1:n-1}^j, X_n^j), \forall j = 1, \dots, N$
6: Calculate the weights $W_n^j \propto W_{n-1}^j \frac{g_n(Y_n|X_n^j)f_n(X_n^j|X_{n-1}^j)}{q_n(X_n^j|X_{1:n-1}^j)}, \forall j = 1, \dots, N$
7: end if
8: if $ESS_n < \Gamma$ then
9: Resample $\{W_n^j, X_{1:n}^j\}$ to obtain N equally weighted particles $\{W_n^j = 1/N, X_{1:n}^j\}$
10: end if
11: Output: Approximation of the smoothing distribution via the following empirical measure:

```
- 

$$\pi(x_{1:n}) \approx \sum_{j=1}^N W_n^j \delta_{X_{1:n}^j}(dx_{1:n})$$


---

of this process increases. As a consequence, this degeneracy is seen even in a single iteration of the algorithm. In [4, 42], a careful analysis shows that the collapse phenomenon occurs unless the sample size  $N$  is taken to be exponential in the dimension, which provides a rigorous statement of the curse of dimensionality. Let us remark that a similar weight degeneracy phenomena could be observed in SMC, even in low dimensional models, when for example the noise driving both the dynamics and the observation has very small variance.

The performance of the SMC strongly depends on the choice of the importance distribution. In the literature, the “optimal” proposal distribution in the sense of minimizing the variance of the importance weights is defined as:

$$\begin{aligned} q_n(x_n|x_{n-1}) &= \pi_n(x_n|x_{1:n-1}) \\ &= p(x_n|y_n, x_{n-1}) \quad (\text{in HMM filtering problems}) \end{aligned} \quad (2.15)$$

which leads to the following incremental weight  $\tilde{w}(x_{1:n}) = p(y_n|x_{n-1})$  whose variance conditional upon  $x_{1:n-1}$  is zero since it is independent of  $x_n$ . Unfortunately, in many scenarios, it is impossible to sample from this “optimal” distribution. Many techniques have been proposed to design “efficient” importance distributions  $q_n(x_n|x_{n-1})$  which approximate  $p(x_n|y_n, x_{n-1})$ . In particular, approximations based on the Extended Kalman Filter or the Unscented Kalman Filter to obtain importance distributions are very popular in the literature [6].

While the practical performance of the SIR algorithm can be largely improved by working with importance distributions that are tailored to the specific model being investigated, the benefit is limited to reducing the constants sitting in front of the error bounds, and this technique does not provide a fundamental solution to the curse of

dimensionality [35, 41]. When the optimal importance distribution is used, the curse of dimensionality would indeed still arise due to the recursive nature of the filtering problem.

In the next section, we will describe several strategies that have been proposed in order to improve the performance of standard particle filter in high-dimensional systems.

### 2.3.3 SMC Strategies for High-Dimensional Systems

#### 2.3.3.1 MCMC Moves and the Use of Bridging Densities

The use of Markov Chain Monte-Carlo (MCMC) algorithms within SMC methods is a well-known strategy to improve the filter performance. As discussed previously, repeated resampling stages progressively impoverish the set of particles, by decreasing the number of distinct values represented in that set. This degeneracy problem has historically been addressed using the resample-move algorithm [20] which consists in applying one or more times after the resampling stage an MCMC transition kernel,  $\mathcal{K}_n(x_{1:n}, x'_{1:n})$ , such as a Gibbs sampler or Metropolis–Hastings scheme [38], having  $\pi(x_{1:n})$  as its stationary distribution which means that the following property holds:

$$\int \pi(x_{1:n}) \mathcal{K}_n(x_{1:n}, x'_{1:n}) dx_{1:n} = \pi(x'_{1:n}). \quad (2.16)$$

As a consequence, if the particles  $X_{1:n}^j$  are truly drawn from  $\pi(x_{1:n})$ , then the Markov kernel applied to any of the particles will simply generate new state sequences which are also drawn from the desired distribution. Moreover, even if the particles are not accurately drawn from  $\pi(x_{1:n})$ , the use of such Markov transition kernel will move the particles so that their distribution is closer to the target one (in total variation norm). The use of such MCMC moves can therefore be very effective in reducing the path degeneracy as well as in improving the accuracy of the empirical measure of the posterior distribution. In practice for filtering problems, in order to keep a truly online algorithm with a computational cost linear in time, the Markov transition kernels will not operate on the entire state history, but rather on some fixed time lag  $L$  by only updating the variables  $X_{n-L+1:n}$ .

An interesting generalization of the combination of SMC and MCMC has been proposed in [21] in which the authors propose to introduce a sequence of bridging densities between the initial sampling distribution (generally, the predictive posterior distribution, i.e.,  $p(x_{0:n}|y_{0:n-1})$ ) and the posterior at time  $n$ . By introducing gradually the effect of the likelihood function, the MCMC sampler is thus expected to converge faster, especially when the likelihood for the new data point is centered far from the points sampled from the importance distribution. As a consequence, such strategy could be more effective than standard SMC techniques in high-dimensional



problems. More specifically, the following sequence of  $M \geq 1$  bridging densities is introduced at time  $n$ :

$$\pi_m(x_{1:n}) \propto p(x_{1:n-1}|y_{1:n-1})f_n(x_n|x_{n-1})g_n(y_n|x_n)^{\alpha_m} \quad (2.17)$$

with  $0 \leq \alpha_1 < \dots < \alpha_M = 1$ . In order to move the particles through this sequence of bridging densities, the authors propose to use the framework of the annealed importance sampling [33] (or its generalization the sequential Monte-Carlo sampler [11, 34]). At time  $n$ , the particles are first propagated like in the standard SMC methods using an importance distribution,  $q_n(x_n|x_{1:n-1})$ , and let us denote them by  $X_{0:n,0}^j$  and set  $W_{n,0}^j = W_{n-1}^j$  and  $\pi_0(x_{1:n}) = q_n(x_n|x_{1:n-1})\pi(x_{1:n-1})$ . Then, at step  $m = 1, \dots, M$ , the importance weights are computed as follows:

$$W_{n,m}^j \propto W_{n,m-1}^j \frac{\pi_m(X_{1:n,m-1}^j)}{\pi_{m-1}(X_{1:n,m-1}^j)}. \quad (2.18)$$

Then, a resampling step can be performed if the weights are too degenerate. Finally, each particle is moved independently to obtain  $X_{1:n,m}^j$  using a Markov transition kernel,  $\mathcal{K}_m(x_{1:n,m-1}^j, \cdot)$  having  $\pi_m(x_{1:n})$  as stationary distribution. After the  $M$  steps, a set of weighted samples from the posterior distribution  $\pi(x_{0:n})$  is therefore obtained by setting  $\{W_n^j, X_{0:n}^j\} = \{W_{n,M}^j, X_{0:n,M}^j\}$ . The algorithm is summarized in Algorithm 2 and its cost is  $\mathcal{O}(nNMd)$  by assuming that the cost of computing the product of the prior and the likelihood distribution is  $\mathcal{O}(d)$  as well as the MCMC kernel used. Let us notice that the resample-move algorithm [20] is a special case when  $M = 1$  and also that this scheme is similar to some strategies known as annealed particle filtering [12, 17].

### 2.3.3.2 Local SMC Methods or Block Particle Filter

The underlying idea of these local SMC methods is to partition the state space into separate subspaces of small dimensions and run one SMC algorithm on each subspace. Such strategies have been developed in [13, 14, 32, 36]. Generally, the common assumption used in these approaches is that there exists an ensemble of disjoint sets  $\{D_{n,j}\}_{j=1}^{B_n}$  with  $\cup_{j=1}^{B_n} D_{n,j} = \{1 : d\}$  and  $D_{n,j} \cap D_{n,i} = \emptyset$  for  $i \neq j$ , for some integer  $0 < B_n \leq d$ , such that we can factorize:

$$g_n(y_n|x_n)f_n(x_n|x_{n-1}) = \prod_{j=1}^{B_n} \alpha_{n,j}(y_n, x_{n-1}, x_n(D_{n,j})), \quad (2.19)$$

for appropriate functions  $\alpha_{n,j}(\cdot)$ , where  $x_n(D) = \{x_n(j) : j \in D\} \in \mathbb{R}^{|D|}$  (Fig. 2.2).

By running an SMC algorithm on each nonoverlapping subset, the filtering distribution of interest is therefore approximated at the end of each iteration as follows:

---

**Algorithm 2** SMC algorithm with Bridging densities for optimal filtering
 

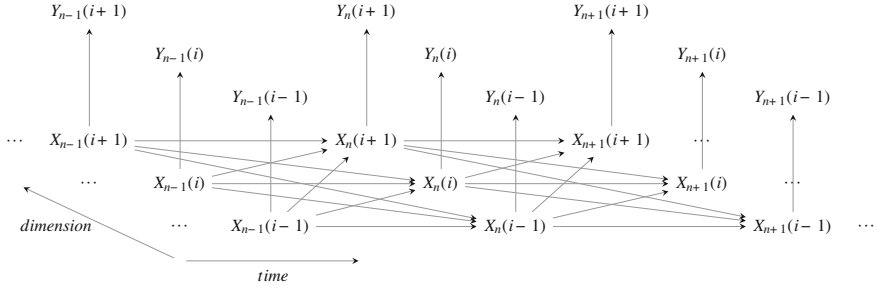
---

**Ensure:**  $n \geq 2$ 

- 1: Sample  $X_{n,0}^j \sim q_n(x_n | X_{1:n-1}^j)$  and set  $X_{1:n,0}^j := (X_{1:n-1}^j, X_{n,0}^j), \forall j = 1, \dots, N$
- 2: Set  $W_{n,0}^j = W_{n-1}^j, \forall j = 1, \dots, N$  and define  $\pi_0(x_{1:n}) = q_n(x_n | x_{1:n-1})\pi(x_{1:n-1})$
- 3: **for**  $m = 1, \dots, M$  **do**
- 4:   Calculate the weights  $W_{n,m}^j \propto W_{n,m-1}^j \frac{\pi_m(X_{1:n,m-1}^j)}{\pi_{m-1}(X_{1:n,m-1}^j)}, \forall j = 1, \dots, N$
- 5:   **if**  $ESS_n < \Gamma$  **then**
- 6:     Resample  $\{W_{n,m}^j, X_{1:n,m-1}^j\}$  to obtain  $N$  equally weighted particles  $\{W_{n,m}^j = 1/N, X_{1:n,m-1}^j\}$
- 7:   **end if**
- 8:   Sample  $X_{1:n,m}^j \sim \mathcal{K}_m(X_{1:n,m-1}^j, x_{1:n}), \forall j = 1, \dots, N$  using  $\mathcal{K}_m(\cdot)$  a Markov kernel having  $\pi_m(\cdot)$  as its stationary distribution.
- 9: **end for**
- 10: Set  $\{W_n^j, X_{1:n}^j\} = \{W_{n,M}^j, X_{1:n,M}^j\}, \forall j = 1, \dots, N$
- 11: **Output:** Approximation of the smoothing distribution via the following empirical measure:

$$\pi(x_{1:n}) \approx \sum_{j=1}^N W_n^j \delta_{X_{1:n}^j}(dx_{1:n})$$


---



**Fig. 2.2** Graphical representation of a hidden Markov model that satisfies an example of a factorization in Eq. (2.19)

$$\pi(x_n) \approx \bigotimes_{j=1}^{B_n} \pi(x_n(D_{n,j})). \quad (2.20)$$

The local SMC method, summarized in Algorithm 3, is well suited to distributed computation as the particles weights are computed locally due to the factorization described in Eq. (2.19). However, this strategy introduces some bias in the algorithm, so that the estimates given by the local SMC method do not converge to the exact filter distributions as the number of particles  $N$  goes to infinity. However, the hope is that by introducing a small amount of bias in the algorithm, its variance can be

reduced significantly since the  $B_n$  SMC algorithms are running on smaller dimension, i.e.,  $|D_{n,j}| \leq d$  for  $j = 1, \dots, B_n$ . Moreover, the local error induced by the approximation of the target distribution as the product of marginals of the nonoverlapping subset is spatially inhomogeneous, i.e., the error will be larger for elements in  $x_n$  closer to subset boundaries. The computational cost of the local particle filter is  $\mathcal{O}(nNd)$  by assuming that the cost of computing the product of the prior and likelihood distribution is  $\mathcal{O}(d)$  (or equivalently that the cost of computing  $\alpha_{n,j}$  is  $\mathcal{O}(|D_{n,j}|)$  for each  $n$  and  $j$ ).

---

**Algorithm 3** Local SMC algorithm

---

- 1: **if** time  $n = 1$  **then**
- 2:   Sample  $X_1^j(D_{1,i}) \sim q_1(x_1(D_{1,i}))$ ,  $\forall j = 1, \dots, N$  and  $\forall i = 1, \dots, B_1$
- 3:   Calculate the weights  $W_{1,i}^j \propto \frac{\alpha_{1,i}(Y_1, X_1^j(D_{1,i}))}{q_1(X_1^j(D_{1,i}))}$ ,  $\forall j = 1, \dots, N$  and  $\forall i = 1, \dots, B_1$
- 4: **else if** time  $n \geq 2$  **then**
- 5:   Sample  $X_{n-1}^j$  from  $\hat{\pi}(x_{n-1})$  for  $j = 1, \dots, N$
- 6:   Sample  $X_n^j(D_{n,i}) \sim q_n(x_n(D_{n,i})|X_{n-1}^j)$   $\forall j = 1, \dots, N$  and  $\forall i = 1, \dots, B_n$
- 7:   Calculate the weights  $W_{n,i}^j \propto \frac{\alpha_{n,i}(Y_n, X_{n-1}^j, X_n^j(D_{n,i}))}{q_n(X_n^j(D_{n,i})|X_{n-1}^j)}$ ,  $\forall j = 1, \dots, N$  and  $\forall i = 1, \dots, B_n$
- 8: **end if**
- 9: **Output:** Approximation of the filtering distribution via the following empirical measure:

$$\hat{\pi}(x_n) = \bigotimes_{i=1}^{B_n} \sum_{j=1}^N W_{n,i}^j \delta_{X_n^j(D_{n,i})}(dx_n(D_{n,i}))$$


---

### 2.3.3.3 Space-Time Particle Filter

The space-time Particle filter (STPF) has been recently proposed in [3]. As in both previous approaches described in Sects. 2.3.3.1 and 2.3.3.2, the idea is to have a gradual introduction of the likelihood  $g_n(y_n|x_n)$  into the successive steps of the algorithm in order to decrease the variance of the importance weights which is the main reason of the collapse of the SMC methods in high-dimensional systems. In this work, the authors assume there exists an increasing sequence of sets  $\{A_{n,j}\}_{j=1}^{B_n}$  with  $A_{n,1} \subset A_{n,2} \subset \dots \subset A_{n,B_n} = \{1 : d\}$ , for some integer  $0 < B_n \leq d$ , such that we can factorize:

$$g_n(y_n|x_n) f_n(x_n|x_{n-1}) = \prod_{j=1}^{B_n} \alpha_{n,j}(y_n, x_{n-1}, x_n(A_{n,j})) \quad (2.21)$$

for appropriate functions  $\alpha_{n,j}(\cdot)$ , where  $x_n(A) = \{x_n(j) : j \in A\} \in \mathbb{R}^{|A|}$ . Let us remark that the assumption required for this factorization is weaker than the one

described in Eq. (2.19) for the local SMC methods since some dependencies between elements of  $x_n$  from different subsets given  $x_{n-1}$  are allowed. As discussed by the authors in their paper, this factorization is not a requirement but in such cases the performance of the filter will be degraded as additional sampling and reweighting steps are necessary.

The underlying idea of the STPF is to exploit the structure in Eq. (2.21) to design a particle filter moving along both the space and time index (as opposed to traditional particle filter that moves only along the time index). This approach can also be viewed as a generalization of the particle island particle filter proposed in [43] since the STPF combines a local filter running  $B_n$  space-step using  $M$  particles with a global particle filter making time steps using  $N$  particles. The authors show that this algorithm is asymptotically consistent and has a subexponential cost in  $d$ . Since the local particle filters are running along the space dimension, a patch degeneracy (on the space dimension) effect can be expected as the dimension of the system increases. The authors describe different strategies based on MCMC rejuvenation that could be employed to improve the performance of the algorithm at the expense of additional cost. The algorithm is summarized in Algorithm 4 and its computational cost is  $\mathcal{O}(nNMd)$  by assuming that the cost of computing  $\alpha_{n,j}$  is  $\mathcal{O}(|A_{n,j}|)$  for each  $n$  and  $j$ . More specifically, the authors present results that, in 1) an i.i.d. scenario both in time and space and 2) a Markovian model along space, the algorithm is stable by setting the number of particles in the local systems equal to the dimension of the system (i.e.,  $M = d$ ), thus leading in that case to a cost of  $\mathcal{O}(nNd^2)$ .

## 2.4 Sequential Markov Chain Monte Carlo

In this section, another class of sequential Bayesian algorithm based on MCMC sampling (unlike importance sampling as in the previous section) is described. MCMC methods are generally more effective than importance sampling techniques in high-dimensional spaces. Their traditional formulation, however, allows sampling from probability distributions in a nonsequential fashion. Recently, advanced sequential MCMC schemes were proposed in [2, 5, 22, 27, 40] for solving online filtering inference problems. These approaches are distinct from the technique described previously in Sect. 2.3.3.1 where the MCMC algorithm is used to move samples following importance sampling resampling since these sequential MCMC use neither resampling nor importance sampling.

### 2.4.1 General Principle

Several sequential MCMC (SMCMC) methods have been proposed in the literature recently. In this section, we will describe a general framework that include all of them. The underlying idea of all these SMCMC approaches is to perform a

**Algorithm 4** Space-Time Particle Filter

- 
- 1: Define the subset  $\Omega_i = A_{1,i} \setminus A_{1,i-1}$
  - 2: **if** time  $n = 1$  **then**
  - 3:   **for**  $j = 1, \dots, N$  **do**
  - 4:     **for**  $i = 1, \dots, B_1$  **do**
  - 5:       Sample  $X_1^{j,l}(\Omega_i) \sim q_1(x_1(\Omega_i) | X_1^{j,l}(A_{1,i-1})), \forall l = 1, \dots, M$
  - 6:       Set  $X_1^{j,l}(A_i) = [X_1^{j,l}(A_{1,i-1}) \ X_1^{j,l}(\Omega_i)]$
  - 7:       Calculate the weights  $w_i^{j,l} = \frac{\alpha_{1,i}(y_1, \cdot, X_1^{j,l}(A_{1,i}))}{q_1(X_1^{j,l}(\Omega_i) | X_1^{j,l}(A_{1,i-1}))}, \forall l = 1, \dots, M$
  - 8:       Resample local particles  $\{X_1^{j,l}(A_{1,i})\}_{l=1}^M$  according to their normalized weights  $\left\{w_i^{j,l} \left[\sum_{k=1}^M w_i^{j,k}\right]^{-1}\right\}_{l=1}^M$
  - 9:     **end for**
  - 10:   **end for**
  - 11: Resample the  $N$ -particle systems, i.e.,  $\{X_1^{j,1:M}\}_{j=1}^N$ , according to their weights defined as:

$$W_1^j \propto \prod_{i=1}^{B_1} \frac{1}{M} \sum_{l=1}^M w_i^{j,l}$$

- 12: **else if** time  $n \geq 2$  **then**
- 13:   **for**  $j = 1, \dots, N$  **do**
- 14:     **for**  $i = 1, \dots, B_n$  **do**
- 15:       Sample  $X_n^{j,l}(\Omega_i) \sim q_1(x_n(\Omega_i) | X_n^{j,l}(A_{n,i-1}), X_{n-1}^{j,l}), \forall l = 1, \dots, M$
- 16:       Set  $X_n^{j,l}(A_{n,i}) = [X_n^{j,l}(A_{n,i-1}) \ X_n^{j,l}(\Omega_i)]$
- 17:       Calculate the weights  $w_i^{j,l} = \frac{\alpha_{n,i}(y_n, X_{n-1}^{j,l}, X_n^{j,l}(A_{n,i}))}{q_n(X_n^{j,l}(\Omega_i) | X_n^{j,l}(A_{n,i-1}), X_{n-1}^{j,l})}, \forall l = 1, \dots, M$
- 18:       Resample local particles  $\{X_{n-1}^{j,l}, X_n^{j,l}(A_{n,i})\}_{l=1}^M$  according to their normalized weights  $\left\{w_i^{j,l} \left[\sum_{k=1}^M w_i^{j,k}\right]^{-1}\right\}_{l=1}^M$
- 19:     **end for**
- 20:   **end for**
- 21: Resample the  $N$ -particle systems, i.e.,  $\{X_{1:n}^{j,1:M}\}_{j=1}^N$ , according to their weights defined as:

$$W_n^j \propto \prod_{i=1}^{B_n} \frac{1}{M} \sum_{l=1}^M w_i^{j,l}$$

22: **end if**

23: **Output:** Approximation of the filtering distribution via the following empirical measure:

$$\pi(x_n) \approx \frac{1}{NM} \sum_{j=1}^N \sum_{l=1}^M \delta_{X_n^{j,l}}(dx_n)$$


---

Metropolis–Hastings (MH) accept-rejection step as a correction for having used a proposal distribution to sample the current state in order to approximate the posterior target distribution as opposed to SMC methods that use a correction based on Importance sampling.

At time step  $n$ , the target distribution of interest to be sampled from is

$$\underbrace{p(x_{1:n}|y_{1:n})}_{\pi_n(x_{1:n})} \propto g_n(y_n|x_n) f_n(x_n|x_{n-1}) \underbrace{p(x_{1:n-1}|y_{1:n-1})}_{\pi_{n-1}(x_{1:n-1})}. \quad (2.22)$$

Unfortunately, it is impossible to sample from  $p(x_{1:n-1}|y_{1:n-1})$  since this distribution is analytically intractable. The key idea of all existing SMC methods is therefore to replace  $p(x_{1:n-1}|y_{1:n-1})$  by an empirical approximation obtained from previous iterations of the algorithm. The target distribution of interest at time step  $n$  is therefore defined as:

$$\pi_n(x_{1:n}) \propto g_n(y_n|x_n) f_n(x_n|x_{n-1}) \hat{\pi}_{n-1}^i(x_{1:n-1}), \quad (2.23)$$

with

$$\hat{\pi}_{n-1}^i(x_{1:n-1}) = \frac{1}{i - N_b} \sum_{m=N_b+1}^i \delta_{X_{n-1,1:n-1}^m}(dx_{1:n-1}), \quad (2.24)$$

where  $\{X_{n-1,1:n-1}^m\}_{m=1}^i$  corresponds to the  $i$  samples of the  $(n-1)$ th Markov chain, whose distribution is  $\pi_{n-1}(x_{1:n-1})$  as defined in Eq. (2.23) that has been generated until the current iteration of the MCMC at time step  $n$  ( $N_b$  represents the length of the burn-in period). By using this empirical approximation of the previous target distribution, an MCMC kernel can be employed in order to obtain a Markov chain, denoted by  $(X_{n,1:n}^1, X_{n,1:n}^2, \dots)$ , with stationary distribution  $\pi_n(x_{1:n})$  as defined Eq. (2.23).

As summarized in Algorithm 5, the SMC proceeds as follows. At time step  $n = 1$ , an MCMC kernel  $\mathcal{K}_1$  of invariant distribution  $\pi_1(x_1) \propto g_1(y_1|x_1)\mu(x_1)$  is employed to generate a Markov chain denoted by  $(X_{1,1}^1, \dots, X_{1,1}^{N+N_b})$ . At time step  $n$ , the  $N + N_b$  iterations of the SMC aims at producing a Markov chain, denoted by  $(X_{n,1:n}^1, \dots, X_{n,1:n}^{N+N_b})$ , by using an MCMC kernel  $\mathcal{K}_n$  of invariant distribution  $\pi_n(x_{1:n})$  as defined in Eq. (2.23). Moreover, samples can be added to the previous  $L$  Markov chains, i.e.,  $X_{n-L,1:n-L}$  with  $L > 1$ , in order to improve the empirical approximation  $\hat{\pi}_{n-1}(x_{1:n-1})$  required in the posterior distribution of interest at time step  $n$ . Once the  $n$ th Markov chain has been generated, the last  $N$  are extracted to obtain the empirical approximation of the filtering distribution:

$$p(x_n|y_{1:n}) \approx \frac{1}{N} \sum_{m=N_b+1}^{N+N_b} \delta_{X_{n,n}^m}(dx_n). \quad (2.25)$$

By assuming that the computation of the product of likelihood and prior as well as the MCMC kernel used is  $\mathcal{O}(d)$ , the cost of this algorithm is  $\mathcal{O}(nLNd)$  since the length of the burn-in period is generally considered to be a percentage of the useful samples, i.e.,  $N_b = \beta N$  with  $0 \leq \beta \leq 1$ .

---

**Algorithm 5** Generic Sequential MCMC algorithm for optimal filtering

---

```

1: Initialization $\{i_n\}_{n \geq 0} = 0$
2: if time $n = 1$ then
3: for $j = 1, \dots, N + N_b$ do
4: Set $i_1 = i_1 + 1$
5: Sample $X_{1,1}^{i_1} \sim \mathcal{K}_1(x_1^{i_1-1}, \cdot)$ with \mathcal{K}_1 an MCMC kernel of invariant distribution $\pi_1(x_1) \propto g_1(y_1|x_1)\mu(x_1)$.
6: end for
7: else if time $n \geq 2$ then
8: for $j = 1, \dots, N + N_b$ do
9: for $k = \max(1, n - L + 1), \dots, n$ do
10: Set $i_k = i_k + 1$
11: Sample $X_{k,1:k}^{i_k} \sim \mathcal{K}_k^{(i_{k-1})}(X_{k,1:k}^{i_{k-1}}, \cdot)$ with $\mathcal{K}_k^{(i_{k-1})}$ an MCMC kernel of invariant distribution $\pi_k^{(i_{k-1})}$ given by:

```

$$\pi_k^{(i_{k-1})}(x_{1:k}) \propto g_k(y_k|x_k) f_k(x_k|x_{k-1}) \widehat{\pi}_{k-1}^{(i_{k-1})}(x_{1:k-1})$$

with  $\widehat{\pi}_{k-1}^{i_{k-1}}$  being the empirical measure obtained using previous samples, i.e.

$$\widehat{\pi}_{k-1}^{(i_{k-1})}(x_{1:k-1}) = \frac{1}{i_{k-1} - N_b} \sum_{m=N_b+1}^{i_{k-1}} \delta_{X_{k-1,1:k-1}^m}(dx_{1:k-1})$$

```

12: end for
13: end for
14: end if
15: Output: Approximation of the smoothing distribution with the following empirical measure:

```

$$\pi(\mathbf{x}_n) \approx \frac{1}{N} \sum_{j=N_b+1}^{N+N_b} \delta_{X_{n,1:n}^j}(dx_{1:n})$$


---

### 2.4.2 Algorithm Settings

The overall performance of the SMCMC algorithm applied to optimal filtering depends heavily upon the choice of the MCMC kernel. One of the attractive features of this SMCMC is to be able to employ all the different MCMC methods that have been proposed in the scientific literature. In practical implementation of the SMCMC and more especially for high-dimensional systems, composite kernel

based on joint and conditional draws are generally very efficient [40]. Summarized in Algorithm 6, such a composite kernel is based on the following two main steps:

1. A joint draw in which a Metropolis–Hastings sampler is used to update all the path of states corresponding to  $x_{1:n}$
2. A refinement step in which previous history  $x_{1:n-1}$  and current state  $x_n$  are updated successively. Moreover, if  $x_n$  is high-dimensional, an efficient way to update it consists in firstly dividing its space into  $P$  disjoint subsets and update them successively either via a random scan or a deterministic scan using a series of block MH-within—Gibbs steps.

The cost of this MCMC kernel is  $\mathcal{O}(d)$  if a factorization such as the one defined in Eq. (2.21) is valid.

As a comparison, Berzuini et al. [2] made only use of the individual refinement step described above (with  $L = 1$  in Algorithm 5). This can potentially lead to poor mixing in high-dimensional problems due to the highly disjoint predictive density of the particle representation. On the other hand, Golightly and Wilkinson [22] made use of only the joint draw to move the MCMC chain. This can potentially reduce the effectiveness of the MCMC as refinement moves are not employed to explore the structured probabilistic space which is very challenging in high-dimensional systems. Indeed, it could be difficult to design a proposal distribution for the joint draw that does not lead to low acceptance rate. In [5], the authors propose a general framework of SMC with the possibility of updating previous Markov chains (i.e.,  $L > 1$  in Algorithm 5). An independent Metropolis–Hastings sampler as MCMC kernel is employed. By doing so, the ratio of the normalizing constant  $Z_n/Z_{n-1}$  can be easily estimated but it could be difficult to design the independent proposal distribution leading to satisfactory performance. Finally, in [39], the authors proposed to incorporate several attractive features of population-based MCMC methods [19, 31] such as genetic moves and simulated annealing in order to improve the mixing of the Markov chain in complex scenarios.

## 2.5 Assessing Local and Global Sample Effective Sample Size and SMC Convergence Diagnostics

Each of the discussed algorithms: SMC with MCMC moves; local SMC; Space–Time SMC; and sequential MCMC will have different features with regard to the effective sample size produced and even how one may consider the effective sample size under each class of algorithm requires further consideration. In this section, we provide a brief overview of two aspects, firstly how to determine the number of independent samples present in the resulting set of samples or particles. Then, we also discuss some convergence diagnostics in standard MCMC that may be adapted for the setting of SMC to adaptively modify the past “population” MCMC chains in the sequence.



**Algorithm 6** Example of an MCMC Kernel  $\mathcal{K}_n^{(i_{n-1})}(X_{n,1:n}^{i_{n-1}}, \cdot)$  for the SMCMC

---

```

1: Joint Draw
2: Propose $\{X_{n,1:n}^*\} \sim q_1(x_{1:n}|X_{n,1:n}^{i_{n-1}})$
3: Compute the MH acceptance probability $\rho_1 = \min \left(1, \frac{\pi_n^{(i_{n-1})}(X_{n,1:n}^*)}{q_1(X_{n,1:n}^*|X_{n,1:n}^{i_{n-1}})} \frac{q_1(X_{n,1:n}^{i_{n-1}}|X_{n,1:n}^*)}{\pi_n^{(i_{n-1})}(X_{n,1:n}^{i_{n-1}})} \right)$
4: Accept $X_{n,1:n}^{i_n} = X_{n,1:n}^*$ with probability ρ_1 otherwise set $X_{n,1:n}^{i_n} = X_{n,1:n}^{i_{n-1}}$
5: Refinement
6: Propose $\{X_{n,1:n-1}^*\} \sim q_{R,1}(x_{1:n-1}|X_{n,1:n}^{i_n})$
7: Compute the MH acceptance probability $\rho_{R,1} =$

$$\min \left(1, \frac{\pi_n^{(i_{n-1})}(X_{n,1:n-1}^*, X_{n,n}^{i_n})}{q_{R,1}(X_{n,1:n-1}^*|X_{n,1:n}^{i_n})} \frac{q_{R,1}(X_{n,1:n-1}^{i_n}|X_{n,1:n-1}^*, X_n^{i_n})}{\pi_n^{(i_{n-1})}(X_{n,1:n-1}^{i_n})} \right)$$

8: Accept $X_{n,1:n-1}^{i_n} = X_{n,1:n-1}^*$ with probability $\rho_{R,1}$.
9: Randomly divide x_n into P disjoint blocks $\{\Omega_p\}_{p=1}^P$ such that $\bigcup_p \Omega_p = \{1 : d\}$
10: for $p = 1, \dots, P$ do
11: Propose $\{X_{n,n}^*(\Omega_p)\} \sim q_{R,p}(x_n(\Omega_p)|X_{n,1:n}^{i_n})$
12: Compute the MH acceptance probability $\rho_{R,p} =$

$$\min \left(1, \frac{\pi_n^{(i_{n-1})}(X_{n,n}^*(\Omega_p), X_{n,n}^{i_n}(\mathbb{R}^d \setminus \Omega_p), X_{n,1:n-1}^{i_n})}{q_{R,p}(X_{n,n}^*(\Omega_p)|X_{n,1:n}^{i_n})} \frac{q_{R,p}(X_{n,n}^{i_n}(\Omega_p)|X_{n,n}^*(\Omega_p), X_{n,n}^{i_n}(\mathbb{R}^d \setminus \Omega_p), X_{n,1:n-1}^{i_n})}{\pi_n^{(i_{n-1})}(X_{n,1:n}^{i_n})} \right)$$

13: Accept $X_{n,n}^{i_n}(\Omega_p) = X_{n,n}^*(\Omega_p)$ with probability $\rho_{R,p}$
14: end for

```

---

### 2.5.1 Assessing Local and Global Sample Effective Sample Size for SMCMC

We start by briefly recalling the properties of effective sample size in the standard markov chain setting before talking about these in the context of the three classes of algorithms we consider in the high-dimensional state space models discussed in this chapter.

In general for a correlated time series one may define the effective sample size which goes back to early studies such as those by [30] who studies the time between effectively independent samples or the reciprocal effective number of independent samples in a time span which is often referred to as the effective sample size (ESS). A simple definition of such a quantity is to equate the ensemble mean square of a time-averaged mean denoted by  $\sigma_{\bar{X}}^2$  which is based on the autocovariance function (acf) to the standard formula for the variance of the mean of independent samples. The solution to the number of independent samples is one measure of ESS. To proceed consider the  $N$  values from a time series  $X_1, \dots, X_N$  of a stationary stochastic process with variance  $\sigma^2$ , then one can write the ensemble mean as follows (see [1]) with respect to the mean  $\mu$  and symmetric covariance between observations lagged by a time interval  $\tau$  denoted  $C(\tau)$  and corresponding lag-correlation function  $\rho(\tau)$  according to:

$$\begin{aligned}
\sigma_X^2 &= \frac{1}{N} \sum_{i,j}^N \langle (X_i - \mu)(X_j - \mu) \rangle = \frac{1}{N^2} \sum_{i,j}^N C(i-j) \\
&= \frac{1}{N^2} \sum_{\tau=-(N-1)}^{(N-1)} [N - |\tau|] C(\tau) \quad (2.26) \\
&= \frac{\sigma_2}{N} \sum_{\tau=-(N-1)}^{(N-1)} \left[ 1 - \frac{|\tau|}{N} \right] \rho(\tau).
\end{aligned}$$

If one then considers the independence case with  $N'$  samples then one would have had the variance of the sample mean given instead by  $\sigma^2/N'$ , by equating these one obtains the effective sample size

$$N' = \frac{\sigma^2}{\sigma_X^2} = N \left[ \sum_{\tau=-(N-1)}^{(N-1)} \left[ 1 - \frac{|\tau|}{N} \right] \rho(\tau) \right]^{-1}. \quad (2.27)$$

Therefore, such an estimator is typically used for standard MCMC settings where the ESS is defined for a MCMC sample of size  $N$  by

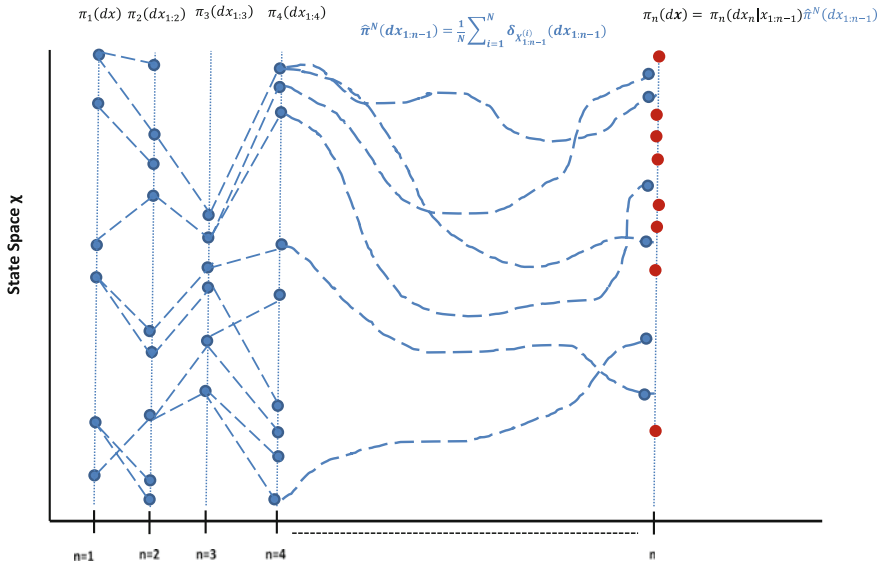
$$\text{ESS}^{\text{MCMC}} = \frac{N}{1 + 2 \sum_{k=1}^{\infty} \rho_k}. \quad (2.28)$$

In the context of Markov chain Monte-Carlo methods, this framework can be adopted to study the asymptotic variance of the mean of a Markov chain, with respect to a bounded and integrable test function generically denoted by  $\varphi$ , under the central limit theorem. In this case, one can state the following results. Let  $X = \{X_i : i = 0, 1, 2, \dots\}$  be a Harris ergodic Markov chain on a general space  $\chi$  with invariant probability distribution  $\pi$  having support  $\chi$ . Let  $\varphi$  be a Borel function and define  $\bar{\varphi}_T := \frac{1}{T} \sum_{i=1}^T \varphi(X_i)$  and  $\mathbb{E}_\pi[\varphi] := \int_\chi \varphi(x) \pi(dx)$ . When  $\mathbb{E}_\pi[|\varphi|] < \infty$  the ergodic theorem guarantees that  $\bar{\varphi}_T \rightarrow \mathbb{E}_\pi[\varphi]$  with probability 1 as  $T \rightarrow \infty$ . The conditions on the Markov chain for this convergence result to hold are stated for a range of MCMC methods in [24] and the following general CLT result applies under these different conditions:

$$\sqrt{T} (\bar{\varphi}_T - \mathbb{E}_\pi[\varphi]) \xrightarrow{d} N(0, \sigma_\varphi^2). \quad (2.29)$$

Here, the asymptotic variance of the estimated mean of a test function  $\varphi$  is given by the finite variance given by

$$\sigma_\varphi^2 := \mathbb{V}\text{ar}_\pi[\varphi(X_0)] + 2 \sum_{i=1}^{\infty} \mathbb{C}\text{ov}_\pi[\varphi(X_0), \varphi(X_i)]. \quad (2.30)$$



**Fig. 2.3** SMC sampler construction

If one selects  $\varphi(X) = X$  and applies a truncation to the number of Markov chain samples to  $N$ , then after renormalization, this is exactly the expression obtained in Eq. (2.26). This is the typical framework used to understand effective sample size and also it acts as a core ingredient in the derivation of the convergence diagnostics for MCMC samplers. In the following, we will explain how to adapt this classical MCMC ESS framework to the case of the SMC setting.

In the SMC setting, we are sampling sequentially the target distribution sequence  $\{\pi_n\}_{n \in \mathbb{N}}$  via a sequence of Markov chains constructed through a Metropolis–Hastings accept reject framework, however, the sequence of chains are constructed based on the previous sequence path-space genealogies. To understand this we refer to Fig. 2.3 where the blue “particle” trajectories correspond to the previously sampled path-space genealogies for the SMC algorithm that comprise the empirical measure  $\hat{\pi}_{n-1}(x_{1:n-1})$  for the construction of the sampler at target  $\pi_n$ . The blue trajectories are the previously accepted sequences of Markov chain samples that have been accepted, so that at iteration  $n$  we would randomly (with replacement) draw a trajectory path, then construct conditionally on this path a new state sample denoted in red which would be accepted or rejected based on a Metropolis–Hastings accept reject mechanism. We will refer to the previous genealogical paths used in the proposal at time  $n$  by the set of path-space branches  $\chi_n$  (in blue).

We can see from this representation that one needs to develop an effective sample size criterion for the SMC algorithm that would adequately reflect the effect of the genealogical path-space behaviour used to construct the sequence of distributions

sampled. To achieve this we consider propose the forward and backward SMCMC effective sample size criterions.

**Definition 1** (*Forward Efficiency of SMCMC*) Consider the path-space genealogy  $\chi_n$  at distribution sequence iteration  $n$ , the conditional forward efficiency  $\eta_n \in [0, 1]$  of the algorithm having sampled  $N$  MCMC iterations is given, for a bounded integrable test function  $\varphi$  by

$$\begin{aligned} \eta_n(\varphi; \chi_n) &:= \frac{\text{Var}(\bar{\varphi}_{N'}^{\text{i.i.d.}} | \chi_n)}{\text{Var}(\bar{\varphi}_N | \chi_n)} \\ &= \frac{1}{1 + 2 \sum_{k=1}^{\infty} \rho_k(\chi_n)} \end{aligned} \quad (2.31)$$

where  $\rho_k(\chi_n)$  denotes the autocorrelation which is implicitly dependent on the genealogical path-space  $\chi_n$ ,  $\bar{\varphi}_N$  is the mean estimated from  $N$  correlated MCMC samples from  $\pi_n(dx_{1:n} | \chi_n) = \pi_n(dx_n | \chi_n) \hat{\pi}^N(dx_{1:n-1})$  and  $\bar{\varphi}_{N'}^{\text{i.i.d.}}$  is the estimator for the optimal case of i.i.d. samples from  $\pi_n(dx_{1:n} | \chi_n)$  with  $N' \leq N$ .

For this forward measure of efficiency of the SMCMC algorithm, which can be computed online for each target distribution  $\pi_n$  one can approximate this efficiency measure by the estimator given by first constructing from the  $N$  correlated MCMC samples  $\varphi_i := \varphi(X_{n,1:n}^i)$  the autocovariance function

$$\hat{\gamma}_\varphi(k; \chi_n) = \frac{1}{N} \sum_{i=1}^{N-|k|} (\varphi_{i+|k|} - \bar{\varphi}_N) (\varphi_i - \bar{\varphi}_N), \quad -N < k < N. \quad (2.32)$$

This would then lead to the estimator for the autocorrelations given by

$$\hat{\rho}_\varphi(k; \chi_n) = \frac{\hat{\gamma}_\varphi(k; \chi_n)}{\hat{\gamma}_\varphi(0; \chi_n)}, \quad (2.33)$$

giving the estimator for the efficiency at stage  $n$  in Eq. 2.31 by substitution. However, it may also be of interest to consider the efficiency in another sense, to capture the path-space implicit effect on the SMCMC. To achieve this, we consider also the backward efficiency at stage  $n$  conditional on the SMCMC samples at stage  $n$ , this is given in the following definition.

**Definition 2** (*Backward Efficiency of SMCMC*) Consider, the path-space genealogy  $\chi_n$  at distribution sequence iteration  $n$  decomposed as  $\chi_n = \chi_{n-1} \cup \{X_{n-1,n-1}^i\}_{i=1}^N$  such that  $\chi_{n-1} \subseteq \chi_n$ , then the conditional one-stage backward efficiency viewed from iteration  $n$  is given by  $\bar{\eta}_n \in [0, 1]$ , for a bounded integrable test function  $\varphi$  according to

$$\overleftarrow{\eta}_n(\varphi; \chi_{n-1}) := \int \eta_n(\varphi; \{X_{n-1,n-1}^i\}_{i=1}^N \cup \chi_{n-1}) \pi_{n-1}(dx_{n-1,1:n-1} | \chi_{n-1}). \quad (2.34)$$

This can then be defined recursively for any number of backward looking steps from distribution sequence iteration  $n$ .

One can approximate this using the path-space samples obtained in each iteration according to the following estimator

$$\widehat{\overleftarrow{\eta}}_n(\varphi; \chi_{n-1}) = \frac{1}{J} \sum_{j=1}^J \eta_n^{(j)}(\varphi; \{X_{n-1,n-1}^i\}_{i=1}^N \cup \chi_{n-1}). \quad (2.35)$$

where  $\eta_n^{(j)}(\varphi; \{X_{n-1,n-1}^{(i,j)}\}_{i=1, j=1}^{N,J} \cup \chi_{n-1})$  is obtained using the estimator at time  $n$

of efficiency in Eq. 2.31 for the  $j$ th population sample of  $\{X_{n-1,n-1}^{(i,j)}\}_{i=1}^{N,J}$  conditional on previous genealogical paths in  $\chi_{n-1}$ , a visual representation of how this estimator is obtained based on resampling of the previous generation at time  $n-1$  is provided in Fig. 2.4. This illustration shows in red the resampled path genealogies for iteration  $n$  looking back in this case one time step to iteration  $n-1$  parents and regenerating these  $j \in \{1, \dots, J\}$  times, with the  $j$ th regeneration producing the backward

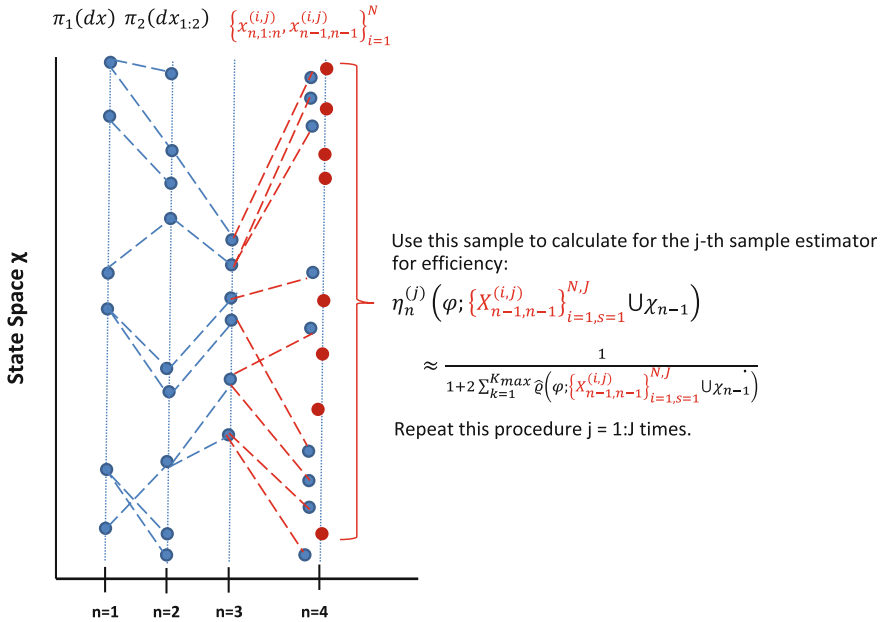


Fig. 2.4 SMC sampler backward efficiency measure estimator

looking effective sample size one step back approximation obtained by the estimated autocorrelations.

In addition to these two measure of efficiency, one can also monitor at iteration  $n$  a related quantity that can be estimated at each MCMC iteration of the SMCMC algorithm at distribution sequence iteration  $n$  to decide if one should perform more local or more global moves. This involves adapting the following well-known Geweke [18] convergence diagnostic for MCMC methods can be adopted in the SMCMC sampler setting at each iteration as follows. If the total chain has length  $N + N_b$ , the initial burn-in stage will correspond to the first  $N_b$  samples. We denote by  $\{X_{n,i}^{(t)}\}_{t=1:N}$  the Markov chain of the  $i$ th parameter after burn-in. The diagnostics we consider are given by:

- For state  $X_{n,i}$  it is calculated as follows:
  1. Split the Markov chain samples into two sequences,  $\{X_{n,i}^{(t)}\}_{t=1:N_1}$  and  $\{X_{n,i}^{(t)}\}_{t=N^*:N}$ , such that  $N^* = N - N_2 + 1$ , and with ratios  $N_1/N$  and  $N_2/N$  fixed such that  $(N_1 + N_2)/N < 1$  for all  $N$ .
  2. Evaluate  $\hat{\mu}(X_{n,i}^{N_1})$  and  $\hat{\mu}(X_{n,i}^{N_2})$  corresponding to the sample means on each subsequence.
  3. Evaluate consistent spectral density estimates for each subsequence, at frequency 0, denoted  $\widehat{SD}(0; N_1, X_{n,i})$  and  $\widehat{SD}(0; N_2, X_{n,i})$ . The spectral density estimator considered in this paper is the classical nonparametric periodogram or power spectral density estimator. We use Welch's method with a Hanning window.
  4. Evaluate convergence diagnostic given by
 
$$Z_N = \frac{\hat{\mu}(X_{n,i}^{N_1}) - \hat{\mu}(X_{n,i}^{N_2})}{N_1^{-1}\widehat{SD}(0; N_1, X_{n,i}) + N_2^{-1}\widehat{SD}(0; N_2, X_{n,i})}.$$
 According to the central limit theorem, as  $N \rightarrow \infty$  one has that  $Z_N \rightarrow \mathcal{N}(0, 1)$  if the sequence  $\{X_{n,i}^{(t)}\}_{t=1:N}$  is stationary.

Note, this can be monitored and tested online for each stage and each parameter subspace of the SMCMC algorithm at iteration  $n$  in the distribution sequence to decide if one should sample more local moves or more global moves.

### 2.5.2 Effective Sample Size for SMC Methods

In the SMC literature, the notion of ESS that is typically adopted is based on the approach discussed for standard SMC algorithms of [28]. In this framework, an approximation to the effective sample size of the filtering distribution is obtained at time  $t$ . To understand this approximation we first define:

- the estimated sample mean from the filtering distribution weighted particle population given by samples drawn from mutation kernel  $q(\cdot)$ ,

$$\widehat{\mathbb{E}}_q^N = \frac{1}{N} \sum_{i=1}^N W(X^{(i)}) \varphi(X^{(i)}); \quad (2.36)$$

- the estimated sample mean from the filtering distribution given by samples drawn from the true filtering target distribution  $\pi(\cdot)$ ,

$$\widehat{\mathbb{E}}_\pi^N = \frac{1}{N} \sum_{i=1}^N \varphi(X^{(i)}). \quad (2.37)$$

Then in the standard SMC setting one typically starts by considering the ratio of the following two sample mean variances and applies a Taylor series expansion and applies the Delta method to obtain

$$\begin{aligned} \eta^{SMC} &= \frac{\text{Var}_\pi [\widehat{\mathbb{E}}_\pi^N]}{\text{Var}_q [\widehat{\mathbb{E}}_q^N]} \approx (1 + \text{Var}_q [W(X)])^{-1} \\ &= (\mathbb{E}_q [W(X)^2])^{-1}. \end{aligned} \quad (2.38)$$

## 2.6 Numerical Simulations

In this section, we study the empirical performance of the different algorithms that have been previously described, namely: (a) SMC in Algorithm 1—(b) SMC-MCMC in Algorithm 2—(c) Local SMC in Algorithm 3— (d) STPF in Algorithm 4, and (e) SMC-MCMC in Algorithm 5 and 6. For all the different SMC-based algorithms, the resampling step is performed when the effective sample size, ESS, is below  $N/2$ . All the proposal distributions required in these algorithms are based on the prior distributions. The MCMC kernel used in the SMC-MCMC defined in Algorithm 2 correspond to series of  $P$  Metropolis–Hastings within Gibbs samplers used in the SMC-MCMC and described in lines 9–13 of Algorithm 6. The parametric function used for the cooling schedule strategy used to design the sequence of bridging densities within the SMC-MCMC in this section is defined as, for  $m = 1, \dots, M$  by the sequence:

$$\alpha_m = \frac{\exp(\gamma m/M) - 1}{\exp(\gamma) - 1} \quad (2.39)$$

with  $\gamma = 5$ . In results presented for the SMC-MCMC algorithm, only the current posterior distribution is updated, i.e.,  $L = 1$ .

### 2.6.1 Linear and Gaussian Dynamical Model

As a first example, a simple linear and Gaussian state space model is considered, i.e., for  $n = 1, \dots, T$ :

$$\begin{aligned} f_n(x_n|x_{n-1}) &= \mathcal{N}(x_n; Hx_{n-1}, \Sigma_x), \\ g_n(y_n|x_n) &= \mathcal{N}(y_n; Gx_n, \Sigma_y). \end{aligned} \quad (2.40)$$

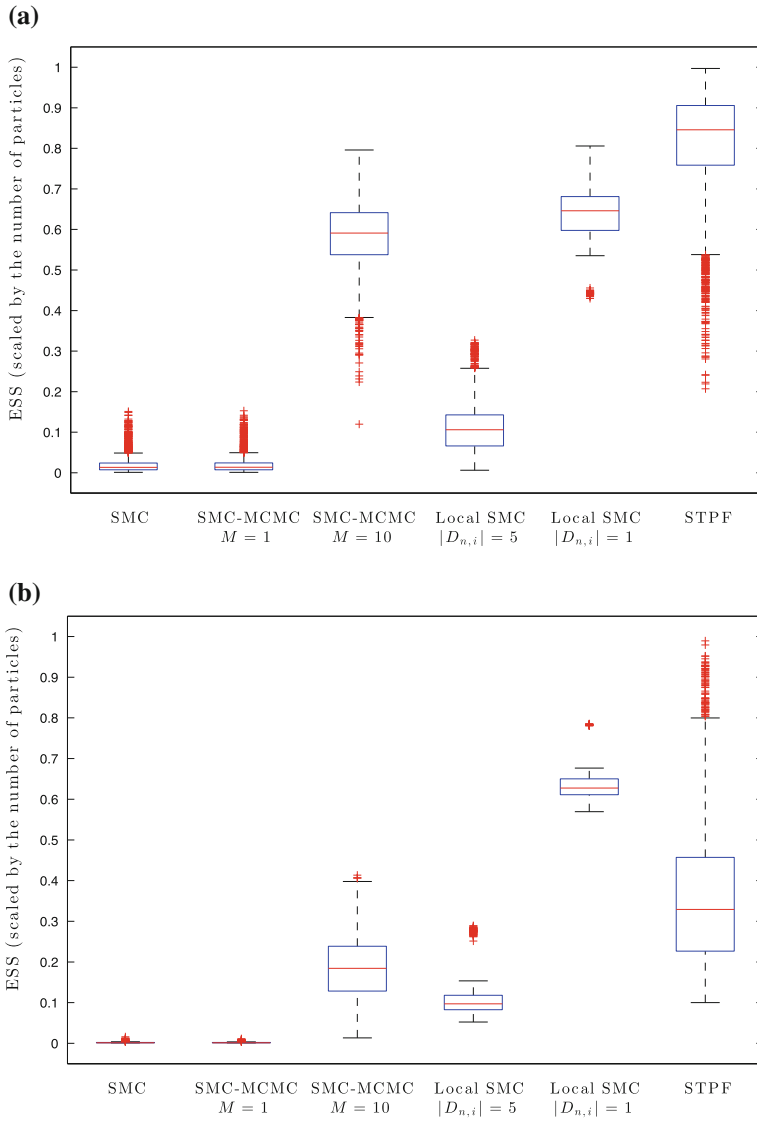
Such a model is interesting for the understanding and the study of approximation methods since the posterior distribution can be derived analytically via the use of the Kalman filter [26]. In our simulation results, the matrix  $H$  of size  $d \times d$  has been obtained by randomly and independently selecting for each row two column indexes for which the value is set to 0.495. The covariance matrices are defined as  $\Sigma_x = I_d$  and  $\Sigma_y = I_{d_y}$ . For a fair comparison between the different algorithms, we decide to set their parameter as shown in Table 2.1 in order to have an equivalent computational cost for all algorithms.

The performances are studied with a scenario in which all the  $d$ -dimensions of the hidden state are observed at each time step, i.e.,  $d_y = d$  and  $G = I_d$ . From this model, owing to the diagonality of  $G$  and the covariance matrix both in the prior and the likelihood distribution, it is obvious that their product can be factorized as in Eq. (2.19) with  $|D_{n,j}| = 1, \forall n, j$ . As a consequence, we define the partitioning of the subset in the STPF such that  $\forall n, j: |A_{n,j} \setminus A_{n,i-1}| = 1$ . In Fig. 2.5, the ESS scaled by the number of particles obtained for the different SMC-based algorithms is depicted. The standard SMC algorithm performs very badly even when the dimension is 10 and completely collapses when  $d = 50$ . The same remarks hold when the SMC-MCMC is used with only  $M = 1$  which corresponds to the resample-move algorithm. However, we can observe that the use of a sequence of bridging densities that gradually introduces the effect of the likelihood distribution (SMC-MCMC with  $M = 10$ ), improves the effective sample size remarkably. The effective sample size of the local SMC filter corresponds to the average of the ESS obtained from the  $d/B_n$  SMC filters used in each  $B_n$  subsets. As a consequence, the ESS depends quite obviously on the cardinality of each subset. Finally, the performance of the STPF in terms of ESS is deteriorating when the dimension increases due to the path degeneracy effect that we have discussed previously. However, this ESS is obtained

**Table 2.1** Value of the different parameters of Monte-Carlo algorithms used in the simulation

|            | SMC                | SMC-MCMC            |     | Local SMC          | STPF                |     | SMCMC               |      |
|------------|--------------------|---------------------|-----|--------------------|---------------------|-----|---------------------|------|
| Complexity | $\mathcal{O}(nNd)$ | $\mathcal{O}(nNMd)$ |     | $\mathcal{O}(nNd)$ | $\mathcal{O}(nNMd)$ |     | $\mathcal{O}(nLNd)$ |      |
| Parameter  | $N$                | $N$                 | $M$ | $N$                | $N$                 | $M$ | $L$                 | $N$  |
| Value      | 1000               | 100                 | 10  | 1000               | 10                  | 100 | 1                   | 1000 |

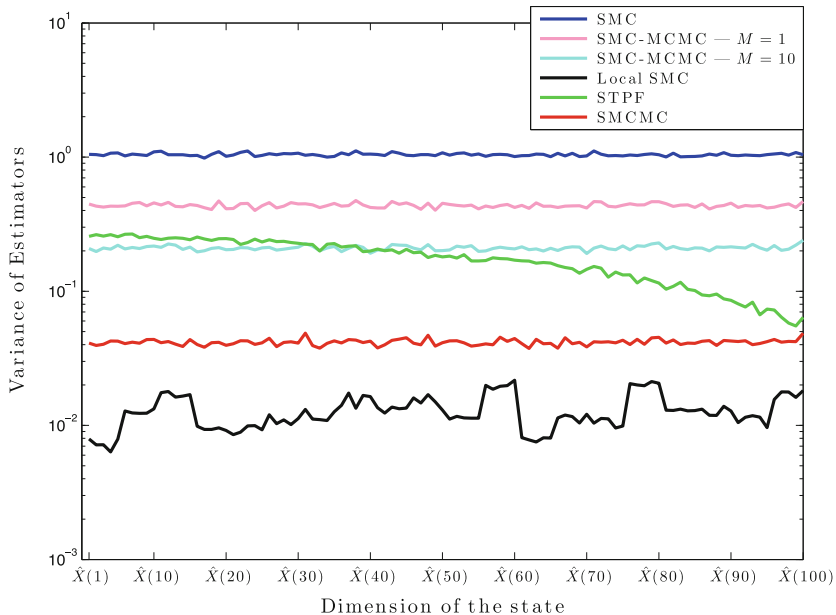




**Fig. 2.5** Effective sample size, scaled by the number of particles, obtained with the different algorithms for the linear and Gaussian state-space model at the different time steps using 100 runs. **a**  $d = 10$ . **b**  $d = 50$

using the global weights (line 21 of Algorithm4) which thus corresponds to the number of local particle systems that contributes to the final estimator.

Figure 2.6 shows the variance for the estimators of the  $d$ -dimensional latent states,  $X(1), \dots, X(d)$  (with  $d = 100$ ), averaged over time and obtained using 100 runs.



**Fig. 2.6** Variance for estimators of  $X(1), \dots, X(d)$  (with  $d = 100$ ) averaged over time and obtained using 100 runs. The MCMC kernel used in both SMC-MCMC and SMCMC partition the space with subsets of dimension 5. The dimension of each subset in the local SMC is also 5

We can clearly see the path degeneracy problem in the STPF which leads to an higher variance for  $\hat{X}(1)$  compared to  $\hat{X}(d)$ . The variance of both SMC-MCMC and SMCMC are quite stable across dimension of the state. The local SMC outperforms, in terms of variance, the other techniques but suffers from a spatially inhomogeneous approximation of the posterior distribution as we can see from unstable variance over space.

Table 2.2 summarizes the bias and the variance for the estimator of the posterior mean for all the algorithms with different parameter configuration. As expected, the performances of the classical SMC algorithm deteriorates quite significantly as  $d$  increases. The introduction of the sequence of bridging densities (SMC-MCMC) clearly improves the performances of the algorithm. Moreover, the use of smaller dimension on each subset ( $P = d$  vs.  $P = d/5$ ) for the Metropolis–Hastings within Gibbs sampler used within the SMC-MCMC leads in that example to better performance. The STPF performs quite well compared to both SMC and SMC-MCMC, but as previously illustrated, could be subject to path degeneracy effects as  $d$  increases. The Local SMC filter with block of dimension 1 (i.e.,  $B_n = d$ ) gives the smallest variance, but at the expense of a nonnegligible bias, due to the approximation of the posterior as a product of marginals on each block—Eq. (2.20). As an example, the bias obtained with this technique is higher than the one from the classical SMC when  $d = 10$ . Finally, the SMCMC algorithm that uses an MCMC kernel with  $P = d$  gives the smallest bias and reasonable variance. Let us remark that both the bias and the

**Table 2.2** Statistical properties of the estimator of the posterior mean—time and space average of the absolute value of the bias and the variance across 100 MC runs

| Algorithm |             |           | $d = 10$      |               | $d = 50$      |               | $d = 100$     |               |
|-----------|-------------|-----------|---------------|---------------|---------------|---------------|---------------|---------------|
|           |             |           | Bias          | Var           | Bias          | Var           | Bias          | Var           |
| SMC       |             |           | 0.0481        | 0.0657        | 0.4460        | 0.7689        | 0.5536        | 1.0430        |
| SMC-MCMC  | $M = 1$     | $P = d/5$ | 0.0426        | 0.0507        | 0.3362        | 0.3703        | 0.4062        | 0.4369        |
|           |             | $P = d$   | 0.0267        | 0.0206        | 0.2324        | 0.1885        | 0.2841        | 0.2366        |
|           | $M = 10$    | $P = d/5$ | 0.0282        | 0.0368        | 0.1225        | 0.1452        | 0.1660        | 0.2110        |
|           |             | $P = d$   | 0.0141        | 0.0139        | 0.0602        | 0.0610        | 0.0929        | 0.1009        |
| Local SMC | $B_n = d/5$ |           | 0.0541        | 0.0101        | 0.0594        | 0.0133        | 0.0620        | 0.0129        |
|           | $B_n = d$   |           | 0.0759        | <b>0.0009</b> | 0.0584        | <b>0.0012</b> | 0.0603        | <b>0.0012</b> |
| STPF      |             |           | 0.0095        | 0.0114        | 0.0502        | 0.1017        | 0.0730        | 0.1778        |
| SMCMC     | $P = d/5$   |           | 0.0074        | 0.0087        | 0.0211        | 0.0235        | 0.0416        | 0.0419        |
|           | $P = d$     |           | <b>0.0038</b> | 0.0026        | <b>0.0162</b> | 0.0169        | <b>0.0388</b> | 0.0366        |

variance tends theoretically to zero asymptotically with the number of particles for all the methods, except for the local SMC filter.

### 2.6.2 Two-Dimensional Graph Model

In this section, we consider a two-dimensional graph that has been used in both [3, 36] to assess the performances of the different algorithms. Let the components of state  $x_n$  be indexed by vertices  $v \in V$ , where  $V = \{1, \dots, \sqrt{d}\}^2$ . The dimension of the model is thus  $d$ . At time step  $n$ , the prior distribution at vertex  $v$  follows the following mixture distribution:

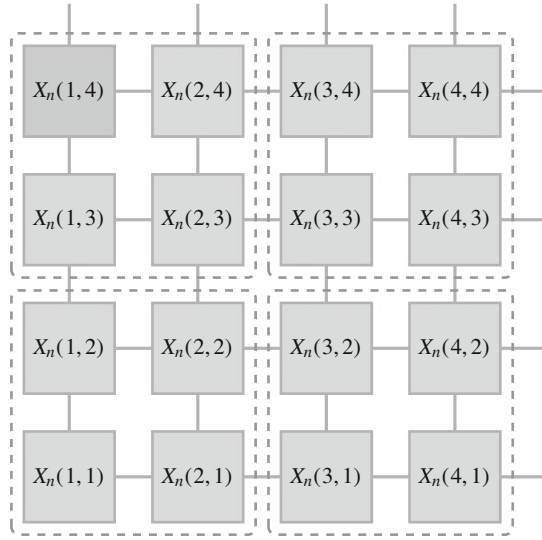
$$f(x_n(v)|x_{n-1}) = \sum_{u \in N(v)} w_u(v) f_u(x_n(v)|x_{n-1}(u)), \quad (2.41)$$

where  $N(v) = \{u : D(u, v) \leq r\}$  corresponds to the neighborhood of vertex  $v$  with  $r \geq 1$  and  $D(u, v) = \sqrt{(a-c)^2 + (b-d)^2}$  the Euclidean distance between the two vertices  $v = (a, b)$  and  $u = (c, d)$ . The observations are

$$Y_n(v) = X_n(v) + \eta_n(v), \quad (2.42)$$

for  $v \in V$  where  $\eta_n(v)$  are i.i.d.  $t$ -distributed random variables with degree of freedom  $\nu$ . In the simulation experiments, a Gaussian mixture is used with component mean  $X_{n-1}(u)$  and unity variance. The mixture weights are set to be  $w_u(v) \propto 1/(D(u, v) + \delta)$  such that  $\sum_{u \in N(v)} w_u(v) = 1$ . Finally, the data has been generated by using  $r = 1$ ,  $\delta = 1$  and  $\nu = 10$ .

**Fig. 2.7** Illustration of the block partitioning of the state for the local SMC in the two-dimensional graph example



**Table 2.3** Mean squared error of the posterior mean averaged over time, space and 100 runs with  $d = 144$

| SMC  | SMC-MCMC  |         |           |         | Local SMC   |           |      | SMCMC     |            |
|------|-----------|---------|-----------|---------|-------------|-----------|------|-----------|------------|
|      | $M = 1$   |         | $M = 10$  |         | $B_n = d/9$ | $B_n = d$ | STPF | $P = d/9$ | $P = d$    |
|      | $P = d/9$ | $P = d$ | $P = d/9$ | $P = d$ |             |           |      |           |            |
| 3154 | 2266      | 1392    | 1154      | 261     | 617         | 152       | 174  | 344       | <b>151</b> |

From this model description, it is straightforward to see that the product of the likelihood and the prior can be factorized as in Eq. (2.19) with  $|D_{n,j}| = 1, \forall n, j$ . As a consequence, we define the partitioning of the subset in the STPF such that  $\forall n, j: |A_{n,j}| = 1$ . For the local SMC filter, the space is partitioned such that each block is itself a square as illustrated in Fig. 2.7. The same configuration concerning the number of particles shown in Table 2.1 is used in this example. Table 2.3 shows the mean squared error for the posterior mean obtained using all the Monte-Carlo algorithms under different settings. The SMCMC ( $P = d$ ) and the local SMC ( $B_n = d$ ) give similar performance and outperform slightly the STPF and more significantly the other algorithms. Once again, the introduction of the sequence of bridging densities within the SMC-MCMC clearly improves the performance of the estimators.

## 2.7 Conclusion

In this chapter, after describing the generic framework of traditional SMC methods for the optimal filtering problem in a general HMM, we discuss their limitation when applied to high-dimensional systems. We thus provide an overview of recent Monte-Carlo-based approaches that have been proposed in order to improve the performance of such approaches in high-dimensional systems. Through two examples, we have shown empirically that the use of these recent developments could lead to a significant improvement. It is however difficult to state that in general case one technique would be better than another. Indeed, the choice will be clearly dependent on the model which is under study and on possible constraints like computing resources available, storage capacity, or desired level of accuracy. A more detailed analysis of all these algorithms with a finite number of samples will be clearly interesting for comparison purpose and could be used to design some automatic strategy to select the “optimal” algorithm and its parameters given the model and the constraints.

## References

1. Anderson, T.W.: The Statistical Analysis of Time Series, vol. 19. Wiley, New York (2011)
2. Berzuini, C., Best, N.G., Gilks, W.R., Larizza, C.: Dynamic conditional independence models and Markov chain Monte Carlo methods. *J. Am. Stat. Assoc.* **92**(440), 1403–1412 (1997)
3. Beskos, A., Crisan, D., Jasra, A., Kamatani, K., Zhou, Y.: A Stable Particle Filter in High-Dimensions (2014). [arXiv.org](https://arxiv.org/abs/1406.0098)
4. Bickel, P., Li, B., Bengtsson, T.: Sharp failure rates for the bootstrap particle filter in high dimensions. *Insti. Math. Stat. Collect.* **3**, 318–329 (2008)
5. Brockwell, A., Del Moral, P., Doucet, A.: Sequentially interacting Markov chain Monte Carlo methods. *Ann. Statist.* **38**(6), 3387–3411 (2010)
6. Cappé, O., Godsill, S., Moulines, E.: An overview of existing methods and recent advances in sequential Monte Carlo. *Proc. IEEE* **95**(5), 899–924 (2007)
7. Cappe, O., Moulines, E., Ryden, T.: *Inference in Hidden Markov Models* (2005)
8. Chopin, N.: Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Ann. Stat.* 2385–2411 (2004)
9. Crisan, D., Doucet, A.: A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Sig. Process.* **50**(3), 736–746 (2002)
10. Del Moral, P.: *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, New York (2004)
11. Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. *J. Roy. Stat. Soc. Ser. B (Stat. Methodol.)* **68**(3), 411–436 (2006)
12. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. *IEEE Conf. Comput. Vis. Patt. Recogn.* **2000**, 126–133 (2000)
13. Djuric, P., Bugallo, M.F.: Particle filtering for high-dimensional systems. In: *IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing Adaptive Processing (CAMSAP)*, pp. 352–355 (2013)
14. Djuric, P., Lu, T., Bugallo, M.F.: Multiple particle filtering. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007, ICASSP (2007)*
15. Doucet, A., De Freitas, N., Gordon, N. (eds.): *Sequential Monte Carlo Methods in Practice*. Springer, New York (2001)

16. Doucet, A., Godsill, S., Andrieu, C.: On sequential Monte-Carlo sampling methods for Bayesian filtering. *Stat. Comput.* **10**, 197–208 (2000)
17. Gall, J., Potthoff, J., Schnörr, C., Rosenhahn, B., Seidel, H.P.: Interacting and annealing particle filters: mathematics and a recipe for applications. *J. Math. Imaging Vis.* **28**(1), 1–18 (2014)
18. Geweke, J., et al.: Evaluating the Accuracy of Sampling-based Approaches to the Calculation of Posterior Moments, vol. 196. Federal Reserve Bank of Minneapolis, Research Department (1991)
19. Geyer, C.J.: Markov chain Monte Carlo maximum likelihood. In: *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pp. 156–163 (1991)
20. Gilks, W.R., Berzuini, C.: Following a moving target-Monte Carlo inference for dynamic Bayesian models. *J. Roy. Stat. Soc. Ser. B (Stat. Methodol.)* **63**, 127–146 (2001)
21. Godsill, S.J., Clapp, T.: Improvement strategies for Monte Carlo particle filters. In: Doucet, A., De Freitas, N., Gordon, N. (eds.) *Sequential Monte Carlo Methods in Practice*. Springer, Berlin (2001)
22. Golightly, A., Wilkinson, D.: Bayesian sequential inference for nonlinear multivariate diffusions. *Stat. Comput.* **16**(4), 323–338 (2006)
23. Gordon, N., Salmond, D., Smith, A.F.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proc. F. Radar. Sig. Process.* **140**, 107–113 (1993)
24. Jones, G.L., et al.: On the Markov chain central limit theorem. *Probab. Surv.* **1**, 299–320 (2004)
25. Julier, S.J., Uhlmann, J.K.: Unscented filtering and nonlinear estimation. In: *Proceedings of the IEEE*, pp. 401–422 (2004)
26. Kalman, R.E.: A new approach to linear filtering and prediction problems. *Trans. ASME J. Basic Eng.* **82**, 35–45 (1960)
27. Khan, Z., Balch, T., Dellaert, F.: MCMC-Based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(11), 1805–1819 (2005)
28. Kong, A., Liu, J.S., Wong, W.H.: Sequential imputations and bayesian missing data problems. *J. Am. Stat. Assoc.* **89**(425), 278–288 (1994)
29. Künsch, H.R.: Recursive Monte Carlo filters: algorithms and theoretical analysis. *Ann. Stat.* 1983–2021 (2005)
30. Leith, C.: The standard error of time-average estimates of climatic means. *J. Appl. Meteorol.* **12**(6), 1066–1069 (1973)
31. Liang, F., Wong, W.H.: Evolutionary monte carlo: applications to  $C_p$  Model sampling and change point problem. *Stat. Sinica* **10**, 317–342 (2000)
32. Mihaylova, L., Hegyi, A., Gning, A., Boel, R.K.: Parallelized particle and gaussian sum particle filters for large-scale freeway traffic systems. *IEEE Trans. Intell. Transp. Syst.* **13**(1), 36–48 (2012)
33. Neal, R.: Annealed importance sampling. *Stat. Comput.* 125–139 (2001)
34. Peters, G.W.: Topics in Sequential Monte Carlo Samplers. Master's thesis, University of Cambridge (2005)
35. Rebeschini, P.: Nonlinear Filtering in High Dimension. Ph.D. thesis, Princeton University (2014)
36. Rebeschini, P., van Handel, R.: Can Local Particle Filters Beat the Curse of Dimensionality? [arXiv.org](https://arxiv.org/abs/1301.3557) (2013)
37. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman filter: particle filters for tracking applications. Artech House (2004)
38. Robert, C.P., Casella, G.: *Monte Carlo Statistical Methods*. Springer, Berlin (2004)
39. Septier, F., Carmi, A., Pang, S., Godsill, S.: Multiple object tracking using evolutionary and hybrid MCMC-based particle algorithms. In: *15th IFAC Symposium on System Identification, (SYSID 2009)*, Saint Malo, France (2009)
40. Septier, F., Pang, S., Carmi, A., Godsill, S.: On MCMC-Based particle methods for bayesian filtering : application to multitarget tracking. In: *International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2009)*. Aruba, Dutch Antilles (2009)

41. Snyder, C.: Particle filters, the“optimal” proposal and high-dimensional systems. In: ECMWF Seminar on Data Assimilation for Atmosphere and Ocean, pp. 1–10 (2011)
42. Snyder, C., Bengtsson, T., Bickel, P., Anderson, J.: Obstacles to high-dimensional particle filtering. *Monthly Weather Rev., Spec. Collect.: Math. Adv. Data Assimilation* **136**(12), 4629–4640 (2008)
43. Vergé, C., Dubarry, C., Del Moral, P., Moulines, E.: On Parallel Implementation of Sequential Monte Carlo Methods: The Island Particle Model. [arXiv.org](https://arxiv.org/abs/1301.3764) (2013)

Theoretical Aspects of Spatial-Temporal Modeling

Peters, G.W.; Matsui, T. (Eds.)

2015, XV, 124 p. 18 illus., 13 illus. in color., Softcover

ISBN: 978-4-431-55335-9