High-Dimensional Filtering Using Nested Sequential Monte Carlo

Christian Andersson Naesseth, Fredrik Lindsten and Thomas B. Schon

The self-archived postprint version of this journal article is available at Linköping University Institutional Repository (DiVA):

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-159547>

N.B.: When citing this work, cite the original publication.

Andersson Naesseth, C., Lindsten, F., Schon, T. B., (2019), High-Dimensional Filtering Using Nested Sequential Monte Carlo, *IEEE Transactions on Signal Processing*, 67(16), 4177-4188.

<https://doi.org/10.1109/TSP.2019.2926035>

Original publication available at:

<https://doi.org/10.1109/TSP.2019.2926035>

Copyright: Institute of Electrical and Electronics Engineers (IEEE)

<http://www.ieee.org/index.html>

©2019 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

High-dimensional Filtering using Nested Sequential Monte Carlo

Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön, *Senior member, IEEE*

Abstract—Sequential Monte Carlo (SMC) methods comprise one of the most successful approaches to approximate Bayesian filtering. However, SMC without a good proposal distribution can perform poorly, in particular in high dimensions. We propose nested sequential Monte Carlo (NSMC), a methodology that generalizes the SMC framework by requiring only approximate, properly weighted, samples from the SMC proposal distribution, while still resulting in a correct SMC algorithm. This way we can compute an “exact approximation” of, e.g., the locally optimal proposal, and extend the class of models for which we can perform efficient inference using SMC. We show improved accuracy over other state-of-the-art methods on several spatio-temporal state space models.

Index Terms—particle filtering, spatio-temporal models, state space models, approximate Bayesian inference, backward simulation.

I. INTRODUCTION

INFERENCE in complex and high-dimensional statistical models is a very challenging problem that is ubiquitous in applications such as climate informatics [1], bioinformatics [2] and machine learning [3].

We are interested in *sequential* Bayesian inference in settings where we have a sequence of posterior distributions that we need to compute. Furthermore, we focus on settings where the model (or state variable) is high-dimensional, but where there are *local dependencies* among state variables. One example of the type of models we consider are so-called spatio-temporal models [4], [5], [6].

Sequential Monte Carlo (SMC) methods comprise one of the most successful methodologies for sequential Bayesian inference. However, SMC struggles in high dimensions [7] and these methods are rarely used for dimensions, say, higher than ten [8]. The purpose of the nested sequential Monte Carlo (NSMC) methodology is to push this limit, allowing for efficient approximate Bayesian inference in higher-dimensional state space models (SSMs). While the methodology is applicable to a wide range of different models, we focus on the spatio-temporal setting. We propose a class of spatio-temporal models based on a combination of Markov random fields and state space models. Inference in this model class is challenging, however, we show that the NSMC method is well suited to this challenge.

The basic strategy is to mimic the behavior of a so-called *fully adapted* (or locally optimal) SMC algorithm. Full adaptation can drastically improve the efficiency of SMC in high

dimensions [9]. 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. This enables us to use a second layer of SMC to simulate approximately from the proposal. The proper weighting condition ensures the validity of NSMC, thus providing a generalization of the family of SMC methods. Furthermore, the NSMC procedure itself generates properly weighted samples, meaning that the procedure can be *nested* to an arbitrary degree.

Related work

There has been much recent interest in using Monte Carlo methods as nested procedures of other Monte Carlo algorithms. The SMC² [10], IS² [11] and nested particle filter [12] algorithms are used for learning static parameters as well as latent variables. In these methods one SMC/IS method for the parameters is coupled with another for the latent variables. In [13] and [14] the state inference problem is addressed by splitting the state vector into two components and run coupled SMC samplers for these. Compared with the present work, these methods solve different problems and the “internal” SMC samplers are constructed differently—for approximate marginalization instead of simulation.

By viewing state inference as a sequential problem in the *components* of the state vector $\mathbf{x}_t$ we can make use of the method for general graphical models introduced in [15]. This method is combined with the island particle filter [16], and studied more closely in [17] under the name space-time particle filter (ST-PF). The ST-PF is related to NSMC, but does not generate an approximation of the fully adapted SMC. Another key distinction is that in the ST-PF each particle in the “outer” SMC sampler corresponds to a complete particle system, whereas for NSMC it will correspond to a hypothesis about the latent state $\mathbf{x}_t$ as in standard SMC. This leads to lower communication costs and better memory efficiency in distributed implementations. We have also found that NSMC typically outperforms ST-PF, even when run on a single machine with matched computing times (see Section IV).

The method proposed in [18] can be viewed as a special case of NSMC when the nested procedure to generate samples is given by IS with the proposal being the transition probability. Independent resampling PF (IR-PF) introduced in [19] generates samples in the same way as NSMC, with IS (instead of SMC) as the nested procedure. However, IR-PF uses a different weighting that requires both the outer and the inner number of particles to tend to infinity for

C. A. Naesseth and F. Lindsten are with the Division of Statistics and Machine Learning, Linköping University, Linköping, Sweden e-mail: christia.a.naesseth@liu.se

T. B. Schön is with Uppsala University.

Manuscript received December 12, 2018.

consistency. On the contrary, NSMC only requires that the number of particles at the outermost level tends to infinity for consistency (Section III-E). Furthermore, we provide results in the supplementary material showing that NSMC significantly outperforms IR-PF on an example studied in [19].

There are other SMC-related methods that have been introduced to tackle high-dimensional problems, e.g., the so-called block PF studied in [20], the location particle smoother [21], and various methods discussed in [22]. These methods are, however, all inconsistent because they are based on approximations that result in systematic errors. For the block PF it is possible to decrease the bias by using larger blocks [20], however we then need to solve increasingly difficult filtering problems for each block. Combining this method with NSMC is an therefore an interesting possibility for future work.

The concept of proper weighting (or random weights) is not new and has been used in the so-called random weights particle filter [23]. They require exact samples from a proposal but use a nested Monte Carlo method to unbiasedly estimate the importance weights. In [24] and [25] the authors study proper weighting as a means to perform *partial resampling*, i.e., only resample a subset of the particles at each time, or to design novel MCMC proposals. The authors introduce the concept of “unnormalized” proper weighting, which is essentially the same as proper weighting that was introduced and used to motivate NSMC in [26]. Furthermore, in [27] proper weighting and NSMC are used to solve an inference problem within statistical historical linguistics.

Another approach to solve the sequential inference problem is the sequential Markov chain Monte Carlo class of methods [28]. It was shown by [28] that the optimal sequential MCMC algorithm actually is equivalent to the fully adapted SMC.

This paper extends preliminary work in [26] with the ability to handle more expressive models, more informative central limit theorems and convergence proofs, as well as new experiments.

II. SEQUENTIAL PROBABILISTIC MODELS

Before presenting the new inference methodology in Section III we present two classes of sequential probabilistic models that will be used to illustrate its applicability. First we review the *Markov random field* (MRF) model and show how this can be used in a spatio-temporal setting via a sequential model decomposition. We then propose a combination of the MRF and a *state space model* (SSM), resulting in a more explicit separation of the spatial and temporal dependencies of the model. These models will serve to illustrate the usefulness and wide applicability of the method we propose. Note, however, that NSMC is by no means restricted to the classes of models we illustrate in this section. It can in principle be applied to any sequence of distributions we would like to approximate. We will refer to this sequence of distributions of interest as the *target distributions*.

A. Markov random fields

The Markov random field is a type of undirected probabilistic graphical model [29]. The MRF is typically not represented

as a sequence of distributions (or models), but it has previously been shown [30], [31], [32], [15], [26], [33], [34] that it can be very useful to artificially introduce a sequence to simplify the inference problem. Furthermore, it is also possible to postulate the model as an MRF that increases with “time”, useful in e.g., climate science [35].

Consider first a standard MRF model of a multivariate random variable $\mathbf{x} = (x_1, \dots, x_{n_x})$, where $\mathbf{x} \in \mathbf{X}$. The conditional independencies among the model variables are described by the structure of the graph $G = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, \dots, n_x\}$ is the vertex set and $\mathcal{E} = \{(i, j) : (i, j) \in \mathcal{V} \times \mathcal{V}, \exists \text{ edge between vertex } i \text{ and } j\}$ is the edge set. Given G we can define a joint probability density function (PDF) for $\mathbf{x}$ that incorporates this structure as

$$\pi(\mathbf{x}) = \frac{1}{Z} \prod_{i \in \mathcal{V}} \phi(x_i, y_i) \prod_{(i, j) \in \mathcal{E}} \psi(x_i, x_j), \quad (1)$$

where $\mathbf{y} = (y_1, \dots, y_{n_x})$ is the observed variable and ϕ, ψ are called observation and interaction potentials, respectively. The normalization constant ensuring that $\pi(\cdot)$ integrates to one is given by

$$Z := \int \prod_{i \in \mathcal{V}} \phi(x_i, y_i) \prod_{(i, j) \in \mathcal{E}} \psi(x_i, x_j) d\mathbf{x}.$$

Note that (1) is usually referred to as a pairwise MRF in the literature due to $\pi(\cdot)$ factorizing into potentials that only depend on pairs of components of the random variable $\mathbf{x}$. For clarity we restrict ourselves to this type, however the method we propose in this paper can be applied to more general types of graphs, see e.g., [15] for ideas on how to extend SMC inference to non-pairwise MRFs.

Now, a sequential MRF is obtained if we consider a sequence of random variables $\mathbf{x}_{1:t} = (\mathbf{x}_1, \dots, \mathbf{x}_t)$, with $\mathbf{x}_{1:t} \in \mathbf{X}^t$ for $t = 1, \dots, T$, with a PDF that factorizes according to

$$\pi_t(\mathbf{x}_{1:t}) = \frac{1}{Z_t} \gamma_t(\mathbf{x}_{1:t}) := \frac{1}{Z_t} \gamma_{t-1}(\mathbf{x}_{1:t-1}) \cdot \prod_{i \in \mathcal{V}} \phi(x_{t,i}, y_{t,i}) \rho(\mathbf{x}_{t-1}, x_{t,i}) \prod_{(i, j) \in \mathcal{E}} \psi(x_{t,i}, x_{t,j}), \quad (2)$$

where $G = \{\mathcal{V}, \mathcal{E}\}$ again encodes the structure of the graphical model and $\rho(\cdot)$ is a new type of interaction potential that links $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$. Furthermore, the normalization constant is given by $Z_t := \int \gamma_t(\mathbf{x}_{1:t}) d\mathbf{x}_{1:t}$. We illustrate a typical example of a sequential MRF in Figure 1. As an example, this type of model was used by [35] in a spatio-temporal application to detect drought based on annual average precipitation rates collected from various sites in North America and Africa over the last century.

We would like to remark on one peculiarity that arises when the sequential MRF is used to model a spatio-temporal process. Consider $\pi_t(\cdot)$ without measurements as a prior on a spatio-temporal latent process, i.e., (2) where the potentials ϕ do not depend on $\mathbf{y}_t$. In this case we get that the marginals for $t < T$ change depending on the value of T , i.e., in general $\pi_t(\mathbf{x}_{1:t}) \neq \pi_T(\mathbf{x}_{1:t}) = \int \pi_T(\mathbf{x}_{1:T}) d\mathbf{x}_{t+1:T}$. Typically

Fig. 1: Illustration of a sequential MRF where G is given by a 2×3 grid with nearest neighbor interaction.

we would expect that *a priori* what happens for a dynamical process at time t should not be affected by the length of the time-series we consider. The next class of models we consider can introduce dependencies in both time and space without giving rise to this counter-intuitive result.

B. Spatio-temporal state space models

Before we move on to define the spatio-temporal state space model (ST-SSM), we will briefly review SSMs, a comprehensive and important model class commonly used for studying dynamical systems. For a more detailed account, and with pointers to the wide range of applications, we refer the readers to [36], [37], [38].

In state space models the sequential structure typically enters as a known, or postulated, dynamics on the unobserved latent state $\mathbf{x}_t$ that is then partially observed through the measurements $\mathbf{y}_t$. A common definition for SSMs is through its functional form

$$\mathbf{x}_t = a(\mathbf{x}_{t-1}, \mathbf{v}_t), \quad \mathbf{v}_t \sim p_{\mathbf{v}}(\cdot), \quad (3a)$$

$$\mathbf{y}_t = c(\mathbf{x}_t, \mathbf{e}_t), \quad \mathbf{e}_t \sim p_{\mathbf{e}}(\cdot), \quad (3b)$$

where $\mathbf{v}_t$ and $\mathbf{e}_t$ —often called process and measurement noise, respectively—are random variables with some given distributions $p_{\mathbf{v}}(\cdot)$ and $p_{\mathbf{e}}(\cdot)$. Furthermore, the initial state $\mathbf{x}_1$ is a random variable with some distribution $\mu(\cdot)$. Equation (3) can equivalently be stated through conditional distributions

$$\mathbf{x}_t | \mathbf{x}_{t-1} \sim f(\mathbf{x}_t | \mathbf{x}_{t-1}), \quad (4a)$$

$$\mathbf{y}_t | \mathbf{x}_t \sim g(\mathbf{y}_t | \mathbf{x}_t), \quad (4b)$$

and we define the sequential probabilistic model (or target distribution) as follows

$$\begin{aligned} \pi_t(\mathbf{x}_{1:t}) &:= \frac{\gamma_t(\mathbf{x}_{1:t})}{Z_t} \\ &= \frac{1}{Z_t} \mu(\mathbf{x}_1) g(\mathbf{y}_1 | \mathbf{x}_1) \prod_{s=2}^t f(\mathbf{x}_s | \mathbf{x}_{s-1}) g(\mathbf{y}_s | \mathbf{x}_s). \end{aligned} \quad (5)$$

We will assume that $g(\mathbf{y}_t | \mathbf{x}_t)$ can be evaluated pointwise. This condition is satisfied in many practical applications.

A typical assumption when using the SSM to model spatio-temporal systems is to introduce the spatial dependency only between time steps $t-1$ and t , see e.g., [39]. This can be achieved by defining the model such that the product of the induced distributions $f(\mathbf{x}_t | \mathbf{x}_{t-1}) g(\mathbf{y}_t | \mathbf{x}_t)$, conditionally on $\mathbf{x}_{t-1}$, completely factorize over the components of $\mathbf{x}_t$; see also [20] where SMC applied to such a model is studied. However, we argue that this approach can be limiting since any spatial dependencies are only implicitly taken into account via the temporal dynamics—indeed, conditionally on $\mathbf{x}_{t-1}$ the spatial components of $\mathbf{x}_t$ are assumed independent. To handle this we propose a generalized spatio-temporal model where spatial dependencies *within* each time step are introduced through the disturbance term $\mathbf{v}_t$. We define the ST-SSM as a combination of the functional and PDF representation of an SSM where the distribution for $\mathbf{v}_t$ is given by an MRF as in (1)

$$\mathbf{x}_t = \begin{pmatrix} x_{t,1} \\ \vdots \\ x_{t,n_x} \end{pmatrix} = \begin{pmatrix} a_1(\mathbf{x}_{t-1}, v_{t,1}) \\ \vdots \\ a_{n_x}(\mathbf{x}_{t-1}, v_{t,n_x}) \end{pmatrix},$$

$$\mathbf{v}_t \sim \frac{1}{Z_{\mathbf{v}}} \prod_{i \in \mathcal{V}} \phi(v_{t,i}) \prod_{(i,j) \in \mathcal{E}} \psi(v_{t,i}, v_{t,j}), \quad (6a)$$

$$\mathbf{y}_t | \mathbf{x}_t \sim g(\mathbf{y}_t | \mathbf{x}_t). \quad (6b)$$

We make no assumptions on local dependencies between $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$, however, to keep it simple we will assume that the graph $G = \{\mathcal{V}, \mathcal{E}\}$ describing the distribution for $\mathbf{v}_t$ does not depend on time t . Furthermore, we will in this paper mainly consider models where dependencies between components in $\mathbf{v}_t$ are “few”, i.e., the MRF is sparse with few elements in $\mathcal{E}$, and where components of $\mathbf{y}_t$ in $g(\cdot)$ only depend on subsets of $\mathbf{x}_t$. To illustrate the dependence structure in an ST-SSM we propose a combination of the traditional undirected graph for the MRF and the directed acyclic graph for the SSM, see Figure 2. This allows us to model more complex dynamical

Fig. 2: Illustration of a spatio-temporal state space model with $n_x = 4$, one conditionally independent observation per component in $\mathbf{x}_t$, and the MRF for $\mathbf{v}_t$ is given by a chain. Grey circles illustrate the observations $\mathbf{y}_t = (y_{t,1}, \dots, y_{t,4})$.

processes than [26] who assumed that $f(\mathbf{x}_t | \mathbf{x}_{t-1}) g(\mathbf{y}_t | \mathbf{x}_t)$ factorized with only local dependencies between components of $\mathbf{x}_t$. Furthermore, we can clearly see that the peculiarity discussed in Section II-A is not present in this model; the marginal of the prior does not change with T as expected.

III. METHODOLOGY

Inference in sequential probabilistic models boils down to computing the target distribution $\pi_t(\mathbf{x}_{1:t})$ for $t = 1, 2, \dots$;

typically an intractable problem with no analytical or simple numerical solution. This means that we have to resort to approximations. In this paper we focus on one particularly successful solution to the problem, the so called sequential Monte Carlo family of algorithms first introduced in [40], [41], [42].

The basic idea with SMC is to move a set of weighted samples (particles) $\{(\mathbf{x}_{1:t-1}^i, w_{t-1}^i)\}_{i=1}^N$ approximating π_{t-1} , to a new set of particles $\{(\mathbf{x}_{1:t}^i, w_t^i)\}_{i=1}^N$ which approximate π_t . These samples define an empirical approximation of the target distribution

$$\pi_t^N(d\mathbf{x}_{1:t}) := \sum_{i=1}^N \frac{w_t^i}{\sum_{\ell} w_t^\ell} \delta_{\mathbf{x}_{1:t}^i}(d\mathbf{x}_{1:t}), \quad (7)$$

where $\delta_{\mathbf{x}}(d\mathbf{x})$ is a Dirac measure at $\mathbf{x}$. In the next section we will detail a particularly efficient way of moving the particles, known as fully adapted SMC [43], [44], [45], ensuring that all normalized weights are equal to $\frac{1}{N}$.

A. Fully Adapted Sequential Monte Carlo

The procedure to move the particles and their weights from time $t-1$ to t in any SMC sampler is typically done in three stages. The first, *resampling*, stochastically chooses N particles at time $t-1$ that seem promising, discarding low-weighted ones. The second stage, *propagation*, generates new samples for time t conditioned on the resampled particles. The final stage, *weighting*, corrects for the discrepancy between the target distribution and the *proposal*, i.e., the instrumental distribution used in the propagation step.

Fully adapted SMC [43] makes specific choices on the resampling weights, ν_{t-1} , and the proposal, $q_t(\mathbf{x}_t|\mathbf{x}_{1:t-1})$, such that all the importance weights w_t are equal. By introducing *ancestor indices* $a_{t-1}^i \in \{1, \dots, N\}$ for $i = 1, \dots, N$, we can describe the resampling step (of the fully adapted SMC sampler) by simulating $\{a_{t-1}^i\}_{i=1}^N$ conditionally independently with

$$\mathbb{P}(a_{t-1}^i = j) = \frac{\nu_{t-1}^j}{\sum_{\ell=1}^M \nu_{t-1}^\ell}, \quad \nu_{t-1}^j := \int \frac{\gamma_t((\mathbf{x}_{1:t-1}^j, \mathbf{x}_t))}{\gamma_{t-1}(\mathbf{x}_{1:t-1}^j)} d\mathbf{x}_t. \quad (8)$$

Propagation then follows by simulating $\mathbf{x}_t^i$ conditionally on its ancestor $\mathbf{x}_{1:t-1}^{a_{t-1}^i}$, for $i = 1, \dots, N$, according to

$$\mathbf{x}_t^i | \mathbf{x}_{1:t-1}^{a_{t-1}^i} \sim q_t(\mathbf{x}_t | \mathbf{x}_{1:t-1}^{a_{t-1}^i}) := \frac{1}{\nu_{t-1}^{a_{t-1}^i}} \frac{\gamma_t((\mathbf{x}_{1:t-1}^{a_{t-1}^i}, \mathbf{x}_t))}{\gamma_{t-1}(\mathbf{x}_{1:t-1}^{a_{t-1}^i})}, \quad (9)$$

$$\mathbf{x}_{1:t}^i = \left(\mathbf{x}_{1:t-1}^{a_{t-1}^i}, \mathbf{x}_t^i \right).$$

This proposal is sometimes referred to as the (locally) *optimal proposal* because it minimizes incremental variances in the importance weights w_t^i . Weighting is easy since all weights are equal, i.e., the unnormalized weights are all set to $w_t^i = 1$. The fully adapted SMC sampler in fact corresponds to a locally optimal choice of both resampling weights and proposal with an incremental variance in the importance weights w_t^i that is zero.

Note that in most cases it is impossible to implement this algorithm exactly, since we can not calculate ν_{t-1} and/or simulate from q_t . Nested SMC solves this by *requiring only approximate resampling weights and approximate samples from q_t* , in the sense that is formalized in Section III-C. However, we will start by detailing some specific cases where we can efficiently implement exact fully adapted SMC. These cases are of interest in themselves, however, here we will mainly use them to build intuition for how the approximations in NSMC are constructed.

B. Leveraging Forward Filtering–Backward Simulation

The problems we need to solve are those of efficiently computing $\{\nu_{t-1}^i\}_{i=1}^N$ and simulating from q_t , defined in (8) and (9), respectively. There are at least two important special cases where we can use fully adapted SMC. The first is if the state space $\mathbf{X}$ is discrete and finite, i.e., $\mathbf{x}_t \in \{1, \dots, S\}^{n_x}, \forall t$. Even though exact algorithms are known in this case [36] the computational complexity scales quadratically with the cardinality of $\mathbf{x}_t$ (which is S^{n_x}). SMC methods can thus still be of interest [46], [32], [26]. The second case is if $\frac{\gamma_t(\mathbf{x}_{1:t})}{\gamma_{t-1}(\mathbf{x}_{1:t-1})}$ is an unnormalized Gaussian distribution, e.g., in the ST-SSM this would correspond to

$$\mathbf{x}_t = a(\mathbf{x}_{t-1}) + \mathbf{v}_t, \quad \mathbf{v}_t \sim \text{Gaussian MRF},$$

$$\mathbf{y}_t | \mathbf{x}_t \sim \mathcal{N}(\mathbf{y}_t; C\mathbf{x}_t, R),$$

for some matrix C , covariance matrix R , and an MRF in the components of $\mathbf{v}_t$ where all pair-wise potentials are Gaussian.

Now, even though in principle the fully adapted SMC is available in these special cases, the computational complexity can be prohibitive—it is of order $\mathcal{O}(S^{n_x})$ and $\mathcal{O}(n_x^3)$ for the finite state space and the Gaussian case, respectively. However, when there are local dependencies among state variables adhering to an underlying chain (or tree) structure it is possible to make use of efficient implementations with only $\mathcal{O}(S^2 n_x)$ and $\mathcal{O}(n_x)$ complexity, respectively. Such an algorithm was proposed in [32] for the finite state space case. This approach makes use of forward filtering–backward simulation (sampling) [47], [48] on the *components* of $\mathbf{x}_t$ to compute $\{\nu_{t-1}^i\}_{i=1}^N$ and sample q_t exactly. As an example, let us consider the above ST-SSM with $C = I$ and $R = I$ and a Gaussian MRF given by

$$p_{\mathbf{v}}(\mathbf{v}_t) = \frac{1}{Z_{\mathbf{v}}} \exp \left(-\frac{\tau}{2} \sum_{d=1}^{n_x} v_{t,d}^2 - \frac{\lambda}{2} \sum_{d=2}^{n_x} (v_{t,d} - v_{t,d-1})^2 \right),$$

for some positive constants τ and λ . For an SSM it is well known that the fully adapted SMC sampler corresponds to $q_t(\mathbf{x}_t | \mathbf{x}_{1:t-1}) = \frac{f(\mathbf{x}_t | \mathbf{x}_{t-1}) g(\mathbf{y}_t | \mathbf{x}_t)}{p(\mathbf{y}_t | \mathbf{x}_{t-1})}$ and $\nu_{t-1}^i = p(\mathbf{y}_t | \mathbf{x}_{t-1}^i)$. A computationally efficient way of computing $\{\nu_{t-1}^i\}_{i=1}^N$ and simulating from $q_t(\cdot | \mathbf{x}_{1:t-1}^i)$ is

$$\mathbf{x}_t = a(\mathbf{x}_{t-1}^i) + \mathbf{v}_t', \quad \mathbf{v}_t' \sim \frac{1}{\nu_{t-1}^i} g(\mathbf{y}_t | a(\mathbf{x}_{t-1}^i) + \mathbf{v}_t) p_{\mathbf{v}}(\mathbf{v}_t),$$

$$\nu_{t-1}^i = \int g(\mathbf{y}_t | a(\mathbf{x}_{t-1}^i) + \mathbf{v}_t) p_{\mathbf{v}}(\mathbf{v}_t) d\mathbf{v}_t.$$

Due to the structure in $p_v(\cdot)$ and $g(y_t|\mathbf{x}_t)$ the distribution to sample $\mathbf{v}'_t$ from corresponds to a Gaussian MRF with a chain-structure in the $v_{t,d}$'s (cf. Figure 2):

$$\begin{aligned} p(\mathbf{v}_t|\mathbf{y}_t, \mathbf{x}_{t-1}^i) &= \frac{g(\mathbf{y}_t|a(\mathbf{x}_{t-1}^i) + \mathbf{v}_t)p_v(\mathbf{v}_t)}{\prod_{d=1}^{n_x} p(y_{t,d}|y_{t,1:d-1}, \mathbf{x}_{t-1}^i)} \\ &\propto \exp\left(-\frac{1}{2} \sum_{d=1}^{n_x} [(y_{t,d} - a_d(\mathbf{x}_{t-1}^i) - v_{t,d})^2 + \tau v_{t,d}^2] \right. \\ &\quad \left. - \frac{\lambda}{2} \sum_{d=2}^{n_x} (v_{t,d} - v_{t,d-1})^2\right). \end{aligned} \quad (10)$$

Because of this structure we can efficiently compute the normalization constant of (10) by means of “forward” filtering over the components of the vector $\mathbf{v}_t$, keeping track of the incremental contributions to ν_{t-1}^j , $p(y_{t,d}|y_{t,1:d-1}, \mathbf{x}_{t-1}^i)$, $d = 1, \dots, n_x$. Sampling the distribution is then done by an explicit “backward” pass, simulating $v'_{t,d} \sim p(v_{t,d}|v'_{t,d+1:n_x}, y_{t,1:d}, \mathbf{x}_{t-1}^i)$, $d = n_x, n_x - 1, \dots, 1$. We provide an illustration of the process in Figure 3. See also [32] for an example of how this is done in practice for a finite state space.

C. Nested Sequential Monte Carlo

The idea behind nested SMC is to emulate the forward-backward-based implementation of the fully adapted SMC sampler detailed in the previous section for arbitrary sequential probabilistic models. Because computing $\{\nu_{t-1}^j\}_{j=1}^N$ and simulating from q_t exactly is intractable in general we propose to run an SMC-based forward filtering–backward simulation method [49], [50] on the components of $\mathbf{x}_t$ (or $\mathbf{v}_t$) to approximate $\{\nu_{t-1}^j\}_{j=1}^N$ and generate approximate draws from q_t . As we shall see below, this can be viewed as an “exact approximation” (see [51] for an explanation of the concept) of a fully adapted SMC algorithm.

To describe the NSMC method we will start from a formal presentation of the procedure and the basic conditions needed for its validity. The description is based on the introduction of a generic auxiliary variable, here denoted by $\mathbf{u}_{t-1} \in \mathcal{U}_{t-1}$, for each time $t \geq 1$. Recall that $\{\pi_t(\mathbf{x}_{1:t}), t \geq 1\}$ with $\pi_t(\mathbf{x}_{1:t}) \propto \gamma_t(\mathbf{x}_{1:t})$ is a sequence of target distributions for the sampler. Let $\{\mathbf{x}_{1:t-1}^i\}_{i=1}^N$ denote an unweighted particle set ($w_{t-1} \equiv 1$) approximating π_{t-1} . Then, one step of the NSMC method, going from iteration $t-1$ to t , proceeds as follows: first, we sample conditionally independently the auxiliary variables $\mathbf{u}_{t-1}^i \sim \eta_{t-1}^M(\mathbf{u}_{t-1}|\mathbf{x}_{1:t-1}^i)$, where η_{t-1}^M is some distribution parameterized by M . In Section III-D we will discuss how to make use of an internal SMC sampler to define this distribution, in which case the parameter M will denote the number of particles in the internal sampler. More precisely, in this case, sampling the auxiliary variables corresponds to the forward filtering step (cf. Section III-B) where $\mathbf{u}_{t-1}$ denotes all the random variables (particles and weights) generated by the internal SMC sampler (with M particles) and η_{t-1}^M denotes the joint distribution of these variables (implicitly defined by the sampling procedure).

Next, we compute the resampling weights based on the auxiliary variables, $\nu_{t-1}^i = \tau_t(\mathbf{u}_{t-1}^i)$, where τ_t is some real-valued function satisfying the requirements

$$\begin{aligned} \int \tau_t(\mathbf{u}_{t-1}) \eta_{t-1}^M(\mathbf{u}_{t-1}|\mathbf{x}_{1:t-1}) d\mathbf{u}_{t-1} &= \frac{\int \gamma_t((\mathbf{x}_{1:t-1}, \mathbf{x}_t)) d\mathbf{x}_t}{\gamma_{t-1}(\mathbf{x}_{1:t-1})}, \\ \tau(\mathbf{u}_{t-1}) &\geq 0 \text{ a.s.} \end{aligned} \quad (11)$$

Note that in general τ_t will also depend on $\mathbf{x}_{1:t-1}$, but this has been suppressed for brevity. Below we will define τ_t as the normalization constant estimate at the final step of the internal SMC samplers and then the unbiasedness condition (11) is satisfied by known properties of SMC [52, Proposition 7.4.1]. Next, we resample the particles $\{\mathbf{x}_{1:t-1}^i\}_{i=1}^N$ jointly with the auxiliary variables $\{\mathbf{u}_{t-1}^i\}_{i=1}^N$, by simulating ancestor variables $\{a_t^i\}_{i=1}^N$ with

$$\mathbb{P}(a_t^i = j) = \frac{\nu_{t-1}^j}{\sum_{\ell} \nu_{t-1}^{\ell}}, \quad j = 1, \dots, N. \quad (12)$$

Note that this means that we formally resample the complete state of the internal SMC samplers, captured by the auxiliary variables $\{\mathbf{u}_{t-1}^i\}_{i=1}^N$. In practical implementations there is no need to save multiple copies of the same internal state.

Next, for propagation we generate samples $\mathbf{x}_t^i \sim \kappa_t^M(\mathbf{x}_t|\mathbf{u}_{t-1}^i)$ from some distribution κ_t^M satisfying

$$\begin{aligned} \int \tau_t(\mathbf{u}_{t-1}) \kappa_t^M(\mathbf{x}_t|\mathbf{u}_{t-1}) \eta_{t-1}^M(\mathbf{u}_{t-1}|\mathbf{x}_{1:t-1}) d\mathbf{u}_{t-1} \\ = \frac{\gamma_t(\mathbf{x}_{1:t})}{\gamma_{t-1}(\mathbf{x}_{1:t-1})}. \end{aligned} \quad (13)$$

The distribution κ_t^M can be realized by running (SMC-based) backward simulation, analogously to the procedure described in Section III-B. However, a simple straightforward alternative that also satisfies (13) is to sample from the corresponding empirical distribution induced by the internal SMC sampler. We discuss the construction of η_{t-1}^M , κ_t^M and τ_t further in the next section.

Finally, we set $\mathbf{x}_{1:t}^i = (\mathbf{x}_{1:t-1}^i, \mathbf{x}_t^i)$, $i = 1, \dots, N$, and have thus obtained a new set of unweighted particles approximating π_t , i.e.,

$$\pi_t^N(d\mathbf{x}_{1:t}) := \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{x}_{1:t}^i}(d\mathbf{x}_{1:t}). \quad (14)$$

The auxiliary variables $\{\mathbf{u}_{t-1}^i\}_{i=1}^N$ can now be discarded.

Algorithm 1 Nested Sequential Monte Carlo (all for $i = 1, \dots, N$)

Require: $\eta_{t-1}^M, \kappa_t^M, \tau_t$ that generate samples properly weighted for $\frac{\gamma_t(\mathbf{x}_{1:t})}{\gamma_{t-1}(\mathbf{x}_{1:t-1})}$

- 1: **for** $t = 1$ to T **do**
- 2: Simulate $\mathbf{u}_{t-1}^i \sim \eta_{t-1}^M(\mathbf{u}_{t-1}|\mathbf{x}_{1:t-1}^i)$
- 3: Draw a_t^i with probability $\mathbb{P}(a_t^i = j) = \frac{\tau_t(\mathbf{u}_{t-1}^j)}{\sum_{\ell} \tau_t(\mathbf{u}_{t-1}^{\ell})}$
- 4: Simulate $\mathbf{x}_t^i \sim \kappa_t^M(\mathbf{x}_t|\mathbf{u}_{t-1}^{a_t^i})$
- 5: Set $\mathbf{x}_{1:t}^i = (\mathbf{x}_{1:t-1}^i, \mathbf{x}_t^i)$
- 6: **end for**

Fig. 3: Illustration of forward filtering for $\mathbf{v}_t$, we then obtain a sample from $\mathbf{v}'_t \sim p(\mathbf{v}_t | \mathbf{y}_t, \mathbf{x}_{t-1}^i)$ using backward sampling for $v_{t,4}, \dots, v_{t,1}$. Setting $\mathbf{x}_t = a(\mathbf{x}_{t-1}^i) + \mathbf{v}'_t$ yields a draw from the locally optimal proposal $q_t(\mathbf{x}_t | \mathbf{x}_{1:t-1}^i)$.

The two conditions on $\eta_{t-1}^M, \tau_t, \kappa_t^M$, i.e., (11) and (13), can in fact be replaced by the single condition that $(\mathbf{x}_t^i, \tau_t(\mathbf{u}_{t-1}^i))$ are *properly weighted* for $\frac{\gamma_t(\mathbf{x}_{1:t})}{\gamma_{t-1}(\mathbf{x}_{1:t-1})}$.

Definition 1. Let γ_{t-1} and γ_t be unnormalized densities on $\mathcal{X}^{t-1}$ and $\mathcal{X}^t$, respectively, and let $\mathbf{x}_{1:t-1}$ be in the support of γ_{t-1} . Let $(\mathbf{x}_t, \mathbf{u}_{t-1})$ be a random pair with distribution possibly depending on $\mathbf{x}_{1:t-1}$ and let $\tau_t : \mathcal{U}_{t-1} \rightarrow \mathbb{R}_+$. We say that $(\mathbf{x}_t, \tau_t(\mathbf{u}_{t-1}))$ are *properly weighted* for the (unnormalized) distribution $\frac{\gamma_t(\mathbf{x}_{1:t})}{\gamma_{t-1}(\mathbf{x}_{1:t-1})}$ if for all (suitably measurable) functions $h : \mathcal{X} \rightarrow \mathbb{R}$

$$\mathbb{E}[h(\mathbf{x}_t) \tau_t(\mathbf{u}_{t-1}) | \mathbf{x}_{1:t-1}] = C \int h(\mathbf{x}_t) \frac{\gamma_t(\mathbf{x}_{1:t})}{\gamma_{t-1}(\mathbf{x}_{1:t-1})} d\mathbf{x}_t, \quad (15)$$

for some positive constant $C > 0$ that is independent of the $\mathbf{x}$'s and $\mathbf{u}$'s.

We provide a summary of the proposed method in Algorithm 1. Although we focus on approximating the fully adapted SMC sampler, the extension to arbitrary resampling weights and proposal is straightforward, see the supplementary material. Next we will illustrate how we can make use of *nested* or *internal* SMC samplers to construct $\eta_{t-1}^M, \tau_t, \kappa_t^M$ that generate properly weighted samples.

D. Constructing η_{t-1}^M, τ_t and κ_t^M

To construct η_{t-1}^M we propose to run an SMC sampler targeting the components of $\mathbf{x}_t$ (or $\mathbf{v}_t$) one-by-one. Note that the internal SMC sampler is run with $\mathbf{x}_{1:t-1}$ fixed (to one of the N particles in the outer SMC sampler), and for notational simplicity we drop the dependence on $\mathbf{x}_{1:t-1}$ throughout this section. The internal SMC sampler is based on some sequence of (unnormalized) targets $p_d(x_{t,1:d})$ and proposals $r_d(x_{t,d} | x_{t,1:d-1})$, $d = 1, \dots, n_x$ such that $p_{n_x}(x_{t,1:n_x}) \propto \frac{\gamma_t(\mathbf{x}_{1:t})}{\gamma_{t-1}(\mathbf{x}_{1:t-1})}$. Note that $\mathbf{x}_t = x_{t,1:n_x}$. We provide a summary in Algorithm 2. In this case, the auxiliary variable $\mathbf{u}_{t-1}$ corresponds to all the random variables generated by the internal SMC sampler, i.e. $\mathbf{u}_{t-1} := \{x_{t,d}^{1:M}\}_{d=1}^{n_x} \cup \{a_{t,d}^{1:M}\}_{d=2}^{n_x}$. The function τ_t is naturally defined as the normalizing constant estimate for the internal SMC procedure, $\tau_t(\mathbf{u}_{t-1}) = \prod_{d=1}^{n_x} \frac{1}{M} \sum_{i=1}^M w_{t,d}^i$.

It remains to define the distribution κ_t^M , used to generate new particles $\mathbf{x}_t^i$ conditionally on the auxiliary variables. A first simple alternative is to simulate directly from the empirical measure defined by the approximation in Algorithm 2, leading to the following procedure.

Algorithm 2 Internal SMC (all for $i = 1, \dots, M$)

Require: Unnormalized target distributions $p_d(x_{t,1:d})$, proposals $r_d(x_{t,d} | x_{t,1:d-1})$, and M

- 1: $x_{t,1}^i \sim r_1(x_{t,1})$
- 2: Set $w_{t,1}^i = \frac{p_1(x_{t,1}^i)}{r_1(x_{t,1}^i)}$
- 3: **for** $d = 2$ to n_x **do**
- 4: Draw $a_{t,d}^i$ with probability $\mathbb{P}(a_{t,d}^i = j) = \frac{w_{t,d-1}^j}{\sum_{\ell} w_{t,d-1}^{\ell}}$
- 5: Simulate $x_{t,d}^i \sim r_d(x_{t,d} | x_{t,1:d-1}^{a_{t,d}^i})$
- 6: Set $x_{t,1:d}^i = (x_{t,1:d-1}^{a_{t,d}^i}, x_{t,d}^i)$
- 7: Set $w_{t,d}^i = \frac{p_d(x_{t,1:d}^i)}{p_{d-1}(x_{t,1:d-1}^{a_{t,d}^i}) r_d(x_{t,d}^i | x_{t,1:d-1}^{a_{t,d}^i})}$
- 8: **end for**

Definition 2 (Internal SMC-based procedure). Let η_{t-1}^M, τ_t , and κ_t^M be defined as follows for some sequence $\{p_d(\cdot)\}_{d=1}^{n_x}$ such that $p_{n_x}(x_{t,1:n_x}) \propto \frac{\gamma_t(\mathbf{x}_{1:t})}{\gamma_{t-1}(\mathbf{x}_{1:t-1})}$:

- 1) Simulate $\mathbf{u}_{t-1} \sim \eta_{t-1}^M(\mathbf{u}_{t-1} | \mathbf{x}_{1:t-1})$ by running Algorithm 2.
- 2) Set $\tau_t(\mathbf{u}_{t-1}) = \prod_{d=1}^{n_x} \frac{1}{M} \sum_{i=1}^M w_{t,d}^i$
- 3) Simulate $\mathbf{x}_t \sim \kappa_t^M(\mathbf{x}_t | \mathbf{u}_{t-1})$, where $\kappa_t^M(\mathbf{x}_t | \mathbf{u}_{t-1}) := \sum_{i=1}^M \frac{w_{t,d}^i}{\sum_{j=1}^M w_{t,d}^j} \delta_{x_{t,1:n_x}^i} (d\mathbf{x}_t)$.

However, although this approach will result in properly weighted samples (Proposition 1 below) it can introduce significant correlation between the samples. To mitigate this we propose to instead make use of backward simulation [49], [50] to construct a more efficient κ_t^M , see Algorithm 3. This mimics the exact backward-simulation-based implementation of the fully adapted SMC sampler discussed in Section III-B. Note that we only need to draw once from κ_t^M for each outer-level particle, implying that for models with local spatial dependencies there is a small constant ($\lesssim 2$) computational overhead in using backward sampling in place of simply sampling from the empirical measure of the forward filter. The NSMC procedure using internal backward simulation is defined as follows.

Definition 3 (Internal SMC-based procedure with backward simulation). Let η_{t-1}^M and τ_t be defined as in Definition 2, but define κ_t^M as:

- 3'. Simulate $\mathbf{x}_t \sim \kappa_t^M(\mathbf{x}_t | \mathbf{u}_{t-1})$ by running Algorithm 3

Algorithm 3 Internal Backward Simulation

Require: $\{(x_{t,1:d}^i, w_{t,d}^i)\}_{i=1}^M$, $d = 1, \dots, n_x$ approximating $p_d(x_{t,1:d})$

- 1: Draw b_{n_x} with probability $\mathbb{P}(b_{n_x} = j) = \frac{w_{t,n_x}^j}{\sum_{\ell} w_{t,n_x}^{\ell}}$
- 2: Set $x_{t,n_x} = x_{t,n_x}^{b_{n_x}}$
- 3: **for** $d = n_x - 1$ **to** **1 do**
- 4: Draw b_d with probability $\mathbb{P}(b_d = j) \propto w_{t,d}^j \frac{p_{n_x}((x_{t,1:d}^j, x_{t,d+1:n_x}))}{p_d(x_{t,1:d}^j)}$
- 5: Set $x_{t,d:n_x} = (x_{t,d}^{b_d}, x_{t,d+1:n_x})$
- 6: **end for**

Proposition 1 (Proper weighting). *The procedure in either Definition 2 or 3 generates $(\mathbf{x}_t, \tau_t(\mathbf{u}_{t-1}))$ that are properly weighted for $\frac{\gamma_t(\mathbf{x}_{1:t})}{\gamma_{t-1}(\mathbf{x}_{1:t-1})}$.*

Proof. The result follows from Theorem 2 in [26]. $\square$

We can think of this as an analogy to the example in Section III-B and Figure 3 where we used forward filtering-backward sampling by considering the components of $v_{t,1:d}$ as our target. Instead of exact forward filtering we can use Algorithm 2, and instead of exact backward sampling we can use Algorithm 3, to generate properly weighted samples.

E. Theoretical Justification

In this section we will provide a central limit theorem that further motivates NSMC, and show how the asymptotic variance depends on the internal approximation of the exact fully adapted SMC. Furthermore, we provide a result that shows how this asymptotic variance converges to that of the corresponding asymptotic variance of the exact fully adapted SMC method as $M \rightarrow \infty$. We define the shorthand $\pi(f) := \int f(x)\pi(dx)$ for a measure π and function f .

Theorem 1 (Central Limit Theorem). *Assume that η_{t-1}^M , τ_t , κ_t^M , generate properly weighted samples for $\gamma_t(\mathbf{x}_{1:t})/\gamma_{t-1}(\mathbf{x}_{1:t-1})$. Under certain (standard) regularity conditions, specified in the supplementary material, we have the following central limit theorem for any fixed M :*

$$\sqrt{N} \left(\frac{1}{N} \sum_{i=1}^N \varphi(\mathbf{x}_{1:t}^i) - \pi_t(\varphi) \right) \xrightarrow{d} \mathcal{N}(0, \Sigma_t^M(\varphi)),$$

where $\varphi : \mathbf{X}_t \rightarrow \mathbb{R}$, and the $\{\mathbf{x}_{1:t}^i\}_{i=1}^N$ are generated by Algorithm 1. The asymptotic variance is given by

$$\Sigma_t^M(\varphi) = \sum_{s=0}^t \sigma_{s,t}^M(\varphi),$$

for $\sigma_{s,t}^M(\varphi)$ defined by

$$\begin{aligned} \sigma_{t,t}^M(\varphi) &= \pi_t \left((\varphi - \pi_t(\varphi))^2 \right), \\ \sigma_{s,t}^M(\varphi) &= \int \Psi_{s,t}^M(\mathbf{x}_{1:s}; \varphi) \pi_s(\mathbf{x}_{1:s}) d\mathbf{x}_{1:s}, \quad \text{for } 0 < s < t, \\ \sigma_{0,t}^M(\varphi) &= \int \frac{\tau_1(\mathbf{u}_0)^2}{Z_1^2} \left(\int (\varphi(\mathbf{x}_{1:t}) - \pi_t(\varphi)) \cdot \frac{\pi_t(\mathbf{x}_{1:t})}{\pi_1(\mathbf{x}_1)} \kappa_1^M(\mathbf{x}_1 | \mathbf{u}_0) d\mathbf{x}_{1:t} \right)^2 \eta_0^M(\mathbf{u}_0) d\mathbf{u}_0. \end{aligned}$$

with

$$\begin{aligned} \Psi_{s,t}^M(\mathbf{x}_{1:s}; \varphi) &:= \mathbb{E}_{\eta_s^M(\mathbf{u}_s | \mathbf{x}_{1:s})} \left[\frac{Z_s^2}{Z_{s+1}^2} \tau_{s+1}(\mathbf{u}_s)^2 \right. \\ &\quad \left. \left(\int (\varphi(\mathbf{x}_{1:t}) - \pi_t(\varphi)) \frac{\pi_t(\mathbf{x}_{1:t})}{\pi_{s+1}(\mathbf{x}_{1:s+1})} \kappa_{s+1}^M(\mathbf{x}_{s+1} | \mathbf{u}_s) d\mathbf{x}_{s+1:t} \right)^2 \right] \end{aligned} \quad (16)$$

Proof. See the supplementary material. $\square$

This theorem shows that, even for a fixed and finite value of M , the NSMC method obtains the standard $\sqrt{N}$ convergence rate of regular SMC. We can see how the asymptotic variance depends on how well we approximate the proposal q_t and its normalization constant with κ_t^M and τ_t . Furthermore, this lets us study convergence of the variance in M (and also analytic expressions for a high-dimensional state space model; see the subsequent section).

To show the convergence to fully adapted SMC as the approximation improves with increasing M we make some further assumptions detailed below.

Assumption 1 (Uniform integrability). *The sequence (in M) of random variables $\{\Psi_{s,t}^M(\mathbf{x}_{1:s}; \varphi)\}$, where $\mathbf{x}_{1:s} \sim \pi_s$, is uniformly integrable.*

Note that a sufficient condition for Assumption 1 to hold is that for some $\delta > 0$ and for all $s, M \geq 1$ the following holds

$$\int \Psi_{s,t}^M(\mathbf{x}_{1:s}; \varphi)^{1+\delta} \pi_s(\mathbf{x}_{1:s}) d\mathbf{x}_{1:s} < \infty.$$

Assumption 2 (Strong mixing). *For all s, t , there exist constants $0 < \lambda_{s+1,t}^-, \lambda_{s+1,t}^+ < \infty$ such that*

$$\begin{aligned} \lambda_{s+1,t}^- \cdot \pi_t(\mathbf{x}_{s+2:t} | \mathbf{x}_{1:s+1}) &\leq \frac{\pi_t(\mathbf{x}_{1:t})}{\pi_{s+1}(\mathbf{x}_{1:s+1})} \\ &\leq \lambda_{s+1,t}^+ \cdot \pi_t(\mathbf{x}_{s+2:t} | \mathbf{x}_{1:s+1}). \end{aligned}$$

In the supplementary material we detail a weaker assumption for which Proposition 2 still holds.

Proposition 2. *Assume that the NSMC method uses the internal sampling procedure specified in Definition 2. Under the assumptions of Theorem 1, Assumption 1 and 2 the following limit holds:*

$$\begin{aligned} \lim_{M \rightarrow \infty} \Sigma_t^M(\varphi) &= \\ &= \pi_t \left((\varphi - \pi_t(\varphi))^2 \right) + \sum_{s=1}^{t-1} \int \frac{\pi_t(\mathbf{x}_{1:s})^2}{\pi_s(\mathbf{x}_{1:s})} \\ &\quad \cdot \left(\int \varphi(\mathbf{x}_{1:t}) \pi_t(\mathbf{x}_{s+1:t} | \mathbf{x}_{1:s}) d\mathbf{x}_{s+1:t} - \pi_t(\varphi) \right)^2 d\mathbf{x}_{1:s}. \end{aligned}$$

Proof. See the supplementary material. $\square$

This result shows that as the number of internal samples M goes to infinity, the asymptotic variance we obtain is exactly equal to what it would be if we would have simulated exactly from the proposal q_t and computed the corresponding ν_{t-1} exactly. This means that the limiting asymptotic variance of NSMC is exactly the asymptotic variance of the fully adapted SMC sampler [44]. For simplicity we state the result for the case without backward sampling in the internal sampling procedure (Definition 2). However, the same result is expected to hold with backward sampling as well (Definition 3).

To study the finite M behavior of NSMC we consider a fairly simple model and test function that leads to analytical expressions for the asymptotic variance in the CLT above. Specifically, we study a high-dimensional SSM, given in Definition 4, obtained by making n_x independent copies of an SSM. A similar model is considered in [53] for analyzing the stability of SMC samplers in high dimensions.

Definition 4. *The spatially independent state space model has target distribution*

$$\pi_t(\mathbf{x}_{1:t}) \propto \prod_{d=1}^{n_x} \left[\mu(x_{1,d}) \prod_{s=1}^t g(y_{s,d}|x_{s,d}) \prod_{s=2}^t f(x_{s,d}|x_{s-1,d}) \right].$$

For simplicity we also assume that $y_{s,d} = y_{s,e}, \forall d, e$ and that $\mathbb{E}_{\pi_t}[\mathbf{x}_t] = 0$.

For this model we can obtain explicit expressions for the asymptotic variances for the exact fully adapted SMC sampler as well as for the NSMC sampler.

Proposition 3. *For the model in Definition 4 and $\varphi(\mathbf{x}_{1:t}) = \sum_{d=1}^{n_x} x_{t,d}$, we have that the asymptotic variance of fully adapted SMC is given by*

$$\Sigma_t^{FA}(\varphi) = n_x A_t + \sum_{s=1}^{t-1} n_x B_s^{n_x-1} A_s + n_x(n_x-1) B_s^{n_x-2} C_s^2.$$

Furthermore, using $r(x_{s,d}|x_{s-1,d})$ as proposal in the inner SMC (Algorithm 2) of NSMC implemented according to Definition 3, we get that the asymptotic variance of NSMC is

$$\begin{aligned} \Sigma_t^M(\varphi) = & n_x A_t + \sum_{s=0}^{t-1} \left[n_x B_s^{n_x-1} \left(A_s + M^{-1} (\tilde{A}_s - A_s) \right) \right. \\ & \cdot \left(1 - \frac{1}{M} \right)^{n_x-1} \left(1 + \frac{\tilde{B}_s}{B_s(M-1)} \right)^{n_x-1} \\ & + n_x(n_x-1) B_s^{n_x-2} \left(C_s + M^{-1} (\tilde{C}_s - C_s) \right)^2 \\ & \cdot \left(1 - \frac{1}{M} \right)^{n_x-2} \left(1 + \frac{\tilde{B}_s}{B_s(M-1)} \right)^{n_x-2} \Big], \end{aligned}$$

for the (finite) positive constants $A_t, A_s, \tilde{A}_s, B_s, \tilde{B}_s, C_s$, and $\tilde{C}_s$ defined in the supplementary material.

Proof. See the supplementary material. $\square$

As expected the asymptotic variance of fully adapted SMC grows exponentially in the dimension n_x of the state. (Note that this result is asymptotic only in N , and not in n_x , in

contrast to the result obtained by [9].) However, to control the additional approximation introduced by NSMC, i.e., not evaluating ν_{t-1} and sampling q_t exactly, we only need to scale $M \propto n_x$, even as $n_x \rightarrow \infty$. This means that it is often possible to cheaply obtain an accurate approximation to the optimal proposal and weights.

We expect that intuition and rules-of-thumb from running standard SMC also apply to the internal approximation targeting $\gamma_t(\mathbf{x}_{1:t})/\gamma_{t-1}(\mathbf{x}_{1:t-1})$, rather than $\gamma_t(\mathbf{x}_{1:t})$. In Section IV-A we empirically study how the choice of M affects the accuracy.

F. Modularity and implementation aspects

The procedure described by Algorithms 1–3 describes a nested SMC procedure with two levels that can be used to sample from high-dimensional models such as the sequential MRF or the ST-SSM described in Section II. Since this procedure is based on using SMC on the components of each $\mathbf{x}_t$ -vector it will, intuitively, work best when the dependencies among these components have a chain (or chain-like) structure, even though this is not a formal requirement. However, the methodology described in this section can be generalized to an arbitrary number of nested SMC procedures, which could prove useful for models with more than one spatial dimension. Indeed, the nested SMC procedure itself produces properly weighted samples [26]. For instance, in an ST-SSM (6) where the MRF describing the noise distribution is a 2-dimensional lattice, we may consider a three-level nested SMC procedure: at the first level the sampler operates on the temporal dimension, the second level simulates complete “rows” of states of the 2d lattice, and at the third level we sample the individual components of each “row”. We investigate this three-level procedure numerically in Section IV-B.

An important aspect of this nesting of the proposed method is that it is completely modular, in the sense that the first level SMC does not need to be aware of the number or specific implementations of the consecutive levels. Indeed, as long as it has access to some procedure of generating properly weighted samples for $\gamma_t(\mathbf{x}_{1:t})/\gamma_{t-1}(\mathbf{x}_{1:t-1})$ it is possible to run Algorithm 1 without caring about *how* these samples are produced (whether we use one or several internal SMC samplers, e.g.).

Related to the modularity of the method it is worth noting that while Algorithm 1 describes the NSMC procedure in mathematical terms, this is not typically how one would like to implement the method in practice. Specifically, at line 2 of Algorithm 1 we simulate the auxiliary variables $\{\mathbf{u}_{t-1}^i\}_{i=1}^N$, which correspond to running N internal SMC samplers. This can be done in separate processes or, indeed, in a distributed environment on separate machines. Importantly, there is no need to return the auxiliary variables $\{\mathbf{u}_{t-1}^i\}_{i=1}^N$ to the “master process”, which would incur a high communication cost. Instead, we simply return the estimates of the normalizing constants $\{\nu_{t-1}^i\}_{i=1}^N$, which is sufficient to carry out the resampling on line 3. Next, for the propagation step on line 4 we request samples from the internal procedures; the backward simulation is run internally in these processes and the resulting samples are returned.

Fig. 4: MSE and error bars of Monte Carlo estimates of $\log p(\mathbf{y}_{1:T})$, $\mathbb{E}[x_{T,1}]$, $\mathbb{E}[x_{T,n_x}]$ for BPF, FAPF and two variants of NSMC. $N = 100$ for FAPF and NSMC and BPF has equivalent computational budget.

IV. NUMERICAL RESULTS

A. Gaussian Model

We start by considering a Gaussian spatio-temporal state space model where the exact solution is available via the Kalman filter [54], and we can implement exact fully adapted SMC as explained in Section III-B. The model is given by

$$\begin{aligned} \mathbf{x}_t &= 0.5\mathbf{x}_{t-1} + \mathbf{v}_t, \\ \mathbf{v}_t &\sim \frac{1}{Z_{\mathbf{v}}} \exp \left(-\frac{\tau}{2} \sum_{d=1}^{n_x} v_{t,d}^2 - \frac{\lambda}{2} \sum_{d=2}^{n_x} (v_{t,d} - v_{t,d-1})^2 \right) \end{aligned} \quad (17a)$$

$$\mathbf{y}_t | \mathbf{x}_t \sim \mathcal{N}(\mathbf{x}_t, \sigma_y^2 I). \quad (17b)$$

The results for $N = 100$, $T = 10$, $\tau = \lambda = 1$ and $\sigma_y^2 = 0.25^2$, i.e., with fairly high signal-to-noise ratio, are given in Figure 4. We compare NSMC with (and without) backward simulation to the bootstrap particle filter (BPF) that uses the transition probability as proposal. We give all methods equivalent computational budget as the number of internal particles M grow, i.e., BPF gets $N_{\text{BPF}} = 100 \cdot M$ particles. Furthermore, for illustrative purposes we include fully adapted SMC (FAPF), the method that NSMC approximates, for a fixed number of particles $N_{\text{FAPF}} = 100$. The experiments are run ten times independently and we show the mean squared error (MSE) as well as one standard deviation error bars, for estimates of the log-likelihood, $\mathbb{E}[x_{T,1}]$ and $\mathbb{E}[x_{T,n_x}]$ with $n_x \in \{10, 100\}$. The expectations are with respect to the posterior distribution.

We can see that NSMC is significantly better than BPF and that it converges quickly towards the fully adapted SMC. Backward simulation also clearly helps with estimates of $\mathbb{E}[x_{T,d}]$ for $d = 1$, alleviating the correlation between generated samples. It is worthwhile to point out that for small M the NSMC seems to improve much more quickly than the standard *asymptotic* rate M^{-1} . For the likelihood estimate the rate almost exceeds M^{-4} . We provide results for different settings of σ_y^2 in the supplementary material. In general we see less striking improvement of NSMC over BPF when the signal to noise ratio is low, i.e., σ_y^2 is high compared to τ^{-1} .

B. Soil Carbon Cycles

We move on to study the performance of NSMC and compare it to ST-PF [17] on a spatio-temporal model inspired by the soil carbon cycle model of [55], [56]. The simplified model that we use to profile the two state-of-the-art methods is defined by

$$\begin{aligned} \mathbf{x}_t &= 0.5(\mathbf{x}_{t-1} + e^{\xi_t})e^{\mathbf{v}_t}, \\ \mathbf{v}_t &\sim \frac{1}{Z_{\mathbf{v}}} \exp \left(-\frac{\tau}{2} \sum_{i \in \mathcal{V}} v_{t,i}^2 - \frac{\lambda}{2} \sum_{(i,j) \in \mathcal{E}} (v_{t,i} - v_{t,j})^2 \right), \end{aligned} \quad (18a)$$

$$\mathbf{y}_t | \mathbf{x}_t \sim \text{Truncated Normal}(\mathbf{x}_t, \sigma^2 I, 0, \infty), \quad (18b)$$

where ξ_t is a known input signal and $(\mathcal{V}, \mathcal{E})$ is a square lattice, $\sqrt{n_x} \times \sqrt{n_x}$, with nearest neighbour interaction, i.e., $(i, j) \in \mathcal{E}$ if i and j are neighbors on the lattice. The latent variables $\mathbf{x}_t$ are positive and it is not possible to implement the exact fully

adapted SMC method. We set $T = 25$, $n_x = 36$ (6×6), $\sigma = 0.2$, $\tau = 2$, and $\lambda = 1.0$ and run NSMC and ST-PF with matched computational costs. Figure 5 displays the mean, over the n_x dimensions, mean squared error for each time-point t estimated by running the algorithms 10 times independently. We tried different values for N and M for both ST-PF and NSMC, and found that $N = 500$ worked well for both methods. The three level NSMC has two sets of nested particles, M_1 and M_2 , which we set to $M_1 = M_2 = 30$. The number of nested particles M for the two level NSMC and ST-PF were chosen to match wall-clock-time.

We can see that the different NSMC versions perform better than ST-PF in terms of MSE. This is without taking into account that NSMC simplifies distribution of the computation and is more memory efficient, only N rather than NM samples need to be retained at each step.

V. CONCLUDING REMARKS

We introduced and developed the nested sequential Monte Carlo (NSMC) class of methods for inference in high-dimensional state space models (SSM). NSMC provides the practitioner with a flexible and powerful approximate Bayesian inference algorithm for their toolbox. We have shown how the methodology generalizes the SMC framework by requiring only approximate, properly weighted, samples from the SMC proposal distribution. This allows for an “exact approximation” of, e.g., the locally optimal proposal, which extends the class of models for which we can perform efficient inference using SMC. Furthermore, we have introduced the spatio-temporal state space model, a new probabilistic model that melds Markov random fields with SSMs.

We have shown numerically that NSMC can significantly increase the range of high-dimensional problems that can be addressed using SMC. However, it should be noted that the algorithm still suffers from the curse of dimensionality, in the sense that its errors are expected to grow exponentially with system dimension (but the constants are drastically reduced compared to, e.g., bootstrap SMC; see Section III-E). Indeed, this is in agreement with known results on the scaling of locally optimal SMC [9] and the fact that NSMC can be viewed as an “exact approximation” thereof.

This means that further developments are needed to address very high-dimensional systems. One promising approach is to combine NSMC with the spatially blocked PF studied in [20]. Controlling the bias of this method requires using large blocks, which could be enabled by using NSMC to filter each local block. Another interesting possibility is to use NSMC for *temporal blocking*, as studied by [57], [58], which also provides a handle to control the effect of dimensionality. We leave these possible combinations as future work.

Another interesting area for future work is to combine NSMC with methods such as SMC² [10] or PMCMC [51] for parameter learning. Using NSMC as a component of such algorithms we can effectively extend the range of models to which they can be applied.

Fig. 5: Mean, over components d , MSE of $x_{t,d}, x_{t,d}^2$ estimated by ST-PF, and NSMC with two or three levels of inner SMC.

ACKNOWLEDGMENT

This research was financially supported by the Swedish Research Council via the projects; *CADICS - Control, Autonomy and Decision-making In Complex Systems*, a Linnaeus Center, *Learning of Large-Scale Probabilistic Dynamical Models* (contract number: 2016-04278), and *NewLEADS - New Directions in Learning Dynamical Systems* (contract number: 621-2016-06079) and the Swedish Foundation for Strategic Research (SSF) via the projects *ASSEMBLE* (contract number: RIT15-0012) and *Probabilistic Modeling and Inference for Machine Learning* (contract number: ICA16-0015).

REFERENCES

- [1] C. Monteleoni, G. A. Schmidt, F. Alexander, A. Niculescu-Mizil, K. Steinhauser, M. Tippet, A. Banerjee, M. B. Blumenthal, J. E. S. Auroop R. Ganguly, and M. Tedesco, "Climate informatics," in *Computational Intelligent Data Analysis for Sustainable Development*. London: Chapman and Hall/CRC, 2013.
- [2] J. Cohen, "Bioinformatics—an introduction for computer scientists," *ACM Computing Surveys (CSUR)*, vol. 36, no. 2, pp. 122–158, 2004.
- [3] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.
- [4] C. K. Wikle, "Modern perspectives on statistics for spatio-temporal data," *WIREs Computational Statistics*, vol. 7, no. 1, pp. 86–98, 2015.
- [5] N. Cressie and C. K. Wikle, *Statistics for spatio-temporal data*. Wiley, 2011.
- [6] H. Rue and L. Held, *Gaussian Markov Random Fields, Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2005.
- [7] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to high-dimensional particle filtering," *Monthly Weather Review*, vol. 136, no. 12, pp. 4629–4640, 2008.
- [8] P. Rebescini and R. van Handel, "Can local particle filters beat the curse of dimensionality?" *Annals of Applied Probability*, vol. 25, no. 5, pp. 2809–2866, 2015.
- [9] C. Snyder, T. Bengtsson, and M. Morzfeld, "Performance bounds for particle filters using the optimal proposal," *Monthly Weather Review*, vol. 143, no. 11, pp. 4750–4761, 2015.
- [10] 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)*, vol. 75, no. 3, pp. 397–426, 2013.
- [11] 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.
- [12] D. Crisan, J. Míguez *et al.*, "Nested particle filters for online parameter estimation in discrete-time state-space Markov models," *Bernoulli*, vol. 24, no. 4A, pp. 3039–3086, 2018.
- [13] T. Chen, T. B. Schön, H. Ohlsson, and L. Ljung, "Decentralized particle filter with arbitrary state decomposition," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 465–478, Feb 2011.
- [14] 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)*, Brussels, Belgium, 2012, pp. 488–493.
- [15] C. A. Naesseth, F. Lindsten, and T. B. Schön, "Sequential Monte Carlo for Graphical Models," in *Advances in Neural Information Processing Systems 27*. Montreal, Canada: Curran Associates, Inc., 2014, pp. 1862–1870.
- [16] C. Vergé, C. Dubarry, P. Del Moral, and E. Moulines, "On parallel implementation of sequential Monte Carlo methods: the island particle model," *Statistics and Computing*, vol. 25, no. 2, pp. 243–260, 2015.
- [17] A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou, "A stable particle filter for a class of high-dimensional state-space models," *Advances in Applied Probability*, vol. 49, no. 1, p. 24–48, 2017.
- [18] N. Jaoua, E. Duflos, P. Vanheeghe, and F. Septier, "Bayesian nonparametric state and impulsive measurement noise density estimation in nonlinear dynamic systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 5755–5759.
- [19] R. Lamberti, Y. Petetin, F. Desbouvries, and F. Septier, "Independent resampling sequential monte carlo algorithms," *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5318–5333, 2017.
- [20] P. Rebescini and R. van Handel, "Can local particle filters beat the curse of dimensionality?" *Ann. Appl. Probab.*, vol. 25, no. 5, pp. 2809–2866, 10 2015.
- [21] J. Briggs, M. Dowd, and R. Meyer, "Data assimilation for large-scale spatio-temporal systems using a location particle smoother," *Environmetrics*, vol. 24, no. 2, pp. 81–97, 2013.
- [22] P. M. Djuric and M. F. Bugallo, "Particle filtering for high-dimensional systems," in *Proceedings of the IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2013, pp. 352–355.
- [23] P. Fearnhead, O. Papaspiliopoulos, G. O. Roberts, and A. Stuart, "Random-weight particle filtering of continuous time processes," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 4, pp. 497–512, 2010.
- [24] L. Martino, V. Elvira, and F. Louzada, "Weighting a resampled particles

- in sequential Monte Carlo (extended preprint),” *viXra e-prints*, Feb. 2016.
- [25] L. Martino, V. Elvira, and G. Camps-Valls, “Group importance sampling for particle filtering and mcmc,” *Digital Signal Processing*, vol. 82, pp. 133–151, 2018.
- [26] C. A. Naesseth, F. Lindsten, and T. B. Schön, “Nested sequential Monte Carlo methods,” in *The 32nd International Conference on Machine Learning*, ser. JMLR W&CP, vol. 37, Lille, France, jul 2015, pp. 1292–1301.
- [27] R. Stern, “A statistical contribution to historical linguistics,” Ph.D. dissertation, Carnegie Mellon University, Department of Statistics, Carnegie Mellon University, Pittsburgh PA 15213, 5 2015.
- [28] F. Septier and G. W. Peters, “Langevin and Hamiltonian based sequential MCMC for efficient Bayesian filtering in high-dimensional spaces,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 312–327, March 2016.
- [29] M. I. Jordan, “Graphical models,” *Statistical Science*, vol. 19, no. 1, pp. 140–155, 2004.
- [30] F. Hamze and N. de Freitas, “Hot coupling: a particle approach to inference and normalization on pairwise undirected graphs of arbitrary topology,” in *Advances in Neural Information Processing Systems (NIPS)*, 2005.
- [31] R. G. Everitt, “Bayesian parameter estimation for latent Markov random fields and social networks,” *Journal of Computational and Graphical Statistics*, vol. 21, no. 4, pp. 940–960, 2012.
- [32] C. A. Naesseth, F. Lindsten, and T. B. Schön, “Capacity estimation of two-dimensional channels using sequential Monte Carlo,” in *Proceedings of the IEEE Information Theory Workshop (ITW)*, Hobart, Tasmania, Australia, November 2014.
- [33] C. A. Naesseth, F. Lindsten, and T. B. Schön, “Towards automated sequential Monte Carlo for probabilistic graphical models,” in *NIPS Workshop on Black Box Inference and Learning*, Montreal, Canada, 2015.
- [34] F. Lindsten, A. M. Johansen, A. C. Naesseth, B. Kirkpatrick, T. B. Schön, J. Aston, and A. Bouchard-Côté, “Divide-and-Conquer with sequential Monte Carlo,” *Journal of Computational and Graphical Statistics (JCGS)*, vol. 26, no. 2, pp. 445–458, 2017.
- [35] Q. Fu, A. Banerjee, S. Liess, and P. K. Snyder, “Drought detection of the last century: An MRF-based approach,” in *Proceedings of the 2012 SIAM International Conference on Data Mining*, Anaheim, CA, USA, April 2012, pp. 24–34.
- [36] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. Springer-Verlag New York, 2005.
- [37] R. Douc, E. Moulines, and D. Stoffer, *Nonlinear time series: Theory, methods and applications with R examples*. CRC Press, 2014.
- [38] R. H. Shumway and D. S. Stoffer, *Time series analysis and its applications: with R examples*. Springer Science & Business Media, 2010.
- [39] C. K. Wikle and M. B. Hooten, “A general science-based framework for dynamical spatio-temporal models,” *Test*, vol. 19, no. 3, pp. 417–451, 2010.
- [40] 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*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
- [41] L. Stewart and P. McCarty, Jr., “Use of Bayesian belief networks to fuse continuous and discrete information for target recognition, tracking, and situation assessment,” in *Proc. SPIE*, vol. 1699, 1992, pp. 177–185.
- [42] G. Kitagawa, “A Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space models,” in *Proceedings of the 2nd US-Japan joint Seminar on Statistical Time Series Analysis*, 1993, pp. 110–131.
- [43] M. K. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filters,” *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.
- [44] A. M. Johansen and A. Doucet, “A note on auxiliary particle filters,” *Statistics & Probability Letters*, vol. 78, no. 12, pp. 1498–1504, 2008.
- [45] R. Douc, E. Moulines, and J. Olsson, “Optimality of the auxiliary particle filter,” *Probability and Mathematical Statistics*, vol. 29, no. 1, pp. 1–28, 2009.
- [46] P. Fearnhead and P. Clifford, “On-line inference for hidden Markov models via particle filters,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 65, no. 4, pp. 887–899, 2003.
- [47] C. K. Carter and R. Kohn, “On Gibbs sampling for state space models,” *Biometrika*, vol. 81, no. 3, pp. 541–553, 1994.
- [48] S. Frühwirth-Schnatter, “Data augmentation and dynamic linear models,” *Journal of Time Series Analysis*, vol. 15, no. 2, pp. 183–202, 1994.
- [49] S. J. Godsill, A. Doucet, and M. West, “Monte Carlo smoothing for nonlinear time series,” *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, Mar. 2004.
- [50] F. Lindsten and T. B. Schön, “Backward simulation methods for Monte Carlo statistical inference,” *Foundations and Trends in Machine Learning*, vol. 6, no. 1, pp. 1–143, 2013.
- [51] C. Andrieu, A. Doucet, and R. Holenstein, “Particle Markov chain Monte Carlo methods,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.
- [52] P. Del Moral, *Feynman-Kac Formulae - Genealogical and Interacting Particle Systems with Applications*, ser. Probability and its Applications. Springer-Verlag New York, 2004.
- [53] A. Beskos, D. Crisan, and A. Jasra, “On the stability of sequential monte carlo methods in high dimensions,” *The Annals of Applied Probability*, vol. 24, no. 4, pp. 1396–1445, 08 2014.
- [54] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME, Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [55] L. Murray, “Personal communication,” 2016.
- [56] D. Clifford, D. Pagendam, J. Baldock, N. Cressie, R. Farquharson, M. Farrell, L. Macdonald, and L. Murray, “Rethinking soil carbon modelling: a stochastic approach to quantify uncertainties,” *Environmetrics*, vol. 25, no. 4, pp. 265–278, 2014.
- [57] A. M. Johansen, “On blocks, tempering and particle MCMC for systems identification,” in *Proceedings of 17th IFAC Symposium on System Identification*, Beijing, China, October 2015.
- [58] A. Doucet, M. Briers, and S. Sénécal, “Efficient block sampling strategies for sequential Monte Carlo methods,” *Journal of Computational and Graphical Statistics*, vol. 15, no. 3, pp. 693–711, 2006.

Christian A. Naesseth is currently a PhD student at the Department of Electrical Engineering, Linköping University. During August 2016 through June 2017 he visited Prof. David M. Blei’s research lab at Columbia University in the City of New York as a Fulbright Visiting Student Researcher. He spent time in October 2015 as a visiting student researcher in Dr. Frank Wood’s lab at the University of Oxford, United Kingdom. He received his MSc in Applied Physics and Electrical Engineering - international with specialization in Control and Information Systems, from Linköping University in 2013. In 2012, he received his BSc degree in Mathematics, also from Linköping University. During his undergraduate studies he spent the third year at Beijing Institute of Technology as an exchange student. Furthermore, Christian has also studied three semesters of Chinese language at Shanghai Jiaotong University.

Christian’s main research interests are machine learning, computational statistics and system identification. The focus is developing new algorithms, theory and practical tools for challenging problems within data science.

Fredrik Lindsten is Associate Professor at the Division of Statistics and Machine Learning, Department of Computer and Information Science, Linköping University, Sweden. He received his MSc degree in Applied Physics and Electrical Engineering in 2008 and a PhD in Automatic Control in 2013, both from Linköping University. In 2014 and 2015 he was a Postdoctoral Research Associate at the Department of Engineering, the University of Cambridge, UK. During spring 2012 he was a Visiting Student Researcher at the Statistical Artificial Intelligence Lab at the University of California, Berkeley, USA and during spring 2015 he was a Visiting Scholar at the Department of Statistics, the University of Oxford, UK. He has received the Ingvar Carlsson Award by the Swedish Foundation for Strategic Research, and the Benzelius Award by the Royal Society of Sciences in Uppsala. Lindsten’s main research interests are in statistical machine learning and computational statistics.

Thomas B. Schön is Professor of the Chair of Automatic Control in the Department of Information Technology at Uppsala University. He received the PhD degree in Automatic Control in Feb. 2006, the MSc degree in Applied Physics and Electrical Engineering in Sep. 2001, the BSc degree in Business Administration and Economics in Jan. 2001, all from Linköping University. He has held visiting positions with the University of Cambridge (UK), the University of Newcastle (Australia) and Universidad Técnica Federico Santa María (Valparaíso, Chile).

He is a member of The Royal Swedish Academy of Engineering Sciences (IVA) and The Royal Society of Sciences at Uppsala. He received the Tage Erlander prize for natural sciences and technology in 2017 and the Arnberg prize in 2016, both awarded by the Royal Swedish Academy of Sciences (KVA). He was awarded the Automatica Best Paper Prize in 2014, and in 2013 he received the best PhD thesis award by The European Association for Signal Processing (EURASIP). He received the best teacher award at the Institute of Technology, Linköping University in 2009. Schön's main research interest is probabilistic machine learning, with close links to signal processing and automatic control. He is a Senior member of the IEEE and an Associate Editor of *Automatica*.