

Technical report

Nested Sequential Monte Carlo Methods

Christian A. Naesseth, Fredrik Lindsten and Thomas B. Schön

- **Please cite this version:**

Christian A. Naesseth, Fredrik Lindsten and Thomas B. Schön. Nested Sequential Monte Carlo Methods. In *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015. JMLR: W&CP volume 37.

Abstract

We propose *nested sequential Monte Carlo* (NSMC), a methodology to sample from sequences of probability distributions, even where the random variables are high-dimensional. NSMC generalises the SMC framework by requiring only approximate, *properly weighted*, samples from the SMC proposal distribution, while still resulting in a correct SMC algorithm. Furthermore, NSMC can in itself be used to produce such properly weighted samples. Consequently, one NSMC sampler can be used to construct an efficient high-dimensional proposal distribution for another NSMC sampler, and this *nesting* of the algorithm can be done to an arbitrary degree. This allows us to consider complex and high-dimensional models using SMC. We show results that motivate the efficacy of our approach on several filtering problems with dimensions in the order of 100 to 1 000.

Keywords: high-dimensional inference, high-dimensional particle filter, exact approximation, optimal proposal, sequential Monte Carlo, importance sampling, spatio-temporal models

Nested Sequential Monte Carlo Methods

Christian A. Naesseth

Linköping University, Linköping, Sweden

CHRISTIAN.A.NAESSETH@LIU.SE

Fredrik Lindsten

The University of Cambridge, Cambridge, United Kingdom

FREDRIK.LINDSTEN@ENG.CAM.AC.UK

Thomas B. Schön

Uppsala University, Uppsala, Sweden

THOMAS.SCHON@IT.UU.SE

Abstract

We propose *nested sequential Monte Carlo* (NSMC), a methodology to sample from sequences of probability distributions, even where the random variables are high-dimensional. NSMC generalises the SMC framework by requiring only approximate, *properly weighted*, samples from the SMC proposal distribution, while still resulting in a correct SMC algorithm. Furthermore, NSMC can in itself be used to produce such properly weighted samples. Consequently, one NSMC sampler can be used to construct an efficient high-dimensional proposal distribution for another NSMC sampler, and this *nesting* of the algorithm can be done to an arbitrary degree. This allows us to consider complex and high-dimensional models using SMC. We show results that motivate the efficacy of our approach on several filtering problems with dimensions in the order of 100 to 1 000.

Keywords: high-dimensional inference, high-dimensional particle filter, exact approximation, optimal proposal, sequential Monte Carlo, importance sampling, spatio-temporal models

1. Introduction

Inference in complex and high-dimensional statistical models is a very challenging problem that is ubiquitous in applications. Examples include, but are definitely not limited to, climate informatics (Monteleoni et al., 2013), bioinformatics (Cohen, 2004) and machine learning (Wainwright and Jordan, 2008). In particular, we are interested in *sequential* Bayesian inference, which involves computing integrals of the form

$$\bar{\pi}_k(f) := \mathbb{E}_{\bar{\pi}_k}[f(X_{1:k})] = \int f(x_{1:k}) \bar{\pi}_k(x_{1:k}) dx_{1:k}, \quad (1)$$

for some sequence of probability densities

$$\bar{\pi}_k(x_{1:k}) = Z_{\pi_k}^{-1} \pi_k(x_{1:k}), \quad k \geq 1, \quad (2)$$

with normalisation constants $Z_{\pi_k} = \int \pi_k(x_{1:k}) dx_{1:k}$. Note that $x_{1:k} := (x_1, \dots, x_k) \in \mathcal{X}_k$. The typical scenario that we consider is the well-known problem of inference in time series or state space models (Shumway and Stoffer, 2011; Cappé et al., 2005). Here the index k corresponds to time and we want to process some *observations* $y_{1:k}$ in a sequential manner to compute expectations with respect to the filtering distribution $\bar{\pi}_k(dx_k) = \mathbb{P}(X_k \in dx_k | y_{1:k})$. To be specific, we are interested in settings where

- (i) X_k is high-dimensional, i.e. $X_k \in \mathbb{R}^d$ with $d \gg 1$, and

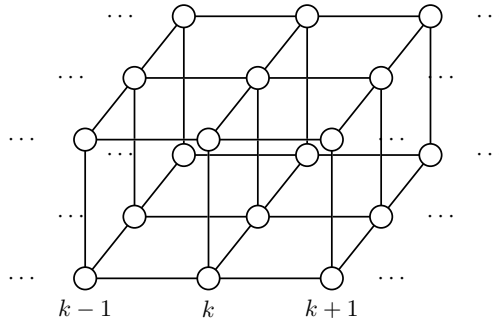


Figure 1: Example of a spatio-temporal model where $\bar{\pi}_k(x_{1:k})$ is described by a $k \times 2 \times 3$ undirected graphical model and $x_k \in \mathbb{R}^{2 \times 3}$.

- (ii) there are *local dependencies* among the latent variables $X_{1:k}$, both w.r.t. time k and between the individual components of the (high-dimensional) vectors X_k .

One example of the type of models we consider are the so-called spatio-temporal models (Wikle, 2015; Cressie and Wikle, 2011; Rue and Held, 2005). In Figure 1 we provide a probabilistic graphical model representation of a spatio-temporal model that we will explore further in Section 6.

Sequential Monte Carlo (SMC) methods, reviewed in Section 2.1, comprise one of the most successful methodologies for sequential Bayesian inference. However, SMC struggles in high-dimensions and these methods are rarely used for dimensions, say, $d \geq 10$ (Rebeschini and van Handel, 2015). The purpose of the NSMC methodology is to push this limit well beyond $d = 10$.

The basic strategy, described in Section 2.2, is to mimic the behaviour of a so-called *fully adapted* SMC algorithm. Full adaptation can drastically improve the efficiency of SMC in high dimensions. Unfortunately, it can rarely be implemented in practice since the fully adapted proposal distributions are typically intractable. NSMC addresses this difficulty by requiring only approximate, *properly weighted*, samples from the proposal distribution. The proper weighting condition ensures the validity of NSMC, thus providing a generalisation of the family of SMC methods. Furthermore, NSMC will itself produce properly weighted samples. Consequently, it is possible to use one NSMC procedure within another to construct efficient high-dimensional proposal distributions. This *nesting* of the algorithm can be done to an arbitrary degree. For instance, for the model depicted in Figure 1 we could use three nested samplers, one for each dimension of the “volume”.

The main methodological development is concentrated to Sections 3–4. We introduce the concept of proper weighting, approximations of the proposal distribution, and nesting of Monte Carlo algorithms. Throughout Section 3 we consider simple importance sampling and in Section 4 we extend the development to the sequential setting.

We deliberately defer the discussion of the existing body of related work until Section 5, to open up for a better understanding of the relationships to the new developments presented in Sections 3–4. We also discuss various attractive features of NSMC that are of interest in high-dimensional settings, e.g. the fact that it is easy to distribute the computation, which results in improved memory efficiency and lower communication costs. Section 6 profiles our method extensively with a state-of-the-art competing algorithm on several high-dimensional data sets. We also show the performance

of inference and the modularity of the method on a $d = 1\,056$ dimensional climatological spatio-temporal model (Fu et al., 2012) structured according to Figure 1. Finally, in Section 7 we conclude the paper with some final remarks.

2. Background and Inference Strategy

2.1 Sequential Monte Carlo

Evaluating $\bar{\pi}_k(f)$ as well as the normalisation constant Z_{π_k} in (2) is typically intractable and we need to resort to approximations. SMC methods, or particle filters (PF), constitute a popular class of numerical approximations for sequential inference problems. Here we give a high-level introduction to the concepts underlying SMC methods, and postpone the details to Section 4. For a more extensive treatment we refer to Doucet and Johansen (2011); Cappé et al. (2005); Doucet et al. (2001). In particular, we will use the auxiliary SMC method as proposed by Pitt and Shephard (1999).

At iteration $k - 1$, the SMC sampler approximates the target distribution $\bar{\pi}_{k-1}$ by a collection of weighted particles (samples) $\{(X_{1:k-1}^i, W_{k-1}^i)\}_{i=1}^N$. These samples define an empirical point-mass approximation of the target distribution

$$\bar{\pi}_{k-1}^N(dx_{1:k-1}) := \sum_{i=1}^N \frac{W_{k-1}^i}{\sum_{\ell} W_{k-1}^{\ell}} \delta_{X_{1:k-1}^i}(dx_{1:k-1}), \quad (3)$$

where $\delta_X(dx)$ denotes a Dirac measure at X . Each iteration of the SMC algorithm can then conceptually be described by three steps, resampling, propagation, and weighting.

The resampling step puts emphasis on the most promising particles by discarding the unlikely ones and duplicating the likely ones. The propagation and weighting steps essentially correspond to using importance sampling when changing the target distribution from $\bar{\pi}_{k-1}$ to $\bar{\pi}_k$, i.e. simulating new particles from a *proposal distribution* and then computing corresponding importance weights.

2.2 Adapting the Proposal Distribution

The first working SMC algorithm was the bootstrap PF by Gordon et al. (1993), which propagates particles by sampling from the system dynamics and computes importance weights according to the observation likelihood (in the state space setting). However, it is well known that the bootstrap PF suffers from weight collapse in high-dimensional settings (Bickel et al., 2008), i.e. the estimate is dominated by a single particle with weight close to one. This is an effect of the mismatch between the importance sampling proposal and the target distribution, which typically gets more pronounced in high dimensions.

More efficient proposals, partially alleviating the degeneracy issue for some models, can be designed by *adapting* the proposal distribution to the target distribution (see Section 4.2). In Naesseth et al. (2014a) we make use of the *fully adapted* SMC method (Pitt and Shephard, 1999) for doing inference in a (fairly) high-dimensional *discrete* model where x_k is a 60-dimensional discrete vector. We can then make use of forward filtering and backward simulation, operating on the individual *components* of each x_k , in order to sample from the fully adapted SMC proposals. However, this method is limited to models where the latent space is either discrete or Gaussian and the optimal proposal can be identified with a tree-structured graphical model. Our development here can be

seen as a non-trivial extension of this technique. Instead of coupling one SMC sampler with an *exact* forward filter/backward simulator (which in fact reduces to an instance of standard SMC), we derive a way of coupling multiple SMC samplers and SMC-based backward simulators. This allows us to construct procedures for mimicking the efficient fully adapted proposals for arbitrary latent spaces and structures in high-dimensional models.

3. Proper Weighting and Nested Importance Sampling

In this section we will lay the groundwork for the derivation of the class of NSMC algorithms. We start by considering the simpler case of importance sampling (IS), which is a fundamental component of SMC, and introduce the key concepts that we make use of. In particular, we will use a (slightly nonstandard) presentation of an algorithm as an instance of a *class*, in the object-oriented sense, and show that these classes can be nested to an arbitrary degree.

3.1 Exact Approximation of the Proposal Distribution

Let $\bar{\pi}(x) = Z_{\bar{\pi}}^{-1}\pi(x)$ be a target distribution of interest. IS can be used to estimate an expectation $\bar{\pi}(f) := \mathbb{E}_{\bar{\pi}}[f(X)]$ by sampling from a proposal distribution $\bar{q}(x) = Z_{\bar{q}}^{-1}q(x)$ and computing the estimator $(\sum_{i=1}^N W^i)^{-1} \sum_{i=1}^N W^i f(X^i)$, with $W^i = \frac{Z_{\bar{q}}\pi(X^i)}{q(X^i)}$, and where $\{(X^i, W^i)\}_{i=1}^N$ are the weighted samples. It is possible to replace the IS weight by a nonnegative unbiased estimate, and still obtain a valid (consistent, etc.) algorithm (Liu, 2001, p. 37). One way to motivate this approach is by considering the random weight to be an auxiliary variable and to extend the target distribution accordingly. Our development is in the same flavour, but we will use a more explicit condition on the relationship between the random weights and the simulated particles. Specifically, we will make use of the following key property to formally justify the proposed algorithms.

Definition 1 (Properly weighted sample). *A (random) pair (X, W) is properly weighted for an unnormalised distribution p if $W \geq 0$ and $\mathbb{E}[f(X)W] = p(f) := \int f(x)p(x)dx$ for all measurable functions f .*

Note that proper weighting of $\{(X^i, W^i)\}_{i=1}^N$ implies unbiasedness of the estimate of the normalising constant of p . Indeed, taking $f(x) \equiv 1$ gives $\mathbb{E} \left[\frac{1}{N} \sum_{i=1}^N W^i \right] = \int p(x)dx =: Z_p$.

Interestingly, to construct a valid IS algorithm for our target $\bar{\pi}$ it is sufficient to generate samples that are properly weighted *w.r.t. the proposal* distribution q . To formalise this claim, assume that we are not able to simulate exactly from \bar{q} , but that it is possible to evaluate the unnormalised density q point-wise. Furthermore, assume we have access to a class Q , which works as follows. The constructor of Q requires the specification of an *unnormalised density function*, say, q , which will be approximated by the procedures of Q . Furthermore, to highlight the fact that we will typically use IS (and SMC) to construct Q , the constructor also takes as an argument a precision parameter M , corresponding to the number of samples used by the “internal” Monte Carlo procedure. An object is then instantiated as $q = Q(q, M)$. The class Q is assumed to have the following properties:

(A1) Let $q = Q(q, M)$. Assume that:

1. The construction of q results in the generation of a (possibly random) member variable, accessible as $\hat{Z}_q = q.\text{GetZ}()$. The variable \hat{Z}_q is a nonnegative, unbiased estimate of the normalising constant $Z_q = \int q(x)dx$.

2. Q has a member function `Simulate` which returns a (possibly random) variable $X = q.\text{Simulate}()$, such that (X, \widehat{Z}_q) is properly weighted for q .

With the definition of Q in place, it is possible to generalise¹ the basic importance sampler as in Algorithm 1, which generates weighted samples $\{(X^i, W^i)\}_{i=1}^N$ targeting $\bar{\pi}$. Note that Algorithm 1 is different from a random weight IS, since it approximates the proposal distribution (and not just the importance weights).

Algorithm 1 Nested IS (steps 1–3 for $i = 1, \dots, N$)

1. Initialise $q^i = Q(q, M)$.
 2. Set $\widehat{Z}_q^i = q^i.\text{GetZ}()$ and $X^i = q^i.\text{Simulate}()$.
 3. Set $W^i = \frac{\widehat{Z}_q^i \pi(X^i)}{q(X^i)}$.
 4. Compute $\widehat{Z}_\pi = \frac{1}{N} \sum_{i=1}^N W^i$.
-

To see the validity of Algorithm 1 we can interpret the sampler as a standard IS algorithm for an extended target distribution, defined as $\bar{\Pi}(x, u) := u \bar{Q}(x, u) \bar{\pi}(x) q^{-1}(x)$, where $\bar{Q}(x, u)$ is the joint PDF of the random pair $(q.\text{Simulate}(), q.\text{GetZ}())$. Note that $\bar{\Pi}$ is indeed a PDF that admits $\bar{\pi}$ as a marginal; for any measurable subset $A \subseteq X$,

$$\bar{\Pi}(A \times \mathbb{R}_+) = \int \mathbb{1}_A(x) \frac{u \bar{\pi}(x)}{q(x)} \bar{Q}(x, u) dx du = \mathbb{E} \left[\widehat{Z}_q \frac{\mathbb{1}_A(X) \bar{\pi}(X)}{q(X)} \right] = \bar{q} \left(\mathbb{1}_A \frac{\bar{\pi}}{q} \right) Z_q = \bar{\pi}(A),$$

where the penultimate equality follows from the fact that (X, \widehat{Z}_q) is properly weighted for q . Furthermore, the standard unnormalised IS weight for a sampler with target $\bar{\Pi}$ and proposal \bar{Q} is given by $u \pi/q$, in agreement with Algorithm 1.

Algorithm 1 is an example of what is referred to as an *exact approximation*; see e.g., Andrieu and Roberts (2009); Andrieu et al. (2010). Algorithmically, the method appears to be an approximation of an IS, but samples generated by the algorithm nevertheless target the correct distribution $\bar{\pi}$.

3.2 Modularity of Nested IS

To be able to implement Algorithm 1 we need to define a class Q with the required properties (A1). The modularity of the procedure (as well as its name) comes from the fact that we can use Algorithm 1 also in this respect. Indeed, let us now view $\bar{\pi}$ —the target distribution of Algorithm 1— as the *proposal distribution* for another Nested IS procedure and consider the following definition of Q :

1. Algorithm 1 is executed at the construction of the object $p = Q(\bar{\pi}, N)$, and $p.\text{GetZ}()$ returns the normalising constant estimate \widehat{Z}_π .
2. $p.\text{Simulate}()$ simulates a categorical random variable B with $\mathbb{P}(B = i) = W^i / \sum_{\ell=1}^N W^\ell$ and returns X^B .

1. With $q.\text{GetZ}() \mapsto Z$ and $q.\text{Simulate}()$ returning a sample from \bar{q} we obtain the standard IS method.

Now, for any measurable f we have,

$$\begin{aligned} \mathbb{E}[f(X^B)\widehat{Z}_\pi] &= \sum_{i=1}^N \mathbb{E} \left[f(X^i) \widehat{Z}_\pi \frac{W^i}{N \widehat{Z}_\pi} \right] = \frac{1}{N} \sum_{i=1}^N \mathbb{E} \left[f(X^i) \frac{\widehat{Z}_q^i \pi(X^i)}{q(X^i)} \right] \\ &= \bar{q} \left(\frac{f\pi}{q} \right) Z_q = \bar{\pi}(f) Z_\pi, \end{aligned}$$

where, again, we use the fact that (X^i, \widehat{Z}_q^i) is properly weighted for q . This implies that (X^B, \widehat{Z}_π) is properly weighted for π and that our definition of $Q(\pi, N)$ indeed satisfies condition (A1).

The Nested IS algorithm in itself is unlikely to be of direct practical interest. However, in the next section we will, essentially, repeat the preceding derivation in the context of SMC to develop the NSMC method.

4. Nested Sequential Monte Carlo

4.1 Fully Adapted SMC Samplers

Let us return to the sequential inference problem. As before, let $\bar{\pi}_k(x_{1:k}) = Z_{\pi_k}^{-1} \pi_k(x_{1:k})$ denote the target distribution at “time” k . The unnormalised density π_k can be evaluated point-wise, but the normalising constant Z_{π_k} is typically unknown. We will use SMC to simulate sequentially from the distributions $\{\bar{\pi}_k\}_{k=1}^n$. In particular, we consider the fully adapted SMC sampler (Pitt and Shephard, 1999), which corresponds to a specific choice of resampling weights and proposal distribution, chosen in such a way that the importance weights are all equal to $1/N$. Specifically, the proposal distribution (often referred to as the *optimal proposal*) is given by $\bar{q}_k(x_k | x_{1:k-1}) = Z_{q_k}(x_{1:k-1})^{-1} q_k(x_k | x_{1:k-1})$, where

$$q_k(x_k | x_{1:k-1}) := \frac{\pi_k(x_{1:k})}{\pi_{k-1}(x_{1:k-1})}.$$

In addition, the normalising “constant” $Z_{q_k}(x_{1:k-1}) = \int q_k(x_k | x_{1:k-1}) dx_k$ is further used to define the *resampling weights*, i.e. the particles at time $k-1$ are resampled according to $Z_{q_k}(x_{1:k-1})$ before they are propagated to time k . For notational simplicity, we use the convention $x_{1:0} = \emptyset$, $q_1(x_1 | x_{1:0}) = \pi_1(x_1)$ and $Z_{q_1}(x_{1:0}) = Z_{\pi_1}$. The fully adapted auxiliary SMC sampler is given in Algorithm 2.

As mentioned above, at each iteration $k = 1, \dots, n$, the method produces *unweighted* samples $\{X_k^i\}_{i=1}^N$ approximating $\bar{\pi}_k$. It also produces an unbiased estimate \widehat{Z}_{π_k} of Z_{π_k} (Del Moral, 2004, Proposition 7.4.1). The algorithm is expressed in a slightly non-standard form; at iteration k we loop over the ancestor particles, i.e. the particles after resampling at iteration $k-1$, and let each ancestor particle j generate m_k^j offsprings. (The variable L is just for bookkeeping.) This is done to clarify the connection with the NSMC procedure below. Furthermore, we have included a (completely superfluous) resampling step at iteration $k=1$, where the “dummy variables” $\{X_{1:0}^i\}_{i=1}^N$ are resampled according to the (all equal) weights $\{Z_{q_1}(X_{1:0}^i)\}_{i=1}^N = \{Z_{\pi_1}\}_{i=1}^N$. The analogue of this step is, however, used in the NSMC algorithm, where the initial normalising constant Z_{π_1} is *estimated*. We thus have to resample the corresponding initial particle systems accordingly.

Algorithm 2 SMC (fully adapted)

1. Set $\widehat{Z}_{\pi_0} = 1$.
 2. **for** $k = 1$ **to** n
 - (a) Compute $\widehat{Z}_{\pi_k} = \widehat{Z}_{\pi_{k-1}} \times \frac{1}{N} \sum_{j=1}^N Z_{q_k}(X_{1:k-1}^j)$.
 - (b) Draw $m_k^{1:N}$ from a multinomial distribution with probabilities $\frac{Z_{q_k}(X_{1:k-1}^j)}{\sum_{\ell=1}^N Z_{q_k}(X_{1:k-1}^\ell)}$, for $j = 1, \dots, N$.
 - (c) Set $L \leftarrow 0$
 - (d) **for** $j = 1$ **to** N
 - i. Draw $X_k^i \sim \bar{q}_k(\cdot | X_{1:k-1}^j)$ and let $X_{1:k}^i = (X_{1:k-1}^j, X_k^i)$ for $i = L+1, \dots, L+m_k^j$.
 - ii. Set $L \leftarrow L + m_k^j$.
-

4.2 Fully Adapted Nested SMC Samplers

In analogue with Section 3, assume now that we are not able to simulate exactly from \bar{q}_k , nor compute Z_{q_k} . Instead, we have access to a class \mathbf{Q} which satisfies condition (A1). The proposed NSMC method is then given by Algorithm 3.

Algorithm 3 Nested SMC (fully adapted)

1. Set $\widehat{Z}_{\pi_0} = 1$.
 2. **for** $k = 1$ **to** n
 - (a) Initialise $\mathbf{q}^j = \mathbf{Q}(q_k(\cdot | X_{1:k-1}^j), M)$ for $j = 1, \dots, N$.
 - (b) Set $\widehat{Z}_{q_k}^j = \mathbf{q}^j.\text{GetZ}()$ for $j = 1, \dots, N$.
 - (c) Compute $\widehat{Z}_{\pi_k} = \widehat{Z}_{\pi_{k-1}} \times \left\{ \frac{1}{N} \sum_{j=1}^N \widehat{Z}_{q_k}^j \right\}$.
 - (d) Draw $m_k^{1:N}$ from a multinomial distribution with probabilities $\frac{\widehat{Z}_{q_k}^j}{\sum_{\ell=1}^N \widehat{Z}_{q_k}^\ell}$ for $j = 1, \dots, N$.
 - (e) Set $L \leftarrow 0$
 - (f) **for** $j = 1$ **to** N
 - i. Compute $X_k^i = \mathbf{q}^j.\text{Simulate}()$ and let $X_{1:k}^i = (X_{1:k-1}^j, X_k^i)$ for $i = L+1, \dots, L+m_k^j$.
 - ii. **delete** \mathbf{q}^j .
 - iii. Set $L \leftarrow L + m_k^j$.
-

Algorithm 3 can be seen as an *exact approximation* of the fully adapted SMC sampler in Algorithm 2. (In Appendix A.1 we provide a formulation of NSMC with arbitrary proposals and resampling weights.) We replace the exact computation of Z_{d_k} and exact simulation from \bar{q}_k , by the approximate procedures available through Q. Despite this approximation, however, Algorithm 3 is a valid SMC method. This is formalised by the following theorem.

Theorem 2. *Assume that Q satisfies condition (A1). Then, under certain regularity conditions on the function $f : X_k \mapsto \mathbb{R}^d$ and for an asymptotic variance $\Sigma_k^M(f)$, both specified in Appendix A.1.2, we have*

$$N^{1/2} \left(\frac{1}{N} \sum_{i=1}^N f(X_{1:k}^i) - \bar{\pi}_k(f) \right) \xrightarrow{D} \mathcal{N}(0, \Sigma_k^M(f)),$$

where $\{X_{1:k}^i\}_{i=1}^M$ are generated by Algorithm 3 and \xrightarrow{D} denotes convergence in distribution.

Proof See Appendix A.1.2. ■

Remark 3. *The key point with Theorem 2 is that, under certain regularity conditions, the NSMC method converges at rate \sqrt{N} even for a fixed (and finite) value of the precision parameter M . The asymptotic variance $\Sigma_k^M(f)$, however, will depend on the accuracy and properties of the approximative procedures of Q. We leave it as future work to establish more informative results, relating the asymptotic variance of NSMC to that of the ideal, fully adapted SMC sampler.*

4.3 Backward Simulation and Modularity of NSMC

As previously mentioned, the NSMC procedure is modular in the sense that we can make use of Algorithm 3 also to define the class Q. Thus, we now view $\bar{\pi}_n$ as the *proposal distribution* that we wish to approximately sample from using NSMC. Algorithm 3 directly generates an estimate \hat{Z}_{π_n} of the normalising constant of π_n (which indeed is unbiased, see Theorem 6). However, we also need to generate a sample $\tilde{X}_{1:n}$ such that $(\tilde{X}_{1:n}, \hat{Z}_{\pi_n})$ is properly weighted for π_n .

The simplest approach, akin to the Nested IS procedure described in Section 3.2, is to draw B_n uniformly on $\{1, \dots, N\}$ and return $\tilde{X}_{1:n} = X_{1:n}^{B_n}$. This will indeed result in a valid definition of the Simulate procedure. However, this approach will suffer from the well known path degeneracy of SMC samplers. In particular, since we call $q^j.\text{Simulate}()$ multiple times in Step 2(f)i of Algorithm 3, we risk to obtain (very) strongly correlated samples by this simple approach.

It is possible to improve the performance of the above procedure by instead making use of a *backward simulator* (Godsill et al., 2004; Lindsten and Schön, 2013) to simulate $\tilde{X}_{1:n}$. The backward simulator, given in Algorithm 4, is a type of smoothing algorithm; it makes use of the particles generated by a forward pass of Algorithm 3 to simulate backward in “time” a trajectory $\tilde{X}_{1:n}$ approximately distributed according to $\bar{\pi}_n$.

Remark 4. *Algorithm 4 assumes unweighted particles and can thus be used in conjunction with the fully adapted NSMC procedure of Algorithm 2. If, however, the forward filter is not fully adapted the weights need to be accounted for in the backward simulation; see Appendix A.1.3.*

The modularity of NSMC is established by the following result.

Algorithm 4 Backward simulator (fully adapted)

1. Draw B_n uniformly on $\{1, \dots, N\}$.
 2. Set $\tilde{X}_n = X_n^{B_n}$.
 3. **for** $k = n - 1$ **to** 1
 - (a) Compute $\tilde{W}_k^j = \frac{\pi_n((X_{1:k}^j, \tilde{X}_{k+1:n}))}{\pi_k(X_{1:k}^j)}$ for $j = 1, \dots, N$.
 - (b) Draw B_k from a categorical distribution with probabilities $\frac{\tilde{W}_k^j}{\sum_{\ell=1}^N \tilde{W}_k^\ell}$ for $j = 1, \dots, N$.
 - (c) Set $\tilde{X}_{k:n} = (X_k^{B_k}, \tilde{X}_{k+1:n})$.
-

Definition 5. Let $\mathfrak{p} = \mathbb{Q}(\pi_n, N)$ be defined as follows:

1. The constructor executes Algorithm 3 with target distribution π_n and with N particles, and $\mathfrak{p}.\text{GetZ}()$ returns the estimate of the normalising constant \hat{Z}_{π_n} .
2. $\mathfrak{p}.\text{Simulate}()$ executes Algorithm 4 and returns $\tilde{X}_{1:n}$.

Theorem 6. The class \mathbb{Q} defined as in Definition 5 satisfies condition (A1).

Proof See Appendix A.1.3. ■

A direct, and important, consequence of Theorem 6 is that NSMC can be used as a component of powerful learning algorithms, such as the particle Markov chain Monte Carlo (PMCMC) method (Andrieu et al., 2010) and many of the other methods discussed in Section 5. Since standard SMC is a special case of NSMC, Theorem 6 implies proper weighting also of SMC.

5. Practicalities and Related Work

There has been much recent interest in using SMC within SMC in various ways. The SMC² by Chopin et al. (2013) and the recent method by Crisan and Míguez (2013) are sequential learning algorithms for state space models, where one SMC sampler for the parameters is coupled with another SMC sampler for the latent states. Johansen et al. (2012) and Chen et al. (2011) address the state inference problem by splitting the state variable into different components and run coupled SMC samplers for these components. These methods differ substantially from NSMC; they solve different problems and the “internal” SMC sampler(s) is constructed in a different way (for approximate marginalisation instead of for approximate simulation). Another related method is the random weights PF of Fearnhead et al. (2010a), requiring exact samples from \bar{q} and where the importance weights are estimated using a nested Monte Carlo algorithm.

The method most closely related to NSMC is the space-time particle filter (ST-PF) (Beskos et al., 2014a), which has been developed independently and in parallel with our work. The ST-PF is also designed for solving inference problems in high-dimensional models. It can be seen as an island PF (Vergé et al., 2015) implementation of the method presented by Naesseth et al. (2014b). Specifically, for a spatio-temporal models they run an island PF over both spatial and temporal

dimensions. However, the ST-PF does not generate an approximation of the fully adapted SMC sampler.

Another key distinction between NSMC and ST-PF is that in the latter each particle in the “outer” SMC sampler comprises a complete particle system from the “inner” SMC sampler. For NSMC, on the other hand, the particles will simply correspond to different hypotheses about the latent variables (as in standard SMC), regardless of how many samplers that are nested. This is a key feature of NSMC, since it implies that it is easily distributed over the particles. The main computational effort of Algorithm 3 is the construction of $\{q^j\}_{j=1}^N$ and the calls to the Simulate procedure, which can be done independently for each particle. This leads to improved memory efficiency and lower communication costs. Furthermore, we have found (see Section 6) that NSMC can outperform ST-PF even when run on a single machine with matched computational costs.

Another strength of NSMC methods are their relative ease of implementation, which we show in Section 6.3. We use the framework to sample from what is essentially a cubic grid Markov random field (MRF) model just by implementing three nested samplers, each with a target distribution defined on a simple chain.

There are also other SMC-based methods designed for high-dimensional problems, e.g., the block PF studied by Rebeschini and van Handel (2015), the location particle smoother by Briggs et al. (2013) and the PF-based methods reviewed in Djuric and Bugallo (2013). However, these methods are all inconsistent, as they are based on various approximations that result in systematic errors.

The previously mentioned PMCMC (Andrieu et al., 2010) is a related method, where SMC is used as a component of an MCMC algorithm. We make use of a very similar extended space approach to motivate the validity of our algorithm. Note that our proposed algorithm can be used as a component in PMCMC and most of the other algorithms mentioned above, which further increases the scope of models it can handle.

6. Experimental Results

We illustrate NSMC on three high-dimensional examples, both with real and synthetic data. We compare NSMC with standard (bootstrap) PF and the ST-PF of Beskos et al. (2014a) with equal computational budgets on a single machine (i.e., neglecting the fact that NSMC is more easily distributed). These methods are, to the best of our knowledge, the only other available *consistent* online methods for full Bayesian inference in general sequential models. For more detailed explanations of the models and additional results, see Appendix A.3².

6.1 Gaussian State Space Model

We start by considering a high-dimensional Gaussian state space model, where we have access to the true solution through belief propagation. The latent variables and measurements $\{X_{1:k}, Y_{1:k}\}$, with $\{X_k, Y_k\} = \{X_{k,l}, Y_{k,l}\}_{l=1}^d$, are modeled by a $d \times k$ lattice Gaussian MRF. The true data is simulated from a nearly identical state space model (see Appendix A.3.1). We run a 2-level NSMC sampler. The outer level is fully adapted, i.e. the proposal distribution is $q_k = p(x_k | x_{k-1}, y_k)$, which thus constitute the target distribution for the inner level. To generate properly weighted samples from q_k , we use a bootstrap PF operating on the d components of the vector x_k . Note that

2. Code available at <https://github.com/can-cs/nestedsmc>

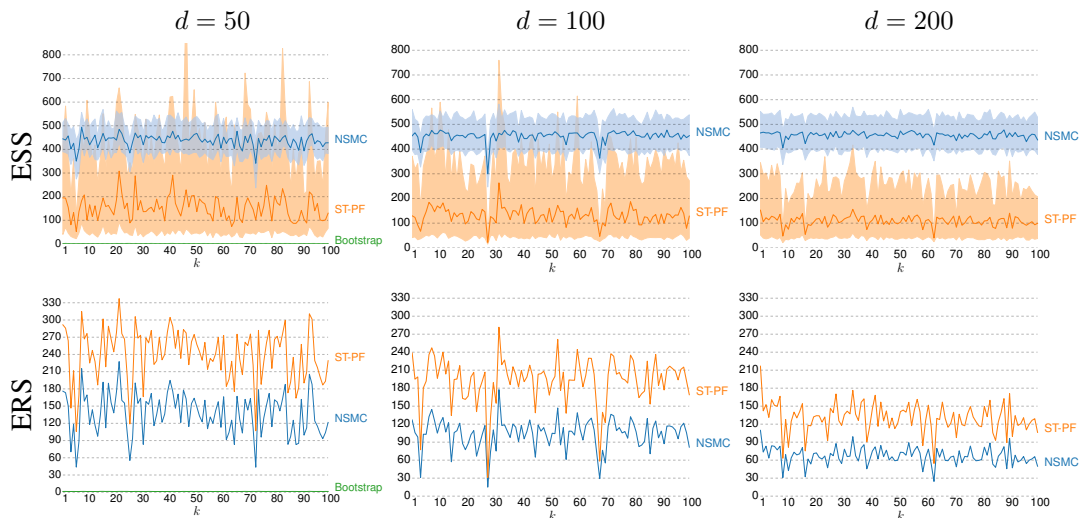


Figure 2: *Top*: Median (over dimension) ESS (4) and 15–85% percentiles (shaded region). *Bottom*: The ERS (5) based on the resampling weights in the (outermost) particle filter. The results are based on 100 independent runs for the Gaussian MRF with dimension d .

we only use bootstrap proposals where the actual sampling takes place, and that the conditional distribution $p(x_k | x_{k-1}, y_k)$ is not explicitly used.

We simulate data from this model for $k = 1, \dots, 100$ for different values of $d = \dim(x_k) \in \{50, 100, 200\}$. The exact filtering marginals are computed using belief propagation. We compare with both the ST-PF and standard (bootstrap) PF.

The results are evaluated based on the effective sample size (ESS, see e.g. Fearnhead et al. (2010b)) defined as,

$$\text{ESS}(x_{k,l}) = \left(\mathbb{E} \left[\frac{(\hat{x}_{k,l} - \mu_{k,l})^2}{\sigma_{k,l}^2} \right] \right)^{-1}, \quad (4)$$

where $\hat{x}_{k,l}$ denote the mean estimates and $\mu_{k,l}$ and $\sigma_{k,l}^2$ denote the true mean and variance of $x_{k,l} | y_{1:k}$ obtained from belief propagation. The expectation in (4) is approximated by averaging over 100 independent runs of the involved algorithms. The ESS reflects the estimator accuracy, obvious by the definition which is tightly related to the mean-squared-error. Intuitively the ESS corresponds to the equivalent number of i.i.d. samples needed for the same accuracy.

We also consider the effective resample size (ERS, Kong et al. (1994)), which is based on the resampling weights at the top levels in the respective SMC algorithms,

$$\text{ERS} = \frac{\left(\sum_{i=1}^N \hat{Z}_{q_k}^i \right)^2}{\sum_{i=1}^N \left(\hat{Z}_{q_k}^i \right)^2}. \quad (5)$$

The ERS is an estimate of the effective number of unique particles (or particle systems in the case of ST-PF) available at each resampling step.

We use $N = 500$ and $M = 2 \cdot d$ for NSMC and match the computational time for ST-PF and bootstrap PF. We report the results in Figure 2. The bootstrap PF is omitted from $d = 100$,

200 due to its poor performance already for $d = 50$ (which is to be expected). Each dimension $l = 1, \dots, d$ provides us with a value of the ESS, so we present the median (lines) and 15–85% percentiles (shaded regions) in the first row of Figure 2. The ERS is displayed in the second row of Figure 2. Note that ESS gives a better reflection of estimation accuracy than ERS.

We have conducted additional experiments with different model parameters and different choices for N and M (some additional results are given in Appendix A.3.1). Overall the results seem to be in agreement with the ones presented here, however ST-PF seems to be more robust to the trade-off between N and M . A rule-of-thumb for NSMC is to generally try to keep N as high as possible, while still maintaining a reasonably large ERS.

6.2 Non-Gaussian State Space Model

Next, we consider an example with a non-Gaussian SSM, borrowed from Beskos et al. (2014a) where the full details of the model are given. The transition probability $p(x_k | x_{k-1})$ is a localised Gaussian mixture and the measurement probability $p(y_k | x_k)$ is t-distributed. The model dimension is $d = 1024$. Beskos et al. (2014a) report improvements for ST-PF over both the bootstrap PF and the block PF by Rebeschini and van Handel (2015). We use $N = M = 100$ for both ST-PF and NSMC (the special structure of this model implies that there is no significant computational overhead from running backward sampling) and the bootstrap PF is given $N = 10000$. In Figure 3

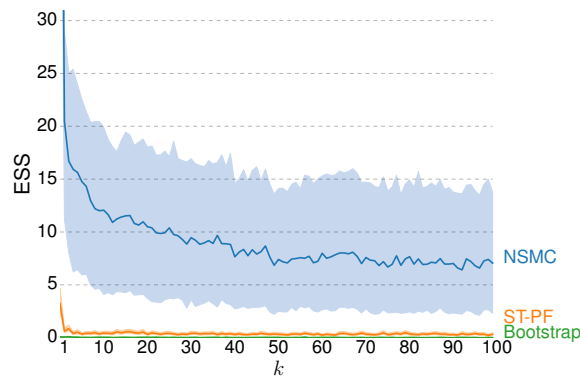


Figure 3: Median ESS with 15 – 85% percentiles (shaded region) for the non-Gaussian SSM.

we report the ESS (4), estimated according to Carpenter et al. (1999). The ESS for the bootstrap PF is close to 0, for ST-PF around 1–2, and for NSMC slightly higher at 7–8. However, we note that all methods perform quite poorly on this model, and to obtain satisfactory results it would be necessary to use more particles.

6.3 Spatio-Temporal Model – Drought Detection

In this final example we study the problem of detecting droughts based on measured precipitation data (Jones and Harris, 2013) for different locations on earth. We look at the situation in North America during the years 1901–1950 and the Sahel region in Africa during the years 1950–2000. These spatial regions and time frames were chosen since they include two of the most devastating droughts during the last century, the so-called Dust Bowl in the US during the 1930s (Schubert

et al., 2004) and the decades long drought in the Sahel region in Africa starting in the 1960s (Foley et al., 2003; Hoerling et al., 2006). We consider the spatio-temporal model defined by Fu et al.

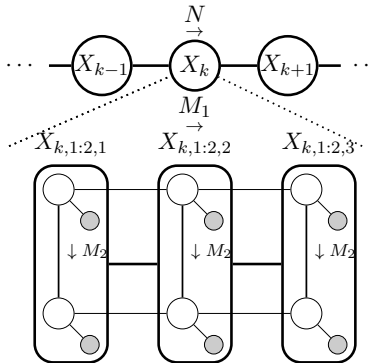


Figure 4: Illustration of the three-level NSMC.

(2012) and compare with the results therein. Each location in a region is modelled to be in either a *normal* state 0 or in an *abnormal* state 1 (drought). Measurements are given by precipitation (in millimeters) for each location and year. At every time instance k our latent structure is described by a rectangular 2D grid $X_k = \{X_{k,i,j}\}_{i=1,j=1}^{I,J}$; in essence this is the model showcased in Figure 1. Fu et al. (2012) considers the problem of finding the maximum a posteriori configuration, using a linear programming relaxation. We will instead compute an approximation of the full posterior filtering distribution $\bar{\pi}_k(x_k) = p(x_k | y_{1:k})$. The rectangular structure is used to instantiate an NSMC method that on the first level targets the full posterior filtering distribution. To sample from X_k we run, on the second level, an NSMC procedure that operates on the “columns” $X_{k,1:I,j}$, $j = 1, \dots, J$. Finally, to sample each column $X_{k,1:I,j}$ we run a third level of SMC, that operates on the individual components $X_{k,i,j}$, $i = 1, \dots, I$, using a bootstrap proposal. The structure of our NSMC method applied to this particular problem is illustrated in Figure 4.

Figure 5 gives the results on the parts of North America that we consider. The first row shows the number of locations where the estimate of $p(x_{k,i,j} = 1)$ exceeds $\{0.5, 0.7, 0.9\}$, for both regions. These results seems to be in agreement with Fu et al. (2012, Figures 3, 6). However, we also receive an approximation of the full posterior and can visualise uncertainty in our estimates, as illustrated by the three different levels of posterior probability for drought. In general, we obtain a rich sample diversity from the posterior distribution. However, for some problematic years the sampler degenerates, with the result that the three credibility levels all coincide. This is also visible in the second row of Figure 5, where we show the posterior estimates $p(x_{k,i,j} | y_{1:k})$ for the years 1939–1941, overlayed on the regions of interest. For year 1940 the sampler degenerates and only reports 0-1 probabilities for all sites. Naturally, one way to improve the estimates is to run the sampler with a larger number of particles, which has been kept very low in this proof-of-concept.

7. Conclusions

We have shown that a straightforward NSMC implementation with fairly few particles can attain reasonable approximations to the filtering problem for dimensions in the order of hundreds, or even thousands. This means that NSMC methods takes the SMC framework an important step closer to being viable for high-dimensional statistical inference problems. However, NSMC is not

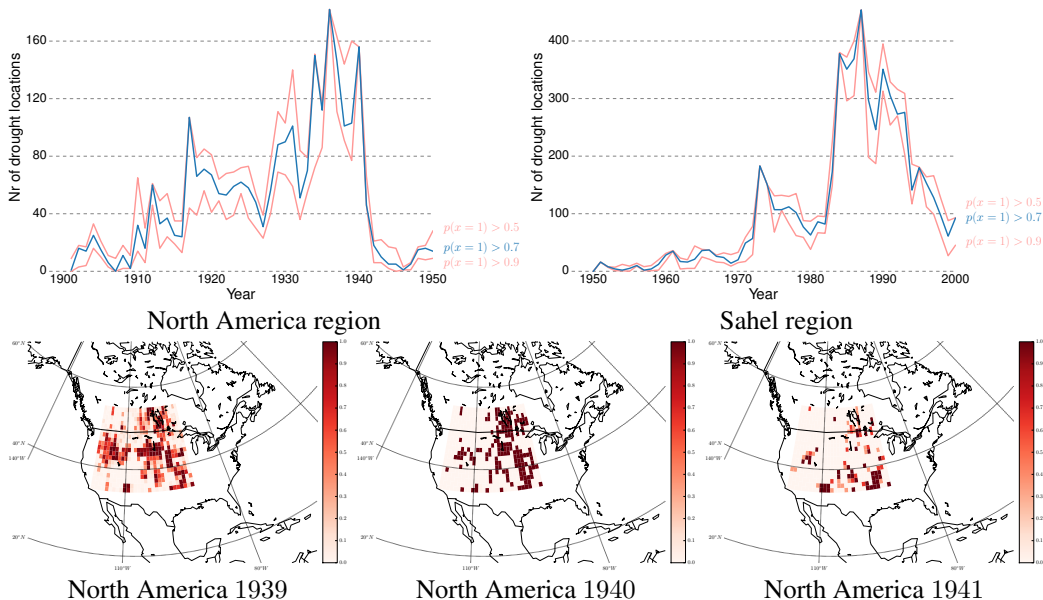


Figure 5: *Top:* Number of locations with estimated $p(x = 1) > \{0.5, 0.7, 0.9\}$ for the two regions. *Bottom:* Estimate of $p(x_{t,i} = 1)$ for all sites over a span of 3 years. All results for $N = 100, N_1 = \{30, 40\}, N_2 = 20$.

a silver bullet for solving high-dimensional inference problems, and the approximation accuracy will be highly model dependent. Hence, much work remains to be done, for instance on combining NSMC with other techniques for high-dimensional inference such as localisation (Rebeschini and van Handel, 2015) and annealing (Beskos et al., 2014b), in order to solve even more challenging problems.

Acknowledgments

This work was supported by the projects: *Learning of complex dynamical systems* (Contract number: 637-2014-466) and *Probabilistic modeling of dynamical systems* (Contract number: 621-2013-5524), both funded by the Swedish Research Council.

Appendix A. Appendix

In this appendix we start out in Section A.1 by providing a more general formulation of the NSMC method and proofs of the central limit and proper weighting theorems of the main manuscript. We also detail (Section A.2) a straightforward extension of nested IS to a sequential version. We show that a special case of this nested sequential IS turns out to be more or less equivalent to the importance sampling squared algorithm by Tran et al. (2013). This relationship serves as evidence that illustrates that the NSMC framework being more widely applicable than the scope of problems considered in this article. Finally, in Section A.3 we give more details and results on the experiments considered in the main manuscript.

A.1 Nested Sequential Monte Carlo

We start by presenting a general formulation of a nested auxiliary SMC sampler in Algorithm 5. In this formulation, $q_k(x_k | x_{1:k-1})$ is an arbitrary (unnormalised) proposal, normalised by

$$Z_{q_k}(x_{1:k-1}) = \int q_k(x_k | x_{1:k-1}) dx_k.$$

Furthermore, the resampling weights are obtain by multiplying the importance weights with the arbitrary *adjustment multipliers* $\nu_{k-1}(x_{1:k-1}, Z_{q_k})$, which may depend on both the state sequence $x_{1:k-1}$ and the normalising constant (estimate). The fully adapted NSMC sampler (Algorithm 3 in the main document) is obtained as a special case if we choose

$$q_k(x_k | x_{1:k-1}) = \frac{\pi_k(x_{1:k})}{\pi_{k-1}(x_{1:k-1})}$$

and $\nu_{k-1}(x_{1:k-1}, Z_{q_k}) = Z_{q_k}$, in which case the importance weights are indeed given by $W_k^i \equiv 1$.

A.1.1 NESTED SMC IS SMC

The validity of Algorithm 5 can be established by interpreting the algorithm as a standard SMC procedure for a sequence of extended target distributions. If \widehat{Z}_{q_k} is computed deterministically, proper weighting (i.e., unbiasedness) ensures that $\widehat{Z}_{q_k} = Z_{q_k}$ and it is evident that the algorithm reduces to a standard SMC sampler. Hence, we consider the case when the normalising constant estimates \widehat{Z}_{q_k} are random.

For $k = 1, \dots, n + 1$, let us introduce the random variable U_{k-1} which encodes the complete internal state of the object q generated by $q = Q(q_k(\cdot | x_{1:k-1}), M)$. Let the distribution of U_{k-1} be denoted as $\bar{\psi}_{k-1}^M(u_{k-1} | x_{1:k-1})$. To put Algorithm 5 into a standard (auxiliary) SMC framework, we shall interpret steps 2a–2b of Algorithm 5 as being the last two steps carried out during iteration $k - 1$, rather than the first two steps carried out during iteration k . This does not alter the algorithm *per se*, but it results in that the resampling step is conducted first at each iteration, which is typically the case for standard auxiliary SMC formulations.

The estimator of the normalising constant is computable from the internal state of q , so that we can introduce a function τ_k such that $\widehat{Z}_{q_k} = \tau_k(U_{k-1})$. Furthermore, note that the simulation of X_k via $X_k = q.\text{Simulate}()$ is based solely on the internal state U_{k-1} , and denote by $\bar{\gamma}_k^M(x_k | U_{k-1})$ the distribution of X_k .

Algorithm 5 Nested SMC (auxiliary SMC formulation)

1. Set $\{X_0^i\}_{i=1}^N$ to arbitrary dummy variables. Set $W_0^i = 1$ for $i = 1, \dots, N$. Set $\widehat{Z}_{\pi_0} = 1$.
 2. **for** $k = 1$ **to** n
 - (a) Initialise $\mathbf{q}^j = \mathbb{Q}(q_k(\cdot | X_{1:k-1}^j), M)$ for $j = 1, \dots, N$.
 - (b) Compute $\widehat{Z}_{q_k}^j = \mathbf{q}^j.\text{GetZ}()$ for $j = 1, \dots, N$.
 - (c) Compute $\widehat{\nu}_{k-1}^j = \nu_{k-1}(X_{1:k-1}^j, \widehat{Z}_{q_k}^j)$ for $j = 1, \dots, N$.
 - (d) Draw $m_k^{1:N}$ from a multinomial distribution with probabilities $\frac{\widehat{\nu}_{k-1}^j W_{k-1}^j}{\sum_{\ell=1}^N \widehat{\nu}_{k-1}^\ell W_{k-1}^\ell}$ for $j = 1, \dots, N$.
 - (e) Set $L \leftarrow 0$
 - (f) **for** $j = 1$ **to** N
 - i. Compute $X_k^i = \mathbf{q}^j.\text{Simulate}()$ and let $X_{1:k}^i = (X_{1:k-1}^j, X_k^i)$ for $i = L+1, \dots, L+m_k^j$.
 - ii. Compute $W_k^i = \frac{\pi_k(X_{1:k}^i)}{\pi_{k-1}(X_{1:k-1}^j)} \frac{\widehat{Z}_{q_k}^j}{\widehat{\nu}_{k-1}^j q_k(X_k^i | X_{1:k-1}^j)}$ for $i = L+1, \dots, L+m_k^j$.
 - iii. **delete** \mathbf{q}^j .
 - iv. Set $L \leftarrow L + m_k^j$.
 - (g) Compute $\widehat{Z}_{\pi_k} = \widehat{Z}_{\pi_{k-1}} \times \left\{ \frac{1}{N} \sum_{j=1}^N \widehat{\nu}_{k-1}^j W_{k-1}^j \right\} \times \left\{ (\sum_{j=1}^N W_k^j) / (\sum_{j=1}^N W_{k-1}^j) \right\}$.
-

Lemma 7. Assume that \mathbb{Q} satisfies condition (A1) in the main manuscript. Then,

$$\int \tau_k(u_{k-1}) \bar{\gamma}_k^M(x_k | u_{k-1}) \bar{\psi}_{k-1}^M(u_{k-1} | x_{1:k-1}) du_{k-1} = q_k(x_k | x_{1:k-1}).$$

Proof The pair $(X_k, \tau_k(U_{k-1}))$ are properly weighted for q_k . Hence, for a measurable function f ,

$$\begin{aligned} \mathbb{E}[f(X_k) \tau_k(U_{k-1}) | x_{1:k-1}] &= \iint f(x_k) \tau_k(u_{k-1}) \bar{\gamma}_k^M(x_k | u_{k-1}) \bar{\psi}_{k-1}^M(u_{k-1} | x_{1:k-1}) du_{k-1} dx_k \\ &= Z_k(x_{1:k-1}) \int f(x_k) \bar{q}_k(x_k | x_{1:k-1}) dx_k = \int f(x_k) q_k(x_k | x_{1:k-1}) dx_k. \end{aligned}$$

Since f is arbitrary, the result follows. ■

We can now define the sequence of (unnormalised) extended target distributions for the Nested SMC sampler as,

$$\Pi_k(x_{1:k}, u_{0:k}) := \frac{\tau_k(u_{k-1}) \bar{\psi}_k^M(u_k | x_{1:k}) \bar{\gamma}_k^M(x_k | u_{k-1})}{q_k(x_k | x_{1:k-1})} \frac{\pi_k(x_{1:k})}{\pi_{k-1}(x_{1:k-1})} \Pi_{k-1}(x_{1:k-1}, u_{0:k-1}),$$

and $\Pi_0(u_0) = \bar{\psi}_0^M(u_0)$. We write $\Theta_k = \mathbf{X}_k \times \mathbf{U}_k$ for the domain of Π_k .

Lemma 8. *Assume that Q satisfies condition (A1) in the main manuscript. Then,*

$$\int \tau_k(u_{k-1}) \bar{\gamma}_k^M(x_k | u_{k-1}) \Pi_{k-1}(x_{1:k-1}, u_{0:k-1}) du_{0:k-1} = \pi_{k-1}(x_{1:k-1}) q_k(x_k | x_{1:k-1}).$$

Proof The proof follows by induction. At $k = 1$, we have $\int \tau_1(u_0) \bar{\gamma}_1^M(x_1 | u_0) \bar{\psi}_0^M(u_0) du_0 = q_1(x_1)$ by Lemma 7. Hence, assume that the hypothesis holds for $k \geq 1$ and consider

$$\begin{aligned} & \int \tau_{k+1}(u_k) \bar{\gamma}_{k+1}^M(x_{k+1} | u_k) \Pi_k(x_{1:k}, u_{0:k}) du_{0:k} \\ &= \frac{\pi_k(x_{1:k})}{\pi_{k-1}(x_{1:k-1}) q_k(x_k | x_{1:k-1})} \\ & \cdot \int \tau_{k+1}(u_k) \bar{\gamma}_{k+1}^M(x_{k+1} | u_k) \tau_k(u_{k-1}) \bar{\psi}_k^M(u_k | x_{1:k}) \bar{\gamma}_k^M(x_k | u_{k-1}) \Pi_{k-1}(x_{1:k-1}, u_{0:k-1}) du_{0:k} \\ &= \frac{\pi_k(x_{1:k}) \left(\int \tau_{k+1}(u_k) \bar{\gamma}_{k+1}^M(x_{k+1} | u_k) \bar{\psi}_k^M(u_k | x_{1:k}) du_k \right)}{\pi_{k-1}(x_{1:k-1}) q_k(x_k | x_{1:k-1})} \\ & \cdot \int \tau_k(u_{k-1}) \bar{\gamma}_k^M(x_k | u_{k-1}) \Pi_{k-1}(x_{1:k-1}, u_{0:k-1}) du_{0:k-1} \\ &= \frac{\pi_k(x_{1:k}) q_{k+1}(x_{k+1} | x_{1:k}) \pi_{k-1}(x_{1:k-1}) q_k(x_k | x_{1:k-1})}{\pi_{k-1}(x_{1:k-1}) q_k(x_k | x_{1:k-1})} = \pi_k(x_{1:k}) q_{k+1}(x_{k+1} | x_{1:k}), \end{aligned}$$

where the penultimate equality follows by applying Lemma 7 and the induction hypothesis to the two integrals, respectively. \blacksquare

As a corollary to Lemma 8, it follows that

$$\int \Pi_k(x_{1:k}, u_{0:k}) du_{0:k} = \pi_k(x_{1:k}). \quad (6)$$

Consequently, Π_k is normalised by the same constant Z_{π_k} as π_k , and by defining $\bar{\Pi}_k(x_{1:k}, u_{0:k}) := Z_{\pi_k}^{-1} \Pi_k(x_{1:k}, u_{0:k})$ we obtain a probability distribution which admits $\bar{\pi}_k$ as a marginal (note that $\bar{\Pi}_0 = \Pi_0$, which is normalised by construction). This implies that we can use $\bar{\Pi}_k$ as a proxy for $\bar{\pi}_k$ in a Monte Carlo algorithm, i.e., samples drawn from $\bar{\Pi}_k$ can be used to compute expectations w.r.t. $\bar{\pi}_k$. This is precisely what Algorithm 5 does; it is a standard auxiliary SMC sampler for the (unnormalised) target sequence Π_k , $k = 0, \dots, n$, with adjustment multiplier weights $\nu_{k-1}(x_{1:k-1}, \tau_k(u_{k-1}))$ and proposal distribution $\bar{\gamma}_k^M(x_k | u_{k-1}) \bar{\psi}_k^M(u_k | x_{1:k})$. The (standard) weight function for this sampler is thus given by

$$W_k(x_{1:k}, u_{0:k}) \propto \frac{\pi_k(x_{1:k})}{\pi_{k-1}(x_{1:k-1})} \frac{\tau_k(u_{k-1})}{\nu_{k-1}(x_{1:k-1}, \tau_k(u_{k-1})) q_k(x_k | x_{1:k-1})}, \quad (7)$$

which is the same as the expression on line 2(f)ii of Algorithm 5.

A.1.2 CENTRAL LIMIT THEOREM – PROOF OF THEOREM 2 IN THE MAIN MANUSCRIPT

Now that we have established that Nested SMC is in fact a standard auxiliary SMC sampler, albeit on an extended state space, we can reuse existing convergence results from the SMC literature; see

e.g., Johansen and Doucet (2008); Douc and Moulines (2008); Douc et al. (2009); Chopin (2004) or the extensive textbook by Del Moral (2004).

Here, in order to prove Theorem 2 of the main manuscript, we make use of the result for the auxiliary SMC sampler by Johansen and Doucet (2008), which in turn is based on the central limit theorem by Chopin (2004). The technique used by Johansen and Doucet (2008) is to reinterpret (as detailed below) the auxiliary SMC sampler as a *sequential importance sampling and resampling* (SISR) particle filter, by introducing the modified (unnormalised) target distribution

$$\Pi'_k(x_{1:k}, u_{0:k}) := \nu_k(x_{1:k}, \tau_{k+1}(u_k)) \Pi_k(x_{1:k}, u_{0:k}). \quad (8)$$

The auxiliary SMC sampler described in the previous section can then be viewed as a SISR algorithm for (8). Indeed, if we write $\bar{Q}_k^M(x_k, u_k | x_{1:k-1}, u_{k-1}) := \bar{\psi}_k^M(u_k | x_{1:k}) \bar{\gamma}_k^M(x_k | u_{k-1})$ for the joint proposal distribution of (x_k, u_k) , then the weight function for this SISR sampler is given by

$$\begin{aligned} W'_k(x_{1:k}, u_{0:k}) &:= \frac{\bar{\Pi}'_k(x_{1:k}, u_{0:k})}{\bar{Q}_k^M(x_k, u_k | x_{1:k-1}, u_{k-1}) \bar{\Pi}'_{k-1}(x_{1:k-1}, u_{0:k-1})} \\ &\propto \nu_k(x_{1:k}, \tau_{k+1}(u_k)) W_k(x_{1:k}, u_{0:k}), \end{aligned} \quad (9)$$

where W_k is defined in (7). This weight expression thus accounts for both the importance weights and the adjustment multipliers of the auxiliary SMC sampler formulation.

Since this SISR algorithm does not target $\bar{\Pi}_k$ (and thus not $\bar{\pi}_k$) directly, we use an additional IS step to compute estimators of expectations w.r.t. to $\bar{\pi}$. The proposal distribution for this IS procedure is given by

$$\bar{\Gamma}_k(x_{1:k}, u_{0:k}) := \bar{Q}_k^M(x_k, u_k | x_{1:k-1}, u_{k-1}) \bar{\Pi}'_{k-1}(x_{1:k-1}, u_{0:k-1}). \quad (10)$$

Note that we obtain an approximation of (10) after the propagation Step 2(f)i of Algorithm 5, but before the weighting step. The resulting IS weights, for target distribution $\bar{\Pi}_k(x_{1:k}, u_{0:k})$ and with proposal distribution (10), are given by

$$\frac{\bar{\Pi}_k(x_{1:k}, u_{0:k})}{\bar{\Gamma}_k(x_{1:k}, u_{0:k})} =: \omega_k(x_{1:k}, u_{0:k}) \propto W_k(x_{1:k}, u_{0:k}).$$

Hence, with $f : X_k \mapsto \mathbb{R}^d$ being a test function of interest we can estimate $\mathbb{E}_{\bar{\pi}_k}[f] = \mathbb{E}_{\bar{\Gamma}_k}[f]$ (with obvious abuse of notation) by the estimator

$$\sum_{i=1}^N \frac{W_k^i f(X_{1:k}^i)}{\sum_{\ell=1}^N W_k^\ell}, \quad (11)$$

which, again, is in agreement with Algorithm 5.

We have now reinterpreted the NSMC algorithm; first as a standard *auxiliary* SMC sampler, and then further as a standard SISR method. Consequently, we are now in the position of directly applying, e.g., the central limit theorem by Chopin (2004, Theorem 1). The conditions and the statement of the theorem are reproduced here for clarity.

For any measurable function $f : \Theta_0 \mapsto \mathbb{R}^d$, let $\tilde{V}_0^M(f) = \text{Var}_{\tilde{\psi}_0^M}(f)$ and define, for any measurable function $f : \Theta_k \mapsto \mathbb{R}^d$,

$$\begin{aligned}\tilde{V}_k^M(f) &= \hat{V}_{k-1}^M(\mathbb{E}_{\tilde{Q}_k^M}[f]) + \mathbb{E}_{\tilde{\Pi}'_{k-1}}[\text{Var}_{\tilde{Q}_k^M}(f)], & k > 0, \\ V_k^M(f) &= \tilde{V}_k^M(W_k'(f - \mathbb{E}_{\tilde{\Pi}'_k}[f])), & k \geq 0, \\ \hat{V}_k^M(f) &= V_k^M(f) + \text{Var}_{\tilde{\Pi}'_k}(f), & k \geq 0.\end{aligned}$$

Define recursively Φ_k to be the set of measurable functions $f : \Theta_k \mapsto \mathbb{R}^d$ such that there exists a $\delta > 0$ with $\mathbb{E}_{\tilde{\Gamma}_k}[\|W_k'f\|^{2+\delta}] < \infty$ and such that the function $(x_{1:k-1}, u_{0:k-1}) \mapsto \mathbb{E}_{\tilde{Q}_k^M}[W_k'f]$ is in Φ_{k-1} . Furthermore, assume that the identity function $f \equiv 1$ belongs to Φ_k for each k . Then, it follows by Chopin (2004, Theorem 1 and Lemma A.1) that

$$N^{1/2} \left(\sum_{i=1}^N \frac{1}{N} f(X_{1:k}^i, U_{0:k}^i) - \mathbb{E}_{\tilde{\Gamma}_k}[f] \right) \xrightarrow{D} \mathcal{N}(0, \tilde{V}_k^M(f)), \quad (12)$$

for any function f such that the function $(x_{1:k-1}, u_{0:k-1}) \mapsto \mathbb{E}_{\tilde{Q}_k^M}[f - \mathbb{E}_{\tilde{\Gamma}_k}[f]]$ is in Φ_{k-1} and there exists a $\delta > 0$ such that $\mathbb{E}_{\tilde{\Gamma}_k}[\|f\|^{2+\delta}] < \infty$. The convergence in (12) thus holds for the unweighted samples obtained after the propagation Step 2(f)i of Algorithm 5, but before the weighting step.

To complete the proof, it remains to translate (12) into a similar result for the IS estimator (11). To this end we make use of Chopin (2004, Lemma A.2) which is related to the IS correction step of the SMC algorithm. Specifically, for a function $f : \mathsf{X}_k \mapsto \mathbb{R}^d$, let $f^e : \Theta_k \mapsto \mathbb{R}^d$ denote the extension of f to Θ_k , defined by $f^e(x_{1:k}, u_{0:k}) = f(x_{1:k})$. Then, for any $f : \mathsf{X}_k \mapsto \mathbb{R}^d$ such that the function $(x_{1:k-1}, u_{0:k-1}) \mapsto \mathbb{E}_{\tilde{Q}_k^M}[\omega_k f^e]$ is in Φ_{k-1} and there exists a $\delta > 0$ such that $\mathbb{E}_{\tilde{\Gamma}_k}[\|\omega_k f^e\|^{2+\delta}] < \infty$, we have

$$N^{1/2} \left(\sum_{i=1}^N \frac{W_k^i f(X_{1:k}^i)}{\sum_{\ell=1}^N W_k^\ell} - \bar{\pi}_k(f) \right) \xrightarrow{D} \mathcal{N}(0, \Sigma_k^M(f)),$$

where $\{(X_{1:k}^i, W_k^i)\}_{i=1}^M$ are generated by Algorithm 5 and $\Sigma_k^M(f) = \tilde{V}_k^M(\omega_k(f^e - \mathbb{E}_{\tilde{\Pi}_k}[f^e]))$.

A.1.3 NESTED SMC GENERATES PROPERLY WEIGHTED SAMPLES – PROOF OF THEOREM 6 IN THE MAIN MANUSCRIPT

In the previous two sections we showed that the NSMC procedure is a valid inference algorithm for $\bar{\pi}_n$. Next, we turn our attention to the modularity of the method and the validity of using the algorithm as a component in another NSMC sampler. Let us start by stating a more general version of the backward simulator in Algorithm 6. Clearly, if the forward NSMC procedure is fully adapted $W_k^i \equiv 1$, Algorithm 6 reduces to the backward simulator stated in the main manuscript.

We will now show that the pair $(\hat{Z}_{\pi_n}, \tilde{X}_{1:n})$ generated by Algorithms 5 and 6 is properly weighted for $\pi_n(x_{1:n})$, and thereby prove Theorem 6 in the main manuscript.

The proof is based on the *particle Markov chain Monte Carlo* (PMCMC) construction (Andrieu et al., 2010). The idea used by Andrieu et al. (2010) was to construct an extended target distribution, incorporating all the random variables generated by an SMC sampler as auxiliary variables. This opened up for using SMC approximations within MCMC in a provably correct way; these seemingly approximate methods simply correspond to standard MCMC samplers for the (nonstandard)

Algorithm 6 Backward simulator

1. Draw B_n from a categorical distribution with probabilities $\frac{W_n^j}{\sum_{\ell=1}^N W_n^\ell}$ for $j = 1, \dots, N$.
 2. Set $\tilde{X}_n = X_n^{B_n}$.
 3. **for** $k = n - 1$ **to** 1
 - (a) Compute $\tilde{W}_k^j = W_k^j \frac{\pi_n((X_{1:k}^j, \tilde{X}_{k+1:n}))}{\pi_k(X_{1:k}^j)}$ for $j = 1, \dots, N$.
 - (b) Draw B_k from a categorical distribution with probabilities $\frac{\tilde{W}_k^j}{\sum_{\ell=1}^N \tilde{W}_k^\ell}$ for $j = 1, \dots, N$.
 - (c) Set $\tilde{X}_{k:n} = (X_k^{B_k}, \tilde{X}_{k+1:n})$.
 4. **return** $\tilde{X}_{1:n}$
-

extended target distribution. Here we will use the same technique to prove the proper weighing property of the NSMC procedure.

We start by introducing some additional notation for the auxiliary variables of the extended target construction. While Algorithm 5 is expressed using multinomial random variables $m_k^{1:N}$ in the resampling step, it is more convenient for the sake of the proof to explicitly introduce the *ancestor indices* $\{A_k^i\}_{i=1}^N$; see e.g., Andrieu et al. (2010). That is, A_k^i is a categorical random variable on $\{1, \dots, N\}$, such that $X_{1:k-1}^{A_k^i}$ is ancestor particle at iteration $k - 1$ of particle X_k^i . The resampling Step 2d of Algorithm 5 can then equivalently be expressed as: simulate independently $\{A_k^i\}_{i=1}^N$ from the categorical distribution with probabilities

$$\frac{\hat{\nu}_{k-1}^j W_{k-1}^j}{\sum_{\ell=1}^N \hat{\nu}_{k-1}^\ell W_{k-1}^\ell}.$$

Let $\mathbf{X}_k = \{X_k^1, \dots, X_k^N\}$, $\mathbf{U}_k = \{U_k^1, \dots, U_k^N\}$, and $\mathbf{A}_k = \{A_k^1, \dots, A_k^N\}$, denote all the particles, internal states of the proposals, and ancestor indices, respectively, generated at iteration k of the NSMC algorithm. We can then write down the joint distribution of all the random variables generated in executing Algorithm 5 (up to an irrelevant permutation of the particle indices) as,

$$\bar{\Psi}_{\text{NSMC}}(\mathbf{x}_{1:n}, \mathbf{u}_{0:n}, \mathbf{a}_{1:n}) = \left\{ \prod_{i=1}^N \bar{\psi}_0^M(u_0^i) \right\} \prod_{k=1}^n \left\{ \prod_{i=1}^N \frac{\hat{\nu}_{k-1}^{a_k^i} W_{k-1}^{a_k^i}}{\sum_{\ell=1}^N \hat{\nu}_{k-1}^\ell W_{k-1}^\ell} \bar{Q}_k^M(x_k^i, u_k^i | x_{1:k-1}^{a_k^i}, u_{k-1}^{a_k^i}) \right\}, \quad (13)$$

where we interpret $\hat{\nu}_k^i$ and W_k^i as deterministic functions of $(x_{1:k}^i, u_{0:k}^i)$.

Let B_n denote a random variable defined on $\{1, \dots, N\}$. The extended target distribution for PMCMC samplers corresponding to (13) is then given by

$$\bar{\Phi}(\mathbf{x}_{1:n}, \mathbf{u}_{0:n}, \mathbf{a}_{1:n}, b_n) := \frac{\widehat{Z}_{\pi_n}}{Z_{\pi_n}} \frac{W_n^{b_n}}{\sum_{\ell=1}^N W_n^\ell} \bar{\Psi}_{\text{NSMC}}(\mathbf{x}_{1:n}, \mathbf{u}_{0:n}, \mathbf{a}_{1:n}), \quad (14)$$

where \widehat{Z}_{π_n} is a deterministic function of $(\mathbf{x}_{1:n}, \mathbf{u}_{0:n}, \mathbf{a}_{1:n})$. We know from Andrieu et al. (2010) that $\bar{\Phi}$ is a probability distribution which admits $\bar{\Pi}_n$ as its marginal distribution for $(X_{1:n}^{b_n}, U_{0:n}^{b_n})$. Consequently, by (6) it follows that the marginal distribution of $X_{1:n}^{b_n}$ is $\bar{\pi}_n$. For later reference we define recursively $b_{k-1} := a_k^{b_k}$ for $k = 1, \dots, n$, the particle indices for the trajectory obtained by tracing backward the genealogy of the b_n 'th particle at iteration n .

We now turn our attention to the backward simulator in Algorithm 6. Backward simulation has indeed been used in the context of PMCMC, see e.g. Whiteley (2010); Lindsten and Schön (2013); Lindsten et al. (2014). The strategy used for combining PMCMC with backward simulation is to show that each step of the backward sampler corresponds to a *partially collapsed* Gibbs sampling step for the extended target distribution $\bar{\Phi}$. This implies that the backward sampler leaves $\bar{\Phi}$ invariant.

We use the same approach here, but we need to be careful in how we apply the existing results, since the PMCMC distribution $\bar{\Phi}$ is defined w.r.t. to $\bar{\Pi}_n$, whereas the backward simulator of Algorithm 6 works with the original target distribution $\bar{\pi}_n$. Nevertheless, from the proof of Lemma 1 by Lindsten et al. (2014) it follows that we can write the following collapsed conditional distribution of $\bar{\Phi}$ as:

$$\begin{aligned} & \bar{\Phi}(b_k, u_{k:n}^{b_k} \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}, \mathbf{a}_{1:k}, x_{k+1:n}^{b_{k+1:n}}, b_{k+1:n}) \\ & \propto W_k^{b_k} \frac{\Pi_n(\{x_{1:k}^{b_k}, x_{k+1:n}^{b_{k+1:n}}\}, \{u_{0:k-1}^{a_k^{b_k}}, u_{k:n}^{b_{k:n}}\})}{\Pi_k(x_{1:k}^{b_k}, \{u_{0:k-1}^{a_k^{b_k}}, u_k^{b_k}\})} \bar{\psi}_k^M(u_k^{b_k} \mid x_{1:k}^{b_k}). \end{aligned} \quad (15)$$

To simplify this expression, consider,

$$\begin{aligned} \frac{\Pi_n(x_{1:n}, u_{0:n})}{\Pi_k(x_{1:k}, u_{0:k})} &= \prod_{s=k+1}^n \left\{ \frac{\tau_s(u_{s-1}) \bar{\psi}_s^M(u_s \mid x_{1:s}) \bar{\gamma}_s^M(x_s \mid u_{s-1})}{q_s(x_s \mid x_{1:s-1})} \frac{\pi_s(x_{1:s})}{\pi_{s-1}(x_{1:s-1})} \right\} \\ &= \frac{\bar{\psi}_n^M(u_n \mid x_{1:n})}{\bar{\psi}_k^M(u_k \mid x_{1:k})} \left\{ \prod_{s=k+1}^n \frac{\tau_s(u_{s-1}) \bar{\psi}_{s-1}^M(u_{s-1} \mid x_{1:s-1}) \bar{\gamma}_s^M(x_s \mid u_{s-1})}{q_s(x_s \mid x_{1:s-1})} \right\} \frac{\pi_n(x_{1:n})}{\pi_k(x_{1:k})}. \end{aligned} \quad (16)$$

By Lemma 7 we know that each factor of the product (in brackets) on the second line integrates to 1 over u_{s-1} . Hence, plugging (16) into (15) and integrating over $u_{k:n}^{b_{k:n}}$ yields

$$\bar{\Phi}(b_k \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}, \mathbf{a}_{1:k}, x_{k+1:n}^{b_{k+1:n}}, b_{k+1:n}) \propto W_k^{b_k} \frac{\pi_n(\{x_{1:k}^{b_k}, x_{k+1:n}^{b_{k+1:n}}\})}{\pi_k(x_{1:k}^{b_k})},$$

which coincides with the expression used to simulate the index B_k in Algorithm 6. Hence, simulation of B_k indeed corresponds to a partially collapsed Gibbs sampling step for $\bar{\Phi}$ and it will thus leave $\bar{\Phi}$ invariant. (Note that, in comparison with the PMCMC sampler derived by Lindsten

et al. (2014) we further marginalise over the variables $u_{k:n}^{b_{k:n}}$ which, however, still results in a valid partially collapsed Gibbs step.)

We now have all the components needed to prove proper weighting of the combined NSMC/backward simulation procedure. For notational simplicity, we write

$$\bar{\Psi}_{\text{BS},k}(b_k) = \bar{\Phi}(b_k \mid \mathbf{x}_{1:k}, \mathbf{u}_{0:k-1}, \mathbf{a}_{1:k}, x_{k+1:n}^{b_{k+1:n}}, b_{k+1:n}),$$

for the distribution of B_k in Algorithm 6. Let $(\hat{Z}_{\pi_n}, \tilde{X}_{1:n})$ be generated by Algorithms 5 and 6. Let f be a measurable function and consider

$$\begin{aligned} \mathbb{E}[\hat{Z}_{\pi_n} f(\tilde{X}_{1:n})] &= \int \hat{Z}_{\pi_n} f(X_{1:n}^{b'_{1:n}}) \left\{ \prod_{k=1}^n \bar{\Psi}_{\text{BS},k}(db'_k) \right\} \bar{\Psi}_{\text{NSMC}}(d(\mathbf{x}_{1:n}, \mathbf{u}_{0:n}, \mathbf{a}_{1:n})) \\ &= Z_{\pi_n} \int f(X_{1:n}^{b'_{1:n}}) \left\{ \prod_{k=1}^{n-1} \bar{\Psi}_{\text{BS},k}(db'_k) \right\} \bar{\Phi}(d(\mathbf{x}_{1:n}, \mathbf{u}_{0:n}, \mathbf{a}_{1:n}, b'_n)), \end{aligned}$$

where, for the second equality, we have used the definition (14) and noted that $\bar{\Psi}_{\text{BS},n}(b_n) = \frac{W_n^{b_n}}{\sum_{\ell=1}^N W_n^\ell}$. However, by the invariance of $\bar{\Psi}_{\text{BS},k}$ w.r.t. $\bar{\Phi}$, it follows that

$$\mathbb{E}[\hat{Z}_{\pi_n} f(\tilde{X}_{1:n})] = Z_{\pi_n} \int f(X_{1:n}^{b_{1:n}}) \bar{\Phi}(d(\mathbf{x}_{1:n}, \mathbf{u}_{0:n}, \mathbf{a}_{1:n}, b_n)) = Z_{\pi_n} \bar{\pi}_n(f),$$

which completes the proof.

A.2 Nested Sequential Importance Sampling

Here we give the definition of the nested sequential importance sampler and we show that a special case of this is the importance sampling squared (IS²) method by Tran et al. (2013).

A.2.1 NESTED SEQUENTIAL IMPORTANCE SAMPLING

We present a straightforward extension of the Nested IS class to a sequential IS version. Consider the following definition of the Nested SIS Q:

1. Algorithm 7 is executed at the construction of the object $p = Q(\pi_n, N)$, and $p.\text{GetZ}()$ returns the normalising constant estimate \hat{Z}_{π_n} .
2. $p.\text{Simulate}()$ simulates a categorical random variable B with $\mathbb{P}(B = i) = W_n^i / \sum_{\ell=1}^N W_n^\ell$ and returns $X_{1:n}^B$.

Note that we do not require that the procedure Q is identical for each individual proposal q_k , thus we have a flexibility in designing our algorithm as can be seen in the example in Section A.2.2. We can motivate the algorithm in the same way as for Nested IS and similar theoretical results hold, i.e. Nested SIS is properly weighted for π_n and it admits $\bar{\pi}_n$ as a marginal.

A.2.2 RELATION TO IS²

Here we will show how IS², proposed by Tran et al. (2013), can be viewed as a special case of Nested SIS. We are interested in approximating the posterior distribution of parameters θ given

Algorithm 7 Nested SIS (all i for $1, \dots, N$)

1. Initialise $q^i = Q(q_1(\cdot), M)$.
 2. Set $\widehat{Z}_{q_1}^i = q^i.\text{GetZ}()$, $X_1^i = q^i.\text{Simulate}()$.
 3. Set $W_1^i = \frac{\widehat{Z}_{q_1}^i \pi_1(X_1^i)}{q_1(X_1^i)}$.
 4. **delete** q^i .
 5. for $k = 2$ to n :
 - (a) Initialise $q^i = Q(q_k(\cdot | X_{1:k-1}^i), M)$.
 - (b) Set $\widehat{Z}_{q_k}^i = q^i.\text{GetZ}()$, $X_k^i = q^i.\text{Simulate}()$.
 - (c) Set $W_k^i = W_{k-1}^i \frac{\widehat{Z}_{q_k}^i \pi_k(X_k^i | X_{1:k-1}^i)}{q_k(X_k^i | X_{1:k-1}^i)}$.
 - (d) **delete** q^i .
 - (e) Set $X_{1:k}^i \leftarrow (X_{1:k-1}^i, X_k^i)$
 6. Compute $\widehat{Z}_{\pi_n} = \frac{1}{N} \sum_{i=1}^N W_n^i$.
-

some observed values y

$$\bar{\pi}(\theta | y) \propto p(y | \theta)p(\theta).$$

We assume that the data likelihood $p(y | \theta)$ can, by introducing a latent variable x , be computed as an integral

$$p(y | \theta) = \int p(y | x, \theta)p(x | \theta) dx.$$

Now, let our target distribution in Nested SIS be $\bar{\pi}_2(\theta, x) = \bar{\pi}_2(x | \theta)\bar{\pi}_1(\theta) = \frac{p(y | x, \theta)p(x | \theta)}{p(y | \theta)}p(\theta)$. We set our proposal distributions to be

$$\begin{aligned} \bar{q}_1(\theta) &= g_{\text{IS}}(\theta), \\ \bar{q}_2(x | \theta) &= \frac{p(y | x, \theta)p(x | \theta)}{p(y | \theta)}. \end{aligned}$$

First, $Q(q_1(\cdot), 1)$ runs an exact sampler from the proposal g_{IS} . Then at iteration $k = 2$ we let the nested procedure $Q(q_2(\cdot | \theta^i), M)$ be a standard IS algorithm with proposal $h(x | y, \theta)$, giving us properly weighted samples for q_2 . Putting all this together gives us samples θ^i distributed according to $g_{\text{IS}}(\theta)$ and weighted by

$$W_2^i \propto \frac{p(\theta^i)}{g_{\text{IS}}(\theta^i)} \cdot \frac{p(y | x^i, \theta^i)p(x^i | \theta^i) \frac{1}{M} \sum_{\ell=1}^M \frac{p(y | x^\ell, \theta^i)p(x^\ell | \theta^i)}{h(x^\ell | y, \theta^i)}}{p(y | x^i, \theta^i)p(x^i | \theta^i)} = \frac{\widehat{p}_M(y | \theta^i)p(\theta^i)}{g_{\text{IS}}(\theta^i)}, \quad (17)$$

where $\widehat{p}_M(y|\theta^i) = M^{-1} \sum_{\ell=1}^M \frac{p(y|x^\ell, \theta^i)p(x^\ell|\theta^i)}{h(x^\ell|y, \theta^i)}$. Thus we obtain a Nested SIS method that is identical to the IS² algorithm proposed by Tran et al. (2013).

A.3 Further Details on the Experiments

We provide some further details and results for the experiments presented in the main manuscript.

A.3.1 GAUSSIAN STATE SPACE MODEL

We generate data from a synthetic d -dimensional ($\dim(x_k) = d$) dynamical/spatio-temporal³ model defined by

$$\begin{aligned} x_k | x_{k-1} &\sim \mathcal{N}(x_k; \mu_k(x_{k-1}), \Sigma), \\ y_k | x_k &\sim \mathcal{N}(y_k; x_k, \tau_\phi^{-1}I), \end{aligned}$$

where Σ and μ_k are given as follows

$$\Sigma = \begin{pmatrix} \tau_\rho + \tau_\psi & -\tau_\psi & 0 & \cdots & \cdots & 0 & 0 \\ -\tau_\psi & \tau_\rho + 2\tau_\psi & -\tau_\psi & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & -\tau_\psi & 0 \\ 0 & 0 & 0 & 0 & -\tau_\psi & \tau_\rho + 2\tau_\psi & -\tau_\psi \\ 0 & 0 & 0 & 0 & 0 & -\tau_\psi & \tau_\rho + \tau_\psi \end{pmatrix}^{-1},$$

$$\mu_k(x_{k-1}) = a\tau_\rho \Sigma x_{k-1}.$$

Alternatively, in a more standard state space model notation, we have

$$\begin{aligned} x_k &= Ax_{k-1} + v_k, \quad v_k \sim \mathcal{N}(0, Q), \\ y_k &= x_k + e_k, \quad e_k \sim \mathcal{N}(0, R), \end{aligned}$$

where $A = a\tau_\rho \Sigma$, $Q = \Sigma$ and $R = \tau_\phi^{-1}I$. We assume that the parameters $\theta = (\tau_\psi, a, \tau_\rho, \tau_\phi) = (1, 0.5, 1, 10)$ are known.

To do inference with this generated data-set $\{y_k\}$ we propose to target the following slightly different model

$$p(x_{1:k}, y_{1:k}) \propto \prod_{j=1}^k \phi(x_j, y_j) \rho(x_j) \psi(x_j, x_{j-1}),$$

3. Note that in a previous version this was erroneously stated as equivalent to the Gaussian MRF we use for sequential inference. Thus this example actually illustrates a problem where we have a misspecified model. However, this misspecification does not lead to any discernible difference in the MSE results. This because the exact filtering marginals for the two different models (LGSS, GMRF) with the parameters chosen differs with orders of magnitudes much lower than the Monte Carlo errors.

where the observation potential ϕ and interaction potentials ρ and ψ are given by

$$\begin{aligned}\phi(x_k, y_k) &= \prod_{l=1}^d \phi_l(x_{k,l}, y_{k,l}) = \prod_{l=1}^d e^{-\frac{\tau_\phi}{2}(x_{k,l}-y_{k,l})^2}, \\ \psi(x_k) &= \prod_{l=2}^d \psi_l(x_{k,l}, x_{k,l-1}) = \prod_{l=2}^d e^{-\frac{\tau_\psi}{2}(x_{k,l}-x_{k,l-1})^2}, \\ \rho(x_k, x_{k-1}) &= \prod_{l=1}^d \rho_l(x_{k,l}, x_{k-1,l}) = \prod_{l=1}^d e^{-\frac{\tau_\rho}{2}(x_{k,l}-ax_{k-1,l})^2}.\end{aligned}$$

This can be visualised as a Gaussian rectangular ($d \times k$) lattice MRF, i.e. it grows with “time” k . The goal is to estimate the filtering distribution $p(x_k | y_{1:k})$. Note that this model has almost identical filtering marginals as the data generating distribution and leads to a simpler implementation of NSMC and ST-PF.

Results (mean-squared-error, MSE) comparing NSMC and ST-PF for different settings of N and M can be found in the first row of Figure 6 and the second row displays the results when comparing ST-PF to the SMC method by Naesseth et al. (2014b) for equal computational budgets. We show median (over dimensions d) MSE for posterior marginal mean and variance estimates of the respective algorithms. True values are obtained using belief propagation. Note that setting $N = 1$ in ST-PF can be viewed as a special case of the SMC method by Naesseth et al. (2014b).

A.3.2 SPATIO-TEMPORAL MODEL – DROUGHT DETECTION

We present the full model for drought detection in our notation, this is essentially the model by Fu et al. (2012) adapted for estimating the filtering distribution. The latent variables for each location on a finite world grid, $x_{k,i,j}$, are binary, i.e. 0 being normal state and 1 being the abnormal (drought) state. Measurements, $y_{k,i,j}$, are available as real valued precipitation values in millimeters. The probabilistic model for filtering is given as,

$$p(x_{1:k}, y_{1:k}) \propto \prod_{n=1}^k \phi(x_n, y_n) \rho(x_n) \psi(x_n, x_{n-1}), \quad (18a)$$

where

$$\phi(x_k, y_k) = \prod_{i=1}^I \prod_{j=1}^J \exp \left\{ -\frac{1}{2\sigma_{i,j}^2} (y_{k,i,j} - \mu_{ab,i,j}x_{k,i,j} - \mu_{norm,i,j}(1 - x_{k,i,j}))^2 \right\}, \quad (18b)$$

$$\rho(x_k) = \prod_{i=1}^I \prod_{j=1}^J \exp \{ C_1 (\mathbb{1}_{x_{k,i,j}=x_{k,i,j-1}} + \mathbb{1}_{x_{k,i,j}=x_{k,i-1,j}}) \}, \quad (18c)$$

$$\psi(x_k, x_{k-1}) = \prod_{i=1}^I \prod_{j=1}^J \exp \{ C_2 \mathbb{1}_{x_{k,i,j}=x_{k-1,i,j}} \}. \quad (18d)$$

Here, $\mathbb{1}$ is the indicator function, and with the convention that all expressions in (18c) that end up with index 0 evaluate to 0. The parameters C_1, C_2 are set to 0.5, 3 as in (Fu et al., 2012). Location

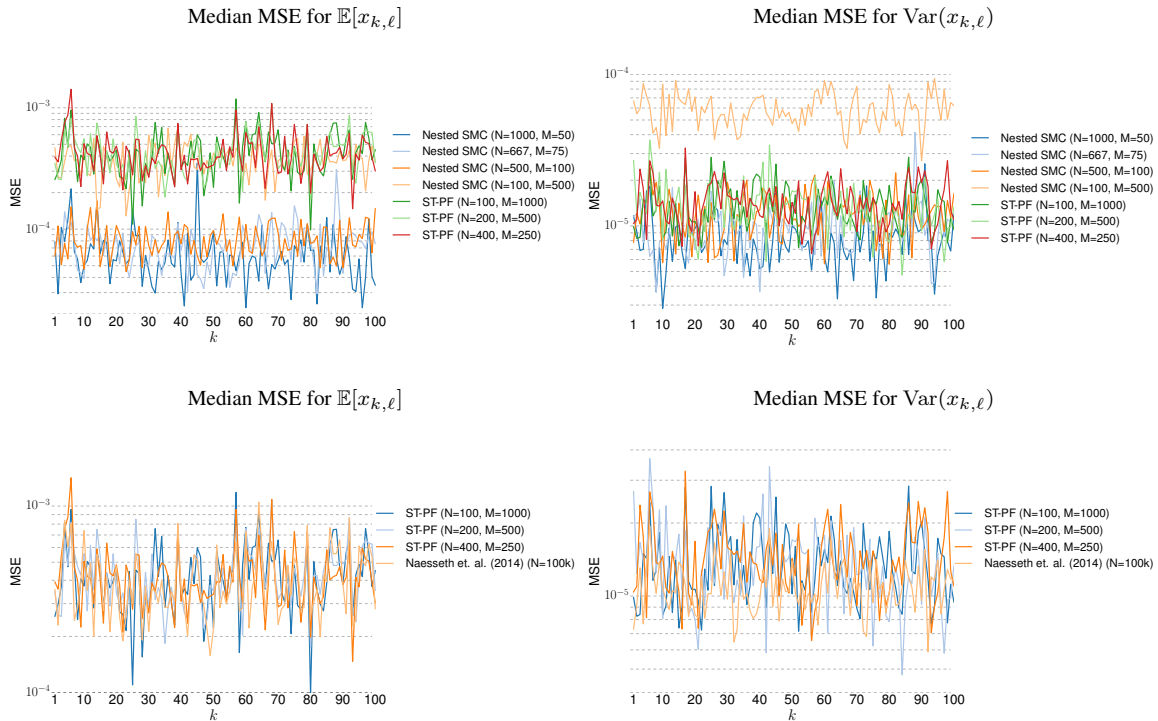


Figure 6: *Top*: Comparisons for different settings of N and M on the 50-dimensional SSM. *Bottom*: Illustrating the connection between ST-PF and the SMC method by Naesseth et al. (2014b).

based parameters $\sigma_{i,j}$, $\mu_{ab,i,j}$, $\mu_{\text{norm},i,j}$ are estimated based on data from the CRU dataset with world precipitation data from years 1901 – 2012. For the North America region we consider a 20×30 region with latitude $35 - 55^\circ N$ and longitude $90 - 120^\circ W$. For the Sahel region we consider a 24×44 region with latitude $6 - 30^\circ N$ and longitude $10^\circ W - 35^\circ E$. Note that for a few locations in Africa (Sahel region) the average yearly precipitation was constant. For these locations we simply set $\mu_{\text{norm},i,j}$ to be this value, $\mu_{ab,i,j} = 0$ and $\sigma_{i,j}^2$ to be the mean variance of all locations, thus this might have introduced some artifacts. Some representative results for the Sahel region are displayed in Figure 7.

References

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342, 2010.

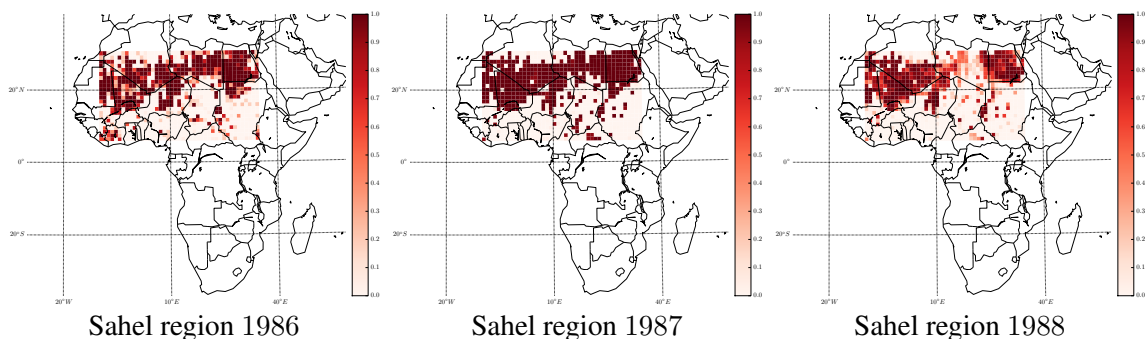


Figure 7: Estimate of $\mathbb{P}(X_{k,i,j} = 1 \mid y_{1:k})$ for all sites over a span of 3 years. All results for $N = 100$, $N_1 = \{30, 40\}$, $N_2 = 20$.

A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou. A stable particle filter in high-dimensions. *ArXiv:1412.3501*, December 2014a.

Alexandros Beskos, Dan Crisan, and Ajay Jasra. On the stability of sequential Monte Carlo methods in high dimensions. *Ann. Appl. Probab.*, 24(4):1396–1445, 08 2014b.

Peter Bickel, Bo Li, and Thomas Bengtsson. *Sharp failure rates for the bootstrap particle filter in high dimensions*, volume Volume 3 of *Collections*, pages 318–329. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2008.

Jonathan Briggs, Michael Dowd, and Renate Meyer. Data assimilation for large-scale spatio-temporal systems using a location particle smoother. *Environmetrics*, 24(2):81–97, 2013.

Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387402640.

J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings Radar, Sonar and Navigation*, 146(1):2–7, 1999.

Tianshi Chen, Thomas B. Schön, Henrik Ohlsson, and Lennart Ljung. Decentralized particle filter with arbitrary state decomposition. *IEEE Transactions on Signal Processing*, 59(2):465–478, Feb 2011.

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

N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. SMC2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426, 2013.

Jacques Cohen. Bioinformatics an introduction for computer scientists. *ACM Computing Surveys (CSUR)*, 36(2):122–158, 2004.

N. Cressie and C. K. Wikle. *Statistics for spatio-temporal data*. Wiley, 2011.

- D. Crisan and J. Míguez. Nested particle filters for online parameter estimation in discrete-time state-space Markov models. *ArXiv:1308.1883*, August 2013.
- P. Del Moral. *Feynman-Kac Formulae - Genealogical and Interacting Particle Systems with Applications*. Probability and its Applications. Springer, 2004.
- Petar M Djuric and Mónica F Bugallo. Particle filtering for high-dimensional systems. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*, pages 352–355. IEEE, 2013.
- R. Douc and E. Moulines. Limit theorems for weighted samples with applications to sequential Monte Carlo. *The Annals of Statistics*, 36(5):2344–2376, 2008.
- R. Douc, E. Moulines, and J. Olsson. Optimality of the auxiliary particle filter. *Probability and Mathematical Statistics*, 29:1–28, 2009.
- A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *Nonlinear Filtering Handbook*. Oxford University Press, 2011.
- Arnaud Doucet, Nando De Freitas, and Neil Gordon. *An introduction to sequential Monte Carlo methods*. Springer, 2001.
- Paul Fearnhead, Omiros Papaspiliopoulos, Gareth O. Roberts, and Andrew Stuart. Random-weight particle filtering of continuous time processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):497–512, 2010a.
- Paul Fearnhead, David Wyncoll, and Jonathan Tawn. A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2):447–464, 2010b.
- J. A. Foley, M. T. Coe, M. Scheffer, and G. Wang. Regime shifts in the sahara and sahel: Interactions between ecological and climatic systems in northern africa. *Ecosystems*, 6:524–539, 2003.
- Qiang Fu, Arindam Banerjee, Stefan Liess, and Peter K. Snyder. Drought detection of the last century: An MRF-based approach. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 24–34, Anaheim, CA, USA, April 2012.
- S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168, March 2004.
- N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, April 1993.
- M. Hoerling, J. Hurrell, J. Eischeid, and A. Phillips. Detection and attribution of twentieth-century northern and southern african rainfall change. *Journal of Climate*, 19:3989–4008, 2006.
- A. M. Johansen and A. Doucet. A note on auxiliary particle filters. *Statistics & Probability Letters*, 78(12):1498–1504, 2008.

- A. M. Johansen, N. Whiteley, and A. Doucet. Exact approximation of Rao-Blackwellised particle filters. In *Proceedings of the 16th IFAC Symposium on System Identification (SYSID)*, pages 488–493, Brussels, Belgium, 2012.
- P.D. Jones and I. Harris. CRU TS3.21: Climatic research unit (CRU) time-series (ts) version 3.21 of high resolution gridded data of month-by-month variation in climate (jan. 1901- dec. 2012). NCAS British Atmospheric Data Centre, sep 2013. URL <http://dx.doi.org/10.5285/D0E1585D-3417-485F-87AE-4FCECF10A992>.
- R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.
- A. Kong, J. S. Liu, and W. H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.
- F. Lindsten and T. B. Schön. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143, 2013.
- F. Lindsten, M. I. Jordan, and T. B. Schön. Particle Gibbs with ancestor sampling. *Journal of Machine Learning Research*, 15:2145–2184, 2014.
- Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2001.
- Claire Monteleoni, Gavin A. Schmidt, Francis Alexander, Alexandru Niculescu-Mizil, Karsten Steinhaeuser, Michael Tippett, Arindam Banerjee, M. Benno Blumenthal, Jason E. Smerdon Auroop R. Ganguly, and Marco Tedesco. Climate informatics. In Ting Yu, Nitesh Chawla, and Simeon Simoff, editors, *Computational Intelligent Data Analysis for Sustainable Development*. Chapman and Hall/CRC, London, 2013.
- Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön. Capacity estimation of two-dimensional channels using sequential Monte Carlo. In *The 2014 IEEE Information Theory Workshop (ITW)*, pages 431–435, Nov 2014a.
- Christian A. Naesseth, Fredrik Lindsten, and Thomas B Schön. Sequential Monte Carlo for graphical models. In *Advances in Neural Information Processing Systems 27*, pages 1862–1870. Curran Associates, Inc., 2014b.
- Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.
- P. Rebeschini and R. van Handel. Can local particle filters beat the curse of dimensionality? *Ann. Appl. Probab. (to appear)*, 2015.
- H. Rue and L. Held. *Gaussian Markov Random Fields, Theory and Applications*. CDC Press, Boca Raton, FL, USA, 2005.
- S. D. Schubert, M. J. Suarez, P. J. Pegion, R. D. Koster, and J. T. Bacmeister. On the cause of the 1930s dust bowl. *Science*, 303:1855–1859, 2004.

- R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications – with R examples*. Springer Texts in Statistics. Springer, New York, USA, third edition, 2011.
- M.-N. Tran, M. Scharth, M. K. Pitt, and R. Kohn. Importance sampling squared for Bayesian inference in latent variable models. *ArXiv:1309.3339*, sep 2013.
- Christelle Vergé, Cyrille Dubarry, Pierre Del Moral, and Eric Moulines. On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, 25(2): 243–260, 2015.
- Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.
- N. Whiteley. Discussion on Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):306–307, 2010.
- C. K. Wikle. Modern perspectives on statistics for spatio-temporal data. *WIREs Computational Statistics*, 7(1):86–98, 2015.