

RESEARCH

Open Access



On the performance of parallelisation schemes for particle filtering

Dan Crisan¹, Joaquín Míguez^{2,3*}  and Gonzalo Ríos-Muñoz^{2,3}

Abstract

Considerable effort has been recently devoted to the design of schemes for the parallel implementation of sequential Monte Carlo (SMC) methods for dynamical systems, also widely known as particle filters (PFs). In this paper, we present a brief survey of recent techniques, with an emphasis on the availability of analytical results regarding their performance. Most parallelisation methods can be interpreted as running an ensemble of lower-cost PFs, and the differences between schemes depend on the degree of interaction among the members of the ensemble. We also provide some insights on the use of the simplest scheme for the parallelisation of SMC methods, which consists in splitting the computational budget into M non-interacting PFs with N particles each and then obtaining the desired estimators by averaging over the M independent outcomes of the filters. This approach minimises the parallelisation overhead yet still displays desirable theoretical properties. We analyse the mean square error (MSE) of estimators of moments of the optimal filtering distribution and show the effect of the parallelisation scheme on the approximation error rates. Following these results, we propose a time–error index to compare schemes with different degrees of parallelisation. Finally, we provide two numerical examples involving stochastic versions of the Lorenz 63 and Lorenz 96 systems. In both cases, we show that the ensemble of non-interacting PFs can attain the approximation accuracy of a centralised PF (with the same *total* number of particles) in just a fraction of its running time using a standard multicore computer.

Keywords: Particle filtering, Parallelisation, Convergence analysis, Particle islands, Lorenz 63, Lorenz 96

1 Introduction

Over the past decade, there has been a continued interest in the design of schemes for the implementation of particle filtering algorithms using parallel or distributed hardware of various types, including general purpose devices such as multi-core CPUs or graphical processing units (GPUs) [1] and application-tailored devices such as field-programmable gate arrays (FPGAs) [2]. A particle filter (PF) is a recursive algorithm for the approximation of the sequence of posterior probability distributions that arise from a stochastic dynamical system in state-space form (see, e.g. [3–6] and references therein for a general view of the field). A typical PF includes three steps that are repeated sequentially:

- Monte Carlo sampling in the space of the state variables,
- Computation of weights for the generated samples and, finally,
- Resampling according to the weights.

While at first sight the algorithm may look straightforward to parallelise (sampling and weighting can be carried out concurrently without any constraint), the resampling step involves the interaction of the whole set of Monte Carlo samples. Several authors have proposed schemes for ‘splitting’ the resampling step into simpler tasks that can be carried out concurrently. The approaches are diverse and range from the heuristic [7–9] to the mathematically well-principled [2, 10–14]. However, the former are largely based on (often loose) approximations that prevent the claim of any rigorous guarantees of convergence, whereas the latter involve non-negligible overhead to ensure the proper interaction of particles.

The goal of this paper is to provide

*Correspondence: joaquin.miguez@uc3m.es

²Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Madrid, Spain

³Instituto de Investigación Sanitaria Gregorio Marañón (IISGM), Madrid, Spain
Full list of author information is available at the end of the article

- (i) A survey of recently proposed, and mathematically well grounded, parallelisation schemes for particle filtering and
- (ii) Analytical insights into the performance of the simplest parallelisation method, namely the averaging of statistically independent PFs.

Besides describing the various methodologies, we aim at characterising their performance analytically whenever possible. For that purpose, we need to introduce accurate notation, unfortunately a bit more involved than needed for the mere description of the algorithmic steps. Then, we describe and provide a basic convergence result for the standard PF and proceed to describe four different approaches to its parallelisation: the simple averaging of statistically independent (i.e. non interacting) low-complexity PFs, the method based on the distributed resampling with non-proportional allocation (DRNA) procedure of [2, 10, 11], the particle island model of [13, 14] and the adaptive interaction scheme termed α -sequential Monte Carlo (α -SMC) in [12].

The simplest parallelisation scheme consists in running M statistically independent PFs with N particles (i.e. Monte Carlo samples) each and then averaging the M independent estimators. This approach has the limitation that the bias of the averaged estimator depends only on N . Hence, if N is relatively small, the bias is large even if we use a very high number M of parallel filters. This drawback can be overcome by allowing some degree of interaction among the M concurrently running PFs. The DRNA, particle island and α -SMC approaches introduce this interaction in different flavours. In DRNA-based ensembles of PFs, each filter runs separately but it periodically exchanges a few particles with other members of the ensemble using a communication network [10]. Algorithms in the particle island class rely on two levels of resampling: conventional resampling at particle level and island-level resampling, where complete sets of particles (associated to parallel-running PFs) are replicated or eliminated stochastically [13, 14]. Finally, the α -SMC scheme of [12] is a very flexible methodology that enables the adaptive selection of different interaction patterns (i.e. which particles are resampled together) over time. For each one of these techniques, we describe the methodology and establish basic theoretical guarantees for convergence.

In the second part of the paper, we focus on the analysis of the performance of the simplest parallelisation scheme, the averaging of M statistically independent PFs with N particles each. Under mild assumptions, we analyse the mean square error (MSE) of the estimators of one-dimensional statistics of the optimal filtering distribution and show explicitly the effect of the parallelisation scheme on the convergence rate. Specifically, we study the

decomposition of the MSE into variance and bias components, to show that the variance is $O\left(\frac{1}{MN}\right)$, i.e. it decreases linearly with the total number of particles, while the bias is $O\left(\frac{1}{N^2}\right)$, i.e. it goes to 0 quadratically with N . These results have already been obtained, e.g. in [13] using the Feynman-Kac framework of [4]. Here, we aim at providing a self-contained analysis that illustrates the key theoretical issues in the convergence of parallel PFs. All proofs are constructed from elementary principles, and we obtain explicit error rates (for the bias, the variance and the MSE) that hold for all M and N , while the theorems in [13] are strictly asymptotic. While we have focused here on PFs for discrete-time state-space models, the analysis can be similarly done for continuous-time systems, and, indeed, the basic results needed for that case can be found in [15]. Finally, in order to compare different parallelisation schemes, we introduce a time–error index that combines time complexity (asymptotic order of the running time) and estimation accuracy (asymptotic error rates) into a single quantitative figure of merit that can be used to compare schemes with different degrees of interaction.

The rest of the paper is organised as follows. In Section 2, we present basic background material, and notation, for the analysis of PFs. Section 3 is devoted to a survey of parallelisation schemes for particle filtering. Our analysis of the ensemble of non-interacting PFs is presented in Section 4. In Section 5, we present numerical results for two examples, namely the filtering of stochastic versions of the Lorenz 63 and Lorenz 96 systems, respectively. The latter is often used as a simplified model of atmospheric dynamics, and it has the property that it can be scaled to an arbitrary dimension. Our simulation results show that the use of averaged estimators computed from ensembles of non-interacting filters can be advantageous in terms of accuracy (not only running times) as the system dimension grows. Finally, Section 6 is devoted to a discussion of the obtained results, together with some concluding remarks.

2 Background

2.1 Notation and preliminaries

We first introduce some common notations to be used through the paper, broadly classified by topics. Below, \mathbb{R} denotes the real line, while for an integer $d \geq 1$,

$$\mathbb{R}^d = \overbrace{\mathbb{R} \times \dots \times \mathbb{R}}^{d \text{ times}}$$

- Functions.
 - The supremum norm of a real function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is denoted as $\|f\|_\infty = \sup_{x \in \mathbb{R}^d} |f(x)|$.
 - $B(S)$ is the set of bounded real functions over $S \subseteq \mathbb{R}^d$, i.e. $f \in B(\mathbb{R}^d)$ if, and only if, f has domain S and $\|f\|_\infty < \infty$.

- Measures and integrals. Let $S \subseteq \mathbb{R}^d$ be a subset of \mathbb{R}^d .
 - $\mathcal{B}(S)$ is the σ -algebra of Borel subsets of S .
 - $\mathcal{P}(S)$ is the set of probability measures over the measurable space $(\mathcal{B}(S), S)$.
 - $(f, \mu) \triangleq \int f(x)\mu(dx)$ is the integral of a real function $f : S \rightarrow \mathbb{R}$ with respect to (w.r.t.) a measure $\mu \in \mathcal{P}(S)$.
 - Given a probability measure $\mu \in \mathcal{P}(S)$, a Borel set $A \in \mathcal{B}(S)$ and the indicator function

$$I_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{otherwise} \end{cases},$$

$$\mu(A) = (I_A, \mu) \text{ is the probability of } A.$$

- Sequences, vectors and random variables (r.v.'s).
 - We use a subscript notation for sequences, namely $x_{t_1:t_2} \triangleq \{x_{t_1}, \dots, x_{t_2}\}$.
 - For an element $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ of a Euclidean space, its norm is denoted as $\|x\| = \sqrt{x_1^2 + \dots + x_d^2}$.
 - The L_p norm of a real r.v. Z , with $p \geq 1$, is written as $\|Z\|_p \triangleq E[|Z|^p]^{1/p}$, where $E[\cdot]$ denotes expectation w.r.t. the distribution of Z .

2.2 State-space Markov models in discrete time

Consider two random sequences, $\{X_t\}_{t \geq 0}$ and $\{Y_t\}_{t \geq 1}$, taking values in $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ and \mathbb{R}^{d_y} , respectively. Let \mathbb{P}_t be the joint probability measure for the collection of random variables $\{X_0, X_n, Y_n\}_{1 \leq n \leq t}$.

We refer to the sequence $\{X_t\}_{t \geq 0}$ as the state (or signal) process, and we assume that it is an inhomogeneous Markov chain governed by an initial probability measure $\tau_0 \in \mathcal{P}(\mathcal{X})$ and a sequence of Markov transition kernels $\tau_t : \mathcal{B}(\mathcal{X}) \times \mathcal{X} \rightarrow [0, 1]$. To be specific, we define

$$\tau_0(A) \triangleq \mathbb{P}_0 \{X_0 \in A\}, \tag{1}$$

$$\tau_t(A|x_{t-1}) \triangleq \mathbb{P}_t \{X_t \in A | X_{t-1} = x_{t-1}\}, \quad t \geq 1, \tag{2}$$

where $A \in \mathcal{B}(\mathcal{X})$ is a Borel set. The sequence $\{Y_t\}_{t \geq 1}$ is termed the observation process. Each r.v. Y_t is assumed to be conditionally independent of other observations given X_t ; hence, the conditional distribution of the r.v. Y_t given $X_t = x_t$ is fully described by the probability density function (pdf) $g_t(y_t|x_t) > 0$. We often use g_t as a function of x_t (i.e. as a likelihood) and hence we write $g_t^y(x) \triangleq g_t(y|x)$. The prior τ_0 , the kernels $\{\tau_t\}_{t \geq 1}$ and the functions $\{g_t\}_{t \geq 1}$ describe a stochastic Markov state-space model in discrete time.

The stochastic filtering problem consists in the computation of the posterior probability measure of the state X_t given the sequence of observations up to time t . Specifically, for a given observation record $\{y_t\}_{t \geq 1}$, we seek the probability measures

$$\pi_t(A) \triangleq \mathbb{P}_t \{X_t \in A | Y_{1:t} = y_{1:t}\}, \quad t = 0, 1, 2, \dots$$

where $A \in \mathcal{B}(\mathcal{X})$. For many practical problems, the interest actually lies in the computation of statistics of π_t , e.g. the posterior mean or the posterior variance of X_t . Such statistics can be written as integrals of the form (f, π_t) , for some function $f : \mathcal{X} \rightarrow \mathbb{R}$. Note that, for $t = 0$, we recover the prior signal measure, i.e. $\pi_0 = \tau_0$.

An associated problem is the computation of the one-step-ahead predictive measure

$$\xi_t(A) \triangleq \mathbb{P}_t \{X_t \in A | Y_{1:t-1} = y_{1:t-1}\}, \quad t = 1, 2, \dots$$

This measure can be explicitly written in terms of the kernel τ_t and the filter π_{t-1} . Indeed, for any integrable function $f : \mathcal{X} \rightarrow \mathbb{R}$, we readily obtain (see, e.g. ([6] Chapter 10))

$$\begin{aligned} (f, \xi_t) &= \int \int f(x)\tau_t(dx|x')\pi_{t-1}(dx') \\ &= ((f, \tau_t), \pi_{t-1}), \end{aligned} \tag{3}$$

and we write $\xi_t = \tau_t \pi_t$ as shorthand.

The filter at time t , π_t , can be obtained from the predictive measure, ξ_t , and the likelihood, g_t^y , by way of the so-called projective product [6] or Boltzman-Gibbs transformation [4], $\pi_t = g_t^y \star \xi_t$, defined as

$$(f, g_t^y \star \xi_t) \triangleq \frac{(fg_t^y, \xi_t)}{(g_t^y, \xi_t)}$$

for any integrable function $f : \mathcal{X} \rightarrow \mathbb{R}$. Combined with (3), this yields the recursive formula

$$\pi_t = g_t^y \star \tau_t \pi_{t-1}. \tag{4}$$

It is key to the analysis of Section 4 to keep track of the sequence of non-normalised measures $\{\rho_t\}_{t \geq 0}$, where

$$\rho_0 = \pi_0, \quad \rho_t = g_t^y \cdot \tau_t \rho_{t-1} \tag{5}$$

and, for any integrable function $f : \mathcal{X} \rightarrow \mathbb{R}$ and any measure $\alpha \in \mathcal{P}(\mathcal{X})$, we define

$$(f, g_t^y \cdot \alpha) \triangleq (fg_t^y, \alpha). \tag{6}$$

We remark that ρ_t is *not* a probability measure but a non-normalised version of π_t , namely

$$(f, \pi_t) = \frac{(f, \rho_t)}{(\mathbf{1}, \rho_t)},$$

where $\mathbf{1}(x) = 1$ is the constant unit function.

2.3 Standard particle filter

Assume that a sequence of observations $Y_{1:T} = y_{1:T}$, for some $T < \infty$, is given. Then, the sequences of measures $\{\pi_t\}_{t \geq 1}$, $\{\xi_t\}_{t \geq 1}$ and $\{\rho_t\}_{t \geq 0}$ can be numerically approximated using particle filtering. PFs are numerical methods based on the recursive relationships (4) and (6). The simplest algorithm, often called ‘standard particle filter’ or

‘bootstrap filter’ [16] (see also [17]), can be described as follows.

Algorithm 1: Bootstrap filter

1. Initialisation. At time $t = 0$, draw N i.i.d. samples, $x_0^{(n)}, n = 1, \dots, N$, from the distribution τ_0 .
2. Recursive step. Let $\{x_{t-1}^{(n)}\}_{1 \leq n \leq N}$ be the particles (samples) generated at time $t - 1$. At time t , proceed with the two steps below.

- (a) For $n = 1, \dots, N$, draw a sample $\bar{x}_t^{(n)}$ from the probability distribution $\tau_t(\cdot | x_{t-1}^{(n)})$ and compute the normalised weight

$$w_t^{(n)} = \frac{g_t^{y_t}(\bar{x}_t^{(n)})}{\sum_{k=1}^N g_t^{y_t}(\bar{x}_t^{(k)})}. \tag{7}$$

- (b) For $n = 1, \dots, N$, let $x_t^{(n)} = \bar{x}_t^{(k)}$ with probability $w_t^{(k)}, k \in \{1, \dots, N\}$.
-

Step 2.(b) is referred to as resampling or selection. In the form stated here, it reduces to the so-called multinomial resampling algorithm [18, 19], but the convergence of the filter can be easily proved for various other schemes (see, e.g. the treatment of the resampling step in [6]).

Using the sets $\{\bar{x}_t^{(n)}\}_{1 \leq n \leq N}$ and $\{x_t^{(n)}\}_{1 \leq n \leq N}$, we construct random approximations of ξ_t, ρ_t and π_t , namely

$$\xi_t^N = \frac{1}{N} \sum_{n=1}^N \delta_{\bar{x}_t^{(n)}}, \quad \pi_t^N = \frac{1}{N} \sum_{n=1}^N \delta_{x_t^{(n)}}, \quad \rho_t^N = G_t^N \pi_t^N \tag{8}$$

where δ_x is the delta unit-measure located at $x \in \mathbb{R}^{d_x}$ and¹

$$G_t^N = \frac{1}{N^t} \prod_{k=1}^t \left(\sum_{j=1}^N g_k^{y_k}(\bar{x}_k^{(j)}) \right). \tag{9}$$

For any integrable function f on the state space, it is straightforward to approximate the integrals $(f, \xi_t), (f, \pi_t)$ and (f, ρ_t) as

$$(f, \xi_t) \approx (f, \xi_t^N) = \frac{1}{N} \sum_{n=1}^N f(\bar{x}_t^{(n)}),$$

$$(f, \pi_t) \approx (f, \pi_t^N) = \frac{1}{N} \sum_{n=1}^N f(x_t^{(n)}) \quad \text{and}$$

$$(f, \rho_t) \approx (f, \rho_t^N) = G_t^N (f, \pi_t^N),$$

respectively.

The convergence of PFs has been analysed in different ways [4, 6, 20–23]. Here, we use simple results for the convergence of the L_p norms ($p \geq 1$) of the approximation

errors. For the approximation of integrals w.r.t. ξ_t and π_t , we have the following standard result.

Lemma 1 Assume that the sequence of observations $Y_{1:T} = y_{1:T}$ is fixed (with $T < \infty$), $g_t^{y_t} \in B(\mathcal{X})$ and $g_t^{y_t} > 0$ (in particular, $(g_t^{y_t}, \xi_t) > 0$) for every $t = 1, 2, \dots, T$. Then for any $f \in B(\mathcal{X})$, any $p \geq 1$ and every $t = 1, \dots, T$,

$$\|(f, \xi_t^N) - (f, \xi_t)\|_p \leq \frac{\bar{c}_t \|f\|_\infty}{\sqrt{N}} \quad \text{and} \tag{10}$$

$$\|(f, \pi_t^N) - (f, \pi_t)\|_p \leq \frac{c_t \|f\|_\infty}{\sqrt{N}}, \tag{11}$$

where \bar{c}_t and c_t are finite constants independent of N , $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)| < \infty$ and the expectations are taken over the distributions of the measure-valued random variables ξ_t^N and π_t^N , respectively.

Proof This result is a special case of, e.g. Lemma 1 in [24]. \square

Remark 1 The constants \bar{c}_t and c_t can be easily shown to increase exponentially with t . It is possible to find error rates independent of t by imposing additional assumptions on the state-space model (related to the stability of the optimal filter, π_t) [4, 25].

3 Parallelisation schemes for particle filtering

3.1 Non-interacting particle filters

Assume we intend to run a PF with K particles. Most parallelisation schemes split the set of particles $\{x_t^{(k)}\}_{1 \leq k \leq K}$ into subsets and then run separate (but possibly interacting) PFs for each subset. To be specific, assume that the complete set of K particles can be divided into M subsets with N elements each, i.e. $K = MN$, and we construct disjoint subsets

$$\{x_t^{(m,n)}\}_{1 \leq n \leq N}, \quad \text{for } m = 1, \dots, M,$$

such that

$$\bigcup_{m=1}^M \{x_t^{(m,n)}\}_{1 \leq n \leq N} = \{x_t^{(k)}\}_{1 \leq k \leq K}.$$

In the simplest scheme, M independent (i.e. non-interacting) PFs are run separately. Assume for simplicity that the standard PF outlined in Algorithm 1 is used on each subset. Then, at each time t , we have M estimates of the filtering measure, namely

$$\pi_t^{m,N} = \frac{1}{N} \sum_{n=1}^N \delta_{x_t^{(m,n)}}, \quad m = 1, \dots, M. \tag{12}$$

Assuming that the goal is to approximate integrals of the form (f, π_t) , for some integrable real function $f : \mathcal{X} \rightarrow \mathbb{R}$,

then we obtain an ensemble of M independent and identically distributed (i.i.d.) estimators

$$(f, \pi_t^{m,N}) = \frac{1}{N} \sum_{n=1}^N f(x_t^{(m,n)}), \quad m = 1, \dots, M,$$

which can be averaged to yield

$$(f, \pi_t^{M \times N}) = \frac{1}{M} \sum_{m=1}^M (f, \pi_t^{m,N}) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N f(x_t^{(m,n)}), \quad (13)$$

where we have denoted $\pi_t^{M \times N} = \frac{1}{M} \sum_{m=1}^M \pi_t^{m,N}$.

This scheme is straightforward to implement, and it does not involve any parallelisation overhead as the M PFs do not interact. A self-contained analysis of the MSE of the ensemble estimator $(f, \pi_t^{M \times N})$ is presented in Section 4.

A key result, to be explicitly shown in our analysis but also pointed out in [13] and [12], is that the estimation bias $|E[(f, \pi_t) - (f, \pi_t^{M \times N})]|$ decreases as $O(N^{-2})$. This implies that if the number of particles per subset, N , is kept fixed, then the MSE, $E[|(f, \pi_t) - (f, \pi_t^{M \times N})|^2]$, remains bounded away from zero even if the number of subsets is made arbitrarily large, i.e. $M \rightarrow \infty$. This can be a drawback depending on the type of parallel computing configuration to be used. In multicore computers, for example, the number of subsets M can be expected to be moderate (of the order of cores available) and N can often be made large enough to make the bias negligible. On the other hand, implementations based on low-power processors, such as graphical processing units (GPUs) or wireless networks, are more efficient when operating with a large number of subsets, M , and a low number of particles per subset, N . In these scenarios, the bias of the non-interacting ensemble estimator in Eq. (13) can be significant. The solution to this limitation is to introduce some degree of interaction among the M parallel-running PFs. Some relevant schemes are described below.

3.2 Distributed resampling with non-proportional allocation

The scheme termed distributed resampling with non-proportional allocation (DRNA) for the parallelisation of PFs was originally introduced in [2] (Section IV.A.3), but it has been only recently that a theoretical characterisation of its performance has been obtained [10, 11, 26].

The same as in Section 3.1, assume that we have a budget of $K = MN$ particles, which are split into M subsets with N particles each. We run a standard PF for each

subset² which, in addition to the particles and weights, keeps track of the aggregated non-normalised weight

$$W_t^{(m)*} = W_{t-1}^{(m)*} \sum_{n=1}^N g_t^{y_t}(x_t^{(m,n)}). \quad (14)$$

Note that $W_t^{(m)*}$ represents the likelihood of the m th subset of particles $\{x_t^{(m,n)}\}_{1 \leq n \leq N}$. The normalised aggregated weights are computed as

$$W_t^{(m)} = \frac{W_t^{(m)*}}{\sum_{i=1}^M W_t^{(i)*}}, \quad m = 1, \dots, M.$$

In this scheme, the M parallel PFs are not independent. Every t_0 time steps, the PFs exchange subsets of particles and weights using a communication network [2]. This exchange can be formally described by means of a deterministic one-to-one map

$$\beta : \{1, \dots, M\} \times \{1, \dots, N\} \rightarrow \{1, \dots, M\} \times \{1, \dots, N\}$$

that keeps the number of particles per subset, N , invariant. Specifically, $(u, v) = \beta(m, n)$ means that the m th particle of the m th subset is transmitted to the u th subset, where it becomes particle number v . In summary, if we have the particles

$$\{x_t^{(m,n)}\}_{1 \leq n \leq N; 1 \leq m \leq M},$$

then, after the exchange step, the particles are re-labelled as

$$\{x_t^{\beta(m,n)}\}_{1 \leq n \leq N; 1 \leq m \leq M}.$$

Typically, only small subsets of particles are exchanged, hence $\beta(m, n) = (m, n)$ for most values of m and n . The resulting parallel particle filtering algorithm can be outlined as shown below (adapted from [10]).

We remark that every PF operates independently of all others except for the particle exchange, step 2.(c), which is carried out every t_0 time steps. The degree of interaction can be controlled by designing the map $\beta(m, k)$ in a proper way. Typically, exchanging a subset of particles with ‘neighbour’ PFs is sufficient. For example, if we assume the parallel PFs are arranged in a ring configuration, then the m th PF can exchange, say, two particles with PF number $m - 1$ and another two particles with PF number $m + 1$, in such a way that all parallel PFs retain N particles (four of them received from their neighbours) after the exchange.

We also note that the local resampling step is carried out independently, and concurrently, for each parallel-running PF and it does not change the aggregate weights, i.e. $\bar{W}_t^{(m)*} = \sum_{n=1}^N \bar{w}_t^{(m,n)*} = \sum_{n=1}^N \tilde{w}_t^{(m,n)*}$. We assume a multinomial resampling procedure, but other procedures can be used in an obvious manner.

Algorithm 2: DRNA-based parallel PFs with M subsets, N particles per subset and periodic particle exchanges every t_0 time steps

1. Initialisation: For $m = 1, \dots, M$ (concurrently) draw $x_0^{(m,n)} \sim \tau_0(dx)$, $n = 1, \dots, N$, and set $w_0^{(m,n)*} = \frac{1}{MN}$ and $W_0^{(m)*} = 1/M$.
2. Recursive step: Assume $\{x_{t-1}^{(m,n)}, w_{t-1}^{(m,n)*}\}_{1 \leq n \leq N}$ and $W_{t-1}^{(m)*}$ are available for $m = 1, \dots, M$.

- (a) For $m = 1, \dots, M$ (concurrently) and $n = 1, \dots, N$,

$$\text{draw } \tilde{x}_t^{(m,n)} \sim \tau_t(dx|x_{t-1}^{(m,n)}),$$

$$\text{compute } \tilde{w}_t^{(m,n)*} = w_{t-1}^{(m,n)*} g_t^{y_t}(\tilde{x}_t^{(m,n)}),$$

$$\text{and set } \tilde{W}_t^{(m)*} = \sum_{n=1}^N \tilde{w}_t^{(m,n)*}.$$

- (b) Local resampling: for $m = 1, \dots, M$ (concurrently) set $\tilde{x}_t^{(m,n)} = \tilde{x}_t^{(m,j)}$ with probability $\tilde{w}_t^{(m,j)*} = \frac{\tilde{w}_t^{(m,j)*}}{\tilde{W}_t^{(m)*}}$, for $n = 1, \dots, N$ and $j \in \{1, \dots, N\}$.

Set $\tilde{w}_t^{(m,n)*} = \frac{\tilde{W}_t^{(m)*}}{N}$ for each m and all n .

- (c) Particle exchange: If t is an integer multiple of t_0 , then set

$$x_t^{\beta(m,n)} = \tilde{x}_t^{(m,n)} \quad \text{and} \quad w_t^{\beta(m,n)*} = \tilde{w}_t^{(m,n)*}$$

for every $(m, n) \in \{1, \dots, M\} \times \{1, \dots, N\}$. Also set $W_t^{(m)*} = \sum_{n=1}^N w_t^{(m,n)*}$ for every $m = 1, \dots, M$. Otherwise, set $x_t^{(m,n)} = \tilde{x}_t^{(m,n)}$, $w_t^{(m,n)*} = \tilde{w}_t^{(m,n)*}$, $W_t^{(m)*} = \tilde{W}_t^{(m)*}$.

The ensemble estimator of the optimal filter π_t is now computed as the weighted average

$$\pi_t^{M \times N} = \sum_{m=1}^M W_t^{(m)} \pi_t^{m,N}, \quad \text{where} \quad \pi_t^{m,N} = \frac{1}{N} \sum_{n=1}^N \delta_{x_t^{(m,n)}}.$$

The particle estimator of (f, π_t) then becomes $(f, \pi_t^{M \times N}) = \sum_{m=1}^M \frac{W_t^{(m)}}{N} \sum_{n=1}^N f(x_t^{(m,n)})$.

The scheme in Algorithm 2 has been proved to converge uniformly over time, under some standard assumptions, when the number of particles per subset, N , is kept fixed and the number of subsets (i.e. the number of parallel PFs), M , is increased. To be specific, we have the following result, which is proved in [10] (Section 3.2).

Theorem 1 *If the following three assumptions hold:*

- i. The sequence of observations $\{y_t\}_{t \geq 1}$ is fixed (but otherwise arbitrary) and there exists a real constant

$0 < a < \infty$ such that $\frac{1}{a} < g_t^{y_t}(x) < a$ for every $t \geq 1$ and every $x \in \mathcal{X}$.

- ii. The sequence of probability measures $\{\pi_t\}_{t \geq 0}$ is stable (see [25]).

- iii. The particle exchange step guarantees that

$$E \left[\left(\sup_{1 \leq m \leq M} W_{rt_0}^{(m)} \right)^q \right] \leq \frac{c^q}{M^{q-\epsilon}}, \quad \text{for every } r \in \mathbb{N}$$

and some constants $c < \infty$, $0 \leq \epsilon < 1$ and $q \geq 4$ independent of M .

Then, for any fixed $0 < N < \infty$,

$$\lim_{M \rightarrow \infty} \sup_{t \geq 0} \left\| (f, \pi_t^{M \times N}) - (f, \pi_t) \right\|_p = 0$$

for any $f \in B(\mathcal{X})$ and every $1 \leq p \leq q$.

Assumption iii. in the latter theorem indicates that none of the M subsets should accumulate too much aggregate weight compared to the other subsets. This accumulation of weight is precisely controlled by the particle exchange steps. In a practical implementation, the aggregate weights $W_t^{(m)*}$ should be monitored and additional particle exchange steps should be triggered when the weight of any subset increases beyond some prescribed threshold.

3.3 Particle islands

The *particle island* model was introduced in [13] in order to address the parallel processing of subsets of particles in SMC methods in a systematic manner. Similar to the DRNA-based PFs of Section 3.2, the algorithms proposed in [13] are based on running M parallel PFs, each one on a disjoint subset of particles, namely $\{x_t^{(m,n)}\}_{1 \leq n \leq N}$ for the m th filter, and keep track of the non-normalised aggregate weights $W_t^{(m)*}$ defined in Eq. (14).

However, particle island methods do not rely on an exchange of particles between the PFs running the different subsets. Instead, a resampling scheme in two levels is implemented.

- Particle level: resampling is carried out locally within each of the M concurrently running PFs. This is equivalent to the local resampling step in Algorithm 2.
- Island level: the aggregate weights $W_t^{(m)}$ are used to resample the particle subsets, or *islands*, assigned to the individual PFs. In this step, complete subsets can be replicated or eliminated (in the same way as particles are in a conventional, or particle level, resampling step).

We now outline the *double bootstrap filter*, an algorithm described in [13] (Algorithm 1) that performs multinomial resampling at both the particle level and the island

level. While in the version of [13] both resampling steps are taken at every time step t , we describe a slightly more general procedure where the island-level resampling steps are taken periodically, every $t_0 \geq 1$ time steps. For simplicity, we introduce the notation $X_t^{m,N} = \{x_t^{(m,n)}\}_{1 \leq n \leq N}$ for the subset of N particles assigned to the m th island (ie. the m th concurrently running PF).

Algorithm 3: Double bootstrap filter with M islands and N particles per island. Island-level resampling every t_0 time steps

1. **Initialisation:** Draw sets of particles $X_0^{m,N} = \{x_0^{(m,n)}\}_{1 \leq n \leq N}$, $m = 1, \dots, M$, independently from the prior distribution τ_0 .
2. **Recursive step:** Assume $X_{t-1}^{m,N}$ and $W_{t-1}^{(m)*}$ are available for $m = 1, \dots, M$.

- (a) For $m = 1, \dots, M$ (concurrently) and $n = 1, \dots, N$,

$$\text{draw } \tilde{x}_t^{(m,n)} \sim \tau_t(dx|x_{t-1}^{(m,n)}),$$

$$\text{compute } \tilde{w}_t^{(m,n)*} = w_{t-1}^{(m,n)*} g_t^{y_t}(\tilde{x}_t^{(m,n)}),$$

$$\text{and set } \tilde{W}_t^{(m)*} = \sum_{n=1}^N \tilde{w}_t^{(m,n)*}.$$

- (b) **Particle-level resampling:** for $m = 1, \dots, M$ (concurrently) set $\tilde{x}_t^{(m,n)} = \tilde{x}_t^{(m,j)}$ with probability $\tilde{w}_t^{(m,j)*} = \frac{w_t^{(m,j)*}}{\tilde{W}_t^{(m)*}}$, for $n = 1, \dots, N$ and $j \in \{1, \dots, N\}$.

Set $\tilde{w}_t^{(m,n)*} = \frac{\tilde{W}_t^{(m)*}}{N}$, for each m and all n , and $\tilde{X}_t^{m,N} = \{\tilde{x}_t^{(m,n)}\}_{1 \leq n \leq N}$ for $m = 1, \dots, M$.

- (c) **Island-level resampling:** let $\bar{W}_t^{(m)} = \frac{\tilde{W}_t^{(m)*}}{\sum_{l=1}^M \tilde{W}_t^{(l)*}}$, $m = 1, \dots, M$, be the normalised island weights. If t is an integer multiple of t_0 ,

- then, for $m = 1, \dots, M$, let $X_t^{m,N} = \tilde{X}_t^{j,N}$ with probability $\bar{W}_t^{(j)}$, $j \in \{1, \dots, M\}$, and set $W_t^{(m)*} = \frac{1}{M}$;
 - else, for $m = 1, \dots, M$, set $X_t^{m,N} = \tilde{X}_t^{m,N}$ and $W_t^{(m)*} = \bar{W}_t^{(m)*}$.
-

In Algorithm 3, a multinomial resampling procedure is employed both at the particle level and the island level. Other schemes are obviously possible and some of them are explored in [13], including ϵ -interactions and resampling conditional on the effective sample size.

The particle approximation of the optimal filter π_t takes the form $\pi_t^{M \times N} = \sum_{m=1}^M W_t^{(m)} \pi_t^{m,N}$, where $\pi_t^{m,N} = \frac{1}{N}$

$\sum_{n=1}^N \delta_{x_t^{(m,n)}}$. This is formally identical to the DRNA-based Algorithm 2, although the procedure for the computation of the particles and weights is obviously different.

The asymptotic convergence of the double bootstrap filter was proved in [13] using the Feynman-Kac machinery of [4]. Then, in the follow-up paper [14], a central limit theorem was proved and bounds on the asymptotic variance of a class of schemes that includes Algorithm 3 were derived. Here we reproduce the basic convergence result of [13], adapted to the notation of this paper.

Theorem 2 Assume that the sequence of observations $y_{1:T}$ is arbitrary but fixed, T is arbitrarily large but finite and the likelihood functions $g_t^{y_t}(x)$ are positive and bounded for $1 \leq t \leq T$. Then, for any $f \in B(\mathcal{X})$ and every $t = 1, \dots, T$,

$$\lim_{N \rightarrow \infty} \lim_{M \rightarrow \infty} NM \times E \left[\left(f, \pi_t^{M \times N} \right) - (f, \pi_t) \right] = B(f, t) < \infty,$$

and

$$\lim_{N \rightarrow \infty} \lim_{M \rightarrow \infty} NM \times \text{Var} \left[\left(f, \pi_t^{M \times N} \right) \right] = V(f, t) < \infty,$$

where $\text{Var}[\cdot]$ denotes the variance of a random variable and $B(f, t)$ and $V(f, t)$ are finite constants with respect to both M and N .

The results in Theorem 2 can be adapted to the case where the island-level resampling step is removed from Algorithm 3, effectively converting the double bootstrap method into an ensemble of non-interacting PFs. It is proved in [13] that, in such case,

$$\lim_{N \rightarrow \infty} \lim_{M \rightarrow \infty} N \times E \left[\left(f, \pi_t^{M \times N} \right) - (f, \pi_t) \right] = \bar{B}(f, t) < \infty,$$

and

$$\lim_{N \rightarrow \infty} \lim_{M \rightarrow \infty} NM \times \text{Var} \left[\left(f, \pi_t^{M \times N} \right) \right] = \bar{V}(f, t) < \infty$$

where the constants $\bar{B}(f, t)$ and $\bar{V}(f, t)$ are independent of M and N . This implies that the bias of the estimator $\left(f, \pi_t^{M \times N} \right)$ with non-interacting PFs depends only on N and cannot be eliminated by taking $M \rightarrow \infty$ alone. The MSE of Algorithm 3, on the other hand, vanishes as $MN \rightarrow \infty$.

3.4 Adaptive interaction pattern: the α -SMC methodology

Rather than working with *fixed* subsets $X_t^{m,N} = \{x_t^{(m,n)}\}_{1 \leq n \leq N}$, $m = 1, \dots, M$, the α -SMC methodology of [12] enables the construction of particle filtering algorithms with *adaptive* interaction patterns. In particular, it is possible to devise parallelised PFs within this framework where the subsets of particles which are resampled together can change from one time step to the next (including their size, N).

Let K be the total number of particles. The interaction pattern for resampling is specified by means of a sequence of Markov transition matrices $\alpha_t = [\alpha_t^{ij}]$ where $1 \leq i \leq K$ and $1 \leq j \leq K$ are the row and column indices, respectively. Since α_t is a Markov matrix, it satisfies $\sum_{j=1}^K \alpha_t^{ij} = 1$ for every row i . The i th row in α_t determines from which subset of particles we resample $x_t^{(i)}$. The general α -SMC method is outlined below. We assume that either the sequence α_t is pre-determined or there is some prescribed rule to select α_t given the observations $y_{1:t}$ and the particles $\{\bar{x}_t^{(k)}\}_{1 \leq k \leq K}$.

Algorithm 4: α -SMC algorithm

1. Initialisation. At time $t = 0$, draw K i.i.d. samples, $x_0^{(k)}$, from the prior τ_0 and set $w_0^{(k)} = \frac{1}{K}$ for $k = 1, \dots, K$.
2. Recursive step. Let $\{x_{t-1}^{(k)}, w_{t-1}^{(k)}\}_{1 \leq k \leq K}$ be the particles and weights at time $t - 1$.

- (a) For $i = 1, \dots, K$, draw $\bar{x}_t^{(i)}$ from the transition kernel $\tau_t(\cdot | x_{t-1}^{(i)})$.
- (b) Select the matrix α_t and compute the resampling weights

$$a_t^{(k,j)} = \frac{\alpha_t^{kj} w_{t-1}^{(j)} g_t^{y_t}(\bar{x}_t^{(j)})}{\sum_{i=1}^K \alpha_t^{ki} w_{t-1}^{(i)} g_t^{y_t}(\bar{x}_t^{(i)})},$$

$$k = 1, \dots, K \text{ and } j = 1, \dots, K.$$

- (c) For $k = 1, \dots, K$, set $x_t^{(k)} = \bar{x}_t^{(j)}$ with probability $a_t^{(k,j)}$, $j \in \{1, \dots, K\}$, and let

$$w_t^{(k)} = \frac{\sum_{i=1}^K \alpha_t^{ki} w_{t-1}^{(i)} g_t^{y_t}(\bar{x}_t^{(i)})}{\sum_{l=1}^K \sum_{i=1}^K \alpha_t^{li} w_{t-1}^{(i)} g_t^{y_t}(\bar{x}_t^{(i)})}.$$

The particle approximation of π_t produced by Algorithm 4 is $\pi_t^K = \sum_{k=1}^K w_t^{(k)} \delta_{x_t^{(k)}}$. The α -SMC scheme can be particularised to yield most standard particle filtering algorithms ([12] Section 2.2). Of specific interest for the purpose of parallelisation is that the DRNA-based PF (Algorithm 2) can also be described and analysed as an α -SMC procedure [11].

The convergence of α -SMC methods depends on the choice of the sequence of interaction matrices α_t . Let us recursively define the matrices $\alpha_{t,t} = \mathbf{I}_K$ (where \mathbf{I}_K denotes the identity matrix) and $\alpha_{s,t}$, constructed entry-wise as $\alpha_{s,t}^{ij} = \sum_{k=1}^K \alpha_{s+1,t}^{ik} \alpha_s^{kj}$, for $i, j \in \{1, \dots, K\}$ and $0 \leq s < t$. Furthermore, define $\beta_{s,t}^i = \frac{1}{K} \sum_{j=1}^K \alpha_{s,t}^{ji}$, for

$i = 1, \dots, K$ and $0 \leq s \leq t$. Then, we have the following result, proved in [12] (Section 3).

Theorem 3 Assume that $g_t^{y_t}$ is positive and bounded for every $t \geq 1$. If the coefficients $\{\beta_{s,t}^i\}_{1 \leq i \leq K}$ are measurable w.r.t. the trivial σ -algebra $\{\mathcal{X}, \emptyset\}$ and $\lim_{K \rightarrow \infty} \max_{i \in \{1, \dots, K\}} \beta_{s,t}^i = 0$ for all $0 \leq s \leq t$ then

$$\lim_{K \rightarrow \infty} E \left[\left| (f, \pi_t) - (f, \pi_t^K) \right|^p \right] = 0$$

for any $f \in B(\mathcal{X})$ and $p \geq 1$.

4 Error rates for ensembles of non-interacting particle filters

4.1 Averaged estimators

We turn our attention to the analysis of the ensemble of non-interacting PFs outlined in Section 3.1. In particular, we study the accuracy of the particle approximations $\pi_t^{m,N}$ and $\pi_t^{M \times N}$ introduced in Eqs. (12) and (13), respectively. We adopt the mean square error (MSE) for integrals of bounded real functions,

$$\text{MSE} \equiv E \left[\left((f, \pi_t^{M \times N}) - (f, \pi_t) \right)^2 \right], \quad f \in B(\mathcal{X}),$$

as a performance metric. Since the underlying state-space model is the same for all filters and they are run in a completely independent manner, the measured-valued random variables $\pi_t^{m,N}$, $m = 1, \dots, M$, are i.i.d., and it is straightforward to show (via Lemma 1) that

$$E \left[\left((f, \pi_t^{M \times N}) - (f, \pi_t) \right)^2 \right] \leq \frac{c_t^2 \|f\|_\infty^2}{MN}, \quad (15)$$

for some constant t independent of N and M . However, the inequality (15) does not illuminate the effect of the choice of N . In the extreme case of $N = 1$, for example, $\pi_t^{M \times N}$ reduces to the outcome of a sequential importance sampling algorithm, with no resampling, which is known to degenerate quickly in practice. Instead of (15), we seek a bound for the approximation error that provides some indication on the trade-off between the number of independent filters, M , and the number of particles per filter, N .

With this purpose, we tackle the classical decomposition of the MSE in variance and bias terms. First, we obtain preliminary results that are needed for the analysis of the average measure $\pi_t^{M \times N}$. In particular, we prove that the random non-normalised measure ρ_t^N produced by the bootstrap filter (Algorithm 1) is unbiased and attains L_p error rates proportional to $\frac{1}{\sqrt{N}}$, i.e. the same as ξ_t^N and π_t^N . We use these results to derive an upper bound for the bias of π_t^N which is proportional to $\frac{1}{N}$. The latter enables us to deduce an upper bound for the MSE of the ensemble approximation $\pi_t^{M \times N}$ consisting of two additive terms that depend explicitly on M and N . Specifically, we show

that the variance component of the MSE decays linearly with the total number of particles, $K = MN$, while the bias term decreases with N^2 , i.e. quadratically with the number of particles per filter.

4.2 Assumptions on the state space model

All the results to be introduced in the rest of Section 4 hold under the (mild) assumptions of Lemma 1, which we summarise below for convenience of presentation.

Assumption 1 *The sequence of observations $Y_{1:T} = y_{1:T}$ is arbitrary but fixed, with $T < \infty$.*

Assumption 2 *The likelihood functions are bounded and positive, i.e.*

$$g_t^{y_t} \in B(\mathcal{X}) \quad \text{and} \quad g_t^{y_t} > 0 \quad \text{for every } t = 1, 2, \dots, T.$$

Remark 2 *Note that Assumptions 1 and 2 imply that*

- $(g_t^{y_t}, \alpha) > 0$, for any $\alpha \in \mathcal{P}(\mathcal{X})$, and
- $\prod_{k=1}^T g_k^{y_k} \leq \prod_{k=1}^T \|g_k^{y_k}\|_\infty < \infty$,

for every $t = 1, 2, \dots, T$.

Remark 3 *We seek simple convergence results for a fixed time horizon $T < \infty$, similar to Lemma 1. Therefore, no further assumptions related to the stability of the optimal filter for the state-space model [4, 25] are needed. If such assumptions are imposed then stronger (time uniform) asymptotic convergence can be proved, similar to Theorem 1 in Section 3.2. See [11] for additional results that apply to the independent filters $\pi_t^{m,N}$ and the ensemble $\pi_t^{M \times N}$.*

4.3 Bias and error rates

Our analysis relies on some properties of the particle approximations of the non-normalised measures $\rho_t, t \geq 1$. We first show that the estimate ρ_t^N in Eq. (8) is unbiased.

Lemma 2 *If Assumptions 1 and 2 hold, then*

$$E[(f, \rho_t^N)] = (f, \rho_t)$$

for any $f \in B(\mathcal{X})$ and every $t = 1, 2, \dots, T$.

Proof See Appendix 1 for a self-contained proof. \square

Remark 4 *The result in Lemma 2 was originally proved in [4]. For the case $\mathbf{1}(x) = 1$, it states that the estimate $(\mathbf{1}, \rho_t^N)$ of the proportionality constant of the posterior distribution π_t is unbiased. This property is at the core of recent model inference algorithms such as particle MCMC [27], SMC² [28] or some population Monte Carlo [29] methods.*

Combining Lemma 2 with the standard result of Lemma 1 leads to an explicit convergence rate for the L_p norms of the approximation errors $(f, \rho_t^N) - (f, \rho_t)$.

Lemma 3 *If Assumptions 1 and 2 hold, then, for any $f \in B(\mathcal{X})$, any $p \geq 1$ and every $t = 1, 2, \dots, T$, we have the inequality*

$$\|(f, \rho_t^N) - (f, \rho_t)\|_p \leq \frac{\tilde{c}_t \|f\|_\infty}{\sqrt{N}}, \quad (16)$$

where $\tilde{c}_t < \infty$ is a constant independent of N .

Proof See Appendix 2. \square

Finally, Lemmas 2 and 3 together enable the calculation of explicit rates for the bias of the particle approximation of (f, π_t) . This is a key result for the decomposition of the MSE into variance and bias terms. To be specific, we can prove the following theorem.

Theorem 4 *If $0 < (\mathbf{1}, \rho_t) < \infty$ for $t = 1, 2, \dots, T$ and Assumptions 1 and 2 hold, then, for any $f \in B(\mathcal{X})$ and every $0 \leq t \leq T$, we obtain*

$$|E[(f, \pi_t^N) - (f, \pi_t)]| \leq \frac{\hat{c}_t \|f\|_\infty}{N},$$

where $\hat{c}_t < \infty$ is a constant independent of N .

Proof Let us first note that $(f, \pi_t) = (f, \rho_t) / (\mathbf{1}, \rho_t)$ and

$$(f, \pi_t^N) = \frac{(f, \rho_t^N)}{G_t^N} \quad (17)$$

$$= \frac{(f, \rho_t^N)}{G_t^N(\mathbf{1}, \pi_t^N)} \quad (18)$$

$$= \frac{(f, \rho_t^N)}{(\mathbf{1}, \rho_t^N)}, \quad (19)$$

where (17) follows from the construction of ρ_t^N , (18) holds because $(\mathbf{1}, \pi_t^N) = 1$ and (19) is, again, a consequence of the definition of ρ_t^N . Therefore, the difference $(f, \pi_t^N) - (f, \pi_t)$ can be written as

$$(f, \pi_t^N) - (f, \pi_t) = \frac{(f, \rho_t^N)}{(\mathbf{1}, \rho_t^N)} - \frac{(f, \rho_t)}{(\mathbf{1}, \rho_t)}$$

and, since $(f, \rho_t) = E[(f, \rho_t^N)]$ (from Lemma 2), the bias can be expressed as

$$E[(f, \pi_t^N) - (f, \pi_t)] = E \left[\frac{(f, \rho_t^N)}{(\mathbf{1}, \rho_t^N)} - \frac{(f, \rho_t^N)}{(\mathbf{1}, \rho_t)} \right]. \quad (20)$$

Some elementary manipulations on (20) yield the equality

$$E[(f, \pi_t^N) - (f, \pi_t)] = E\left[(f, \pi_t^N) \frac{(\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N)}{(\mathbf{1}, \rho_t)}\right]. \quad (21)$$

If we realise that $E[(\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N)] = 0$ (again, a consequence of Lemma 2) and move the factor $(\mathbf{1}, \rho_t)^{-1}$ out of the expectation, then we easily rewrite Eq. (21) as

$$\begin{aligned} E[(f, \pi_t^N) - (f, \pi_t)] &= \frac{1}{(\mathbf{1}, \rho_t)} E[(f, \pi_t^N) ((\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N))] \\ &\quad - \frac{(f, \pi_t)}{(\mathbf{1}, \rho_t)} E[(\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N)] \\ &= \frac{1}{(\mathbf{1}, \rho_t)} E[((f, \pi_t^N) - (f, \pi_t)) \\ &\quad \quad \quad ((\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N))] \\ &\leq \frac{1}{(\mathbf{1}, \rho_t)} \sqrt{E[(f, \pi_t^N) - (f, \pi_t)]^2} \\ &\quad \times \sqrt{E[(\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N)]^2} \quad (22) \\ &\leq \frac{1}{(\mathbf{1}, \rho_t)} \left(\frac{c_t \|f\|_\infty}{N} \times \frac{\tilde{c}_t}{N} \right) = \frac{\hat{c}_t \|f\|_\infty}{N}, \quad (23) \end{aligned}$$

where we have applied the Cauchy-Schwartz inequality to obtain (22), (23) follows from Lemmas 1 and 3 and

$$\hat{c}_t = \frac{c_t \tilde{c}_t \|f\|_\infty}{(\mathbf{1}, \rho_t)} < \infty$$

is a constant independent of N . \square

The result in Theorem 4 was originally proved in [30], albeit by a different method.

For any $f \in B(\mathcal{X})$, let $\mathcal{E}_t^N(f)$ denote the approximation difference, i.e.

$$\mathcal{E}_t^N(f) \triangleq (f, \pi_t^N) - (f, \pi_t).$$

This is a r.v. whose second-order moment yields the MSE of (f, π_t^N) . It is straightforward to obtain a bound for the MSE from Lemma 1 and, by subsequently using Theorem 4, we readily find a similar bound for the variance of $\mathcal{E}_t^N(f)$, denoted $\text{Var}[\mathcal{E}_t^N(f)]$. These results are explicitly stated by the corollary below.

Corollary 1 *If $0 < (\mathbf{1}, \rho_t) < \infty$ for $t = 1, 2, \dots, T$ and Assumptions 1 and 2 hold, then, for any $f \in B(\mathcal{X})$ and any $0 \leq t \leq T$, we obtain*

$$E[(\mathcal{E}_t^N(f))^2] \leq \frac{c_t^2 \|f\|_\infty^2}{N} \quad \text{and} \quad (24)$$

$$\text{Var}[\mathcal{E}_t^N(f)] \leq \frac{(c_t^v)^2 \|f\|_\infty^2}{N}, \quad (25)$$

where c_t and c_t^v are finite constants independent of N .

Proof The inequality (24) for the MSE is a straightforward consequence of Lemma 1. Moreover, we can write the MSE in terms of the variance and the square of the bias, which yields

$$E[(\mathcal{E}_t^N(f))^2] = \text{Var}[\mathcal{E}_t^N(f)] + E^2[\mathcal{E}_t^N] \leq \frac{c_t^2 \|f\|_\infty^2}{N}. \quad (26)$$

Since Theorem 4 ensures that $|E[\mathcal{E}_t^N]| \leq \frac{\hat{c}_t \|f\|_\infty}{N}$, then the inequality (26) implies that there exists a constant $c_t^v < \infty$ such that (25) holds. \square

4.4 Error rate for the averaged estimators

Let us run M independent PFs with the same (fixed) sequence of observations $Y_{1:T} = y_{1:T}$, $T < \infty$, and N particles each. The random measures output by the m th filter are denoted

$$\xi_t^{m,N}, \quad \pi_t^{m,N} \quad \text{and} \quad \rho_t^{m,N}, \quad \text{with } m = 1, 2, \dots, M.$$

Obviously, all the theoretical properties established in Section 4.3, as well as the basic Lemma 1, hold for each one of the M independent filters.

Definition 1 *The ensemble approximation of π_t with M independent filters is the discrete random measure $\pi_t^{M \times N}$ constructed as*

$$\pi_t^{M \times N} = \frac{1}{M} \sum_{m=1}^M \pi_t^{m,N},$$

and the averaged estimator of (f, π_t) is $(f, \pi_t^{M \times N})$.

It is apparent that similar ensemble approximations can be given for ξ_t and ρ_t . Moreover, the statistical independence of the PFs yields the following corollary as a straightforward consequence of Theorem 4 and Corollary 1.

Corollary 2 *If $0 < (\mathbf{1}, \rho_t) < \infty$ for $t = 1, 2, \dots, T$ and Assumptions 1 and 2 hold, then, for any $f \in B(\mathcal{X})$ and any $0 \leq t \leq T$, the inequality*

$$E\left[\left((f, \pi_t^{M \times N}) - (f, \pi_t)\right)^2\right] \leq \frac{(c_t^v)^2 \|f\|_\infty^2}{MN} + \frac{\hat{c}_t^2 \|f\|_\infty^2}{N^2} \quad (27)$$

holds for some constants c_t^v and \hat{c}_t independent of N and M .

Proof Let us denote

$$\mathcal{E}_t^{M \times N}(f) = (f, \pi_t^{M \times N}) - (f, \pi_t) \quad \text{and}$$

$$\mathcal{E}_t^{m,N}(f) = (f, \pi_t^{m,N}) - (f, \pi_t)$$

for $m = 1, 2, \dots, M$. Since $\pi_t^{M \times N}$ is a linear combination of i.i.d. random measures, we easily obtain that

$$\begin{aligned} \left| E \left[\mathcal{E}_t^{M \times N}(f) \right] \right|^2 &= \left| \frac{1}{M} \sum_{m=1}^M E \left[\mathcal{E}_t^{m,N}(f) \right] \right|^2 \\ &= \left| E \left[\mathcal{E}_t^{m,N}(f) \right] \right|^2 \\ &\leq \frac{\hat{c}_t \|f\|_\infty}{N}, \quad \text{for any } m \leq M, \end{aligned} \quad (28)$$

where the inequality follows from Theorem 4. Moreover, again because of the independence of the random measures, we readily calculate a bound for the variance of $\mathcal{E}_t^{M \times N}(f)$,

$$\text{Var} \left[\mathcal{E}_t^{M \times N}(f) \right] = \frac{1}{M} \text{Var} \left[\mathcal{E}_t^{m,N}(f) \right] \leq \frac{(c_t^v)^2 \|f\|_\infty^2}{MN}, \quad (29)$$

where the inequality follows from Corollary 1. Since $E \left[\left(\mathcal{E}_t^{M \times N} \right)^2 \right] = \text{Var} \left[\mathcal{E}_t^{M \times N} \right] + \left| E \left[\mathcal{E}_t^{M \times N} \right] \right|^2$, combining (29) and (28) yields (27) and concludes the proof. \square

The inequality in Corollary 2 shows explicitly that the bias of the estimator $(f, \pi_t^{M \times N})$ cannot be arbitrarily reduced when N is fixed, even if $M \rightarrow \infty$. This feature is already discussed in Section 3.3. Note that the inequality (27) holds for any choice of M and N , while Theorem 2 yields asymptotic limits.

Remark 5 According to the inequality (27), the bias of the estimator $(f, \pi_t^{M \times N})$ is controlled by the number of particles per subset, N , and converges quadratically, while, for fixed N , the variance decays linearly with M . The MSE rate is $\propto \frac{1}{MN}$ as long as $N \geq M$. Otherwise, the term $\frac{\hat{c}_t^2 \|f\|_\infty^2}{N^2}$ becomes dominant and the resulting asymptotic error bound turns out higher.

Remark 6 While the convergence results presented here have been proved for the standard bootstrap filter, it is straightforward to extend them to other classes of PFs for which Lemmas 1 and 2 hold.

4.5 Comparison of parallelisation schemes via time–error indices

The advantage of parallel computation is the drastic reduction of the time needed to run the PF. Let the running time for a PF with K particles be of order $\mathcal{T}(K)$, where $\mathcal{T} : \mathbb{N} \rightarrow (0, \infty)$ is some strictly increasing function of K . The quantity $\mathcal{T}(K)$ includes the time needed to generate new particles, weight them and perform resampling. The latter step is the bottleneck for parallelisation, as it requires the interaction of all K particles. Also, a ‘straightforward’ implementation of the resampling step leads to

an execution time $\mathcal{T}(K) = K \log(K)$, although efficient algorithms exist that achieve to a linear time complexity, $\mathcal{T}(K) = K$. We can combine the MSE rate and the time complexity to propose a time–error performance metric.

Definition 2 We define the time–error index of a particle filtering algorithm with running time of order \mathcal{T} and asymptotic MSE rate \mathcal{R} as $\mathcal{C} \triangleq \mathcal{T} \times \mathcal{R}$.

The smaller the index \mathcal{C} for an algorithm, the more (asymptotically) efficient its implementation. For the standard (centralised) bootstrap filter (see Algorithm 1) with K particles, the running time is of order $\mathcal{T}(K) = K$ and the MSE rate is of order $\mathcal{R}(K) = \frac{1}{K}$; hence, the time–error index becomes

$$\mathcal{C}_{bf}(K) = \mathcal{T}(K) \times \mathcal{R}(K) = 1.$$

For the computation of the ensemble approximation $\pi_t^{M \times N}$, we can run M independent PFs in parallel, with $N = K/M$ particles each and no interaction among them. Hence, the execution time becomes of order $\mathcal{T}(M, N) = N$. Since the error rate for the ensemble approximation is of order $\mathcal{R}(M, N) = \left(\frac{1}{MN} + \frac{1}{N^2} \right)$, the time–error index of the ensemble approximation is

$$\mathcal{C}_{ens}(M, N) = \mathcal{T}(M, N) \times \mathcal{R}(M, N) = \frac{1}{M} + \frac{1}{N}$$

and hence it vanishes with $M, N \rightarrow \infty$. In particular, since we have to choose $N \geq M$ to ensure a rate of order $\frac{1}{MN}$, then $\lim_{M \rightarrow \infty} \mathcal{C}_{ens} = 0$. In any case, whenever $N > 1$ it is apparent that $\mathcal{C}_{ens} < \mathcal{C}_{bf}$.

We have described alternative ensemble approximations where M non-independent PFs are run with N particles each in Section 3. The overall error rates for these methods are same as for the standard bootstrap filter; however, the time complexity depends not only on the number of particles N allocated to each of the M subsets, but also on the subsequent interactions among subsets.

Let us consider, for example, the double bootstrap algorithm with adaptive selection of [13] (namely, [13] (Algorithm 4)). This is a scheme where

- M bootstrap filters (as Algorithm 1 in this paper) are run in parallel and an aggregate weight is computed for each one of them, denoted $W_t^{(m)}$;
- When the coefficient of variation (CV) of these aggregate weights is greater than a given threshold, the M bootstrap filters are resampled (some filters are discarded and others are replicated using a multinomial resampling procedure).

See [13] (Section 4.2) for details. Assuming that the resampling procedure in the second step above (termed

island-level resampling in [13]) is performed, in the average, once every L time steps, then the running time for this algorithm is

$$\mathcal{T}(M, N, L) = \frac{L-1}{L}N + \frac{1}{L}MN = \frac{N}{L}(M+L-1),$$

while the approximation error is $\mathcal{R}(M, N) = \frac{1}{MN}$ (see ([13] Theorem 5)). Hence, the time-error index for this double bootstrap algorithm is

$$\mathcal{C}_{dbf} = \mathcal{T}(M, N, L) \times \mathcal{R}(M, N) = \frac{M+L-1}{LM}.$$

When $L \ll N$, we readily obtain that $\mathcal{C}_{ens} < \mathcal{C}_{dbf}$. For example, for a configuration with $M = 10$ filters and $N = 100$ particles each and assuming that island-level resampling is performed every $L = 20$ time steps on average, then $\mathcal{C}_{dpf} = 0.145$ and $\mathcal{C}_{ens} = 0.110$. On the contrary, if L is large enough (namely, if $L > N(M-1)/M$), the double bootstrap algorithm becomes more efficient, meaning that $\mathcal{C}_{dbf} < \mathcal{C}_{ens}$.

Computing the time-error index for practical algorithms can be hard and highly dependent on the specific implementation. Different implementations of the double bootstrap algorithm, for example, may yield different time-error indices depending on how the island-level resampling step is carried out.

5 Numerical results and discussion

5.1 Example: Lorenz 63 model

5.1.1 The three-dimensional Lorenz system

Let us consider the problem of tracking the state of a three-dimensional Lorenz system [31] with additive dynamical noise and partial observations [32]. To be specific, consider a three-dimensional stochastic process $\{X(s)\}_{s \in (0, \infty)}$ (s denotes continuous time) taking values on \mathbb{R}^3 , which dynamics is described by the system of stochastic differential equations

$$\begin{aligned} dX_1 &= -s(X_1 - Y_1) + dW_1, \\ dX_2 &= rX_1 - X_2 - X_1X_3 + dW_2, \\ dX_3 &= X_1X_2 - bX_3 + dW_3, \end{aligned}$$

where $\{W_i(s)\}_{s \in (0, \infty)}$, $i = 1, 2, 3$, are independent one-dimensional Wiener processes and

$$(s, r, b) = \left(10, 28, \frac{8}{3}\right)$$

are static model parameters³ that yield chaotic dynamics. A discrete-time version of the latter system using Euler's method with integration step $T_d = 10^{-3}$ is straightforward to obtain and yield the model

$$X_{1,n} = X_{1,n-1} - T_d s(X_{1,n-1} - X_{2,n-1}) + \sqrt{T_d} U_{1,n}, \tag{30}$$

$$X_{2,n} = X_{2,n-1} + T_d (rX_{1,n-1} - X_{2,n-1} - X_{1,n-1}X_{3,n-1}) + \sqrt{T_d} U_{2,n}, \tag{31}$$

$$X_{3,n} = X_{3,n-1} + T_d (X_{1,n-1}X_{2,n-1} - bX_{3,n-1}) + \sqrt{T_d} U_{3,n}, \tag{32}$$

where $\{U_{i,n}\}_{n=0,1,\dots}$, $i = 1, 2, 3$, are independent sequences of i.i.d. normal random variables with 0 mean and variance 1. System (30)–(32) is partially observed every 100 discrete-time steps. Specifically, we collect a sequence of scalar observations $\{Y_t\}_{t=1,2,\dots}$, of the form

$$Y_t = X_{1,100t} + V_t, \tag{33}$$

where $\{V_t\}_{t=1,2,\dots}$ is a sequence of i.i.d. normal random variables with zero mean and variance $\sigma^2 = \frac{1}{2}$.

Let $X_n = (X_{1,n}, X_{2,n}, X_{3,n}) \in \mathbb{R}^3$ be the state vector at discrete time n . The dynamic model given by Eqs. (30)–(32) yields the family of kernels $\tau_{n,\theta}(dx|x_{n-1})$, and the observation model of Eq. (33) yields the likelihood function

$$g_{t,\theta}^{y_t}(x_{100t}) \propto \exp \left\{ -\frac{1}{2\sigma^2} (y_t - x_{1,100t})^2 \right\},$$

both in a straightforward manner. The goal is to track the sequence of joint posterior probability measures π_t , $t = 1, 2, \dots$, for $\{\hat{X}_t\}_{t=1,\dots}$, where $\hat{X}_t = X_{100t}$. Note that one can draw a sample $\hat{X}_t = \hat{x}_t$ conditional on $\hat{X}_{t-1} = \hat{x}_{t-1}$ by successively simulating

$$\tilde{x}_n \sim \tau_{n,\theta}(dx|\tilde{x}_{n-1}), \quad n = 100(t-1) + 1, \dots, 100t,$$

where $\tilde{x}_{100(t-1)} = \hat{x}_{t-1}$ and $\hat{x}_t = \tilde{x}_{100t}$. The prior measure for the state variables is normal, namely $\hat{X}_0 \sim \mathcal{N}(x_*, v_0^2 \mathcal{I}_3)$, where $x_* = (-10.2410; -1.3984; -23.6752)$ is the mean⁴ and $v_0^2 \mathcal{I}_3$ is the covariance matrix, with $v_0^2 = 10$ and \mathcal{I}_3 the three-dimensional identity matrix.

5.1.2 Simulation setup

We aim at illustrating the gain in relative performance, taking into account both estimation errors and running time, that can be attained using ensembles of independent PFs. With this purpose, we have applied

- The standard bootstrap filter (Algorithm 1), termed BF in the sequel, and
- The ensemble of non-interacting bootstrap filters (NIBFs) that we have investigated in Section 4

to track the sequence of probability measures π_t generated by the three-dimensional Lorenz model described in Section 5.1.1. We have generated a sequence of 200 synthetic observations, $\{y_t; t = 1, \dots, 200\}$, spread over an interval of 20 continuous time units, corresponding to

2×10^4 discrete time steps in the Euler scheme (hence, one observation every 100 steps).

The ensemble of NIBFs consists of M filters with N particles each, while the standard BF runs with K particles, where $K = MN$ for a fair comparison.

We have coded the three algorithms in Matlab (version 7.11.0.584 [R2010b] with the parallel computing toolbox) and run the experiments using a pool of identical multi-processor machines, each one having 8 cores at 3.16 GHz and 32 GB of RAM memory. The standard (centralised) BF is run with $K = MN$ particles in a single core. For the ensemble of NIBFs, we allow the parallel computing toolbox to allocate all available cores per server in order to run all BFs concurrently.

To assess the approximation errors, we have computed empirical MSEs for the approximation of the posterior mean, $E[\hat{X}_t | Y_{1:t}] = (I, \pi_t)$, where $I(x) = x$ is the identity function, for the two algorithms at the last update step, $t = 200$. Note, however, that the integral (I, π_t) cannot be computed in closed form for this system. Therefore, we have used the ‘expensive’ estimate

$$(I, \pi_t) \approx (I, \pi_t^J), \quad \text{with } J = 10^5 \text{ particles,}$$

computed via the standard BF, as a proxy of the true value.

5.1.3 Numerical results

Figure 1 displays the empirical MSE, averaged over 100 independent simulation runs, attained by the parallel schemes when the number of filters is fixed, $M = 20$, and the number of particles per filter (particle island) ranges from $N = 100$ to $N = 1000$. The outcome of the

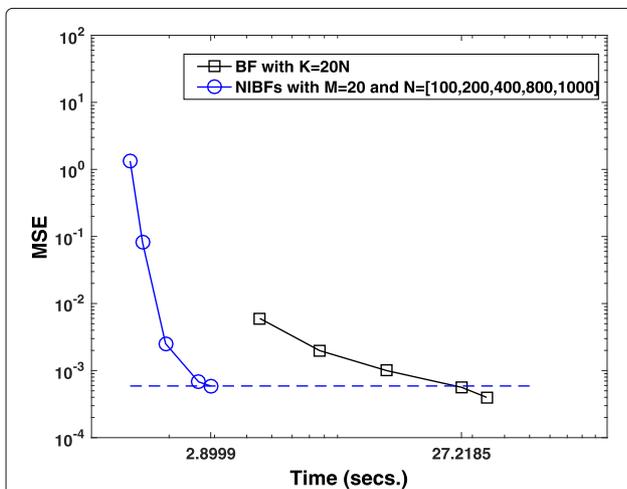


Fig. 1 Empirical mean of the MSE for the centralised BF, with $K = MN$ particles, and the ensemble of NIBFs, with $M = 20$ constant and $N = 100, 200, 400, 800, 1000$. All curves have been obtained from a set of 100 independent simulation trials. Note that the centralised BF is run with $K = 20N$ particles, where N takes values in the same way as for the parallel algorithm

centralised BF with $K = MN$ particles, hence ranging from $K = 20 \times 100$ to $K = 20 \times 1000$, is also shown for comparison. We observe that proposed ensemble of NIBFs achieves a poor performance when the number of particles per filter, N , is relatively low ($N = 100$), while for moderate values ($N \geq 400$) it nearly matches the MSE of the centralised BF.

Next, we look into the relationship between the MSE and the running time for the two algorithms. With the number of filters $M = 20$ fixed, we have run 100 independent simulation trials for each value $N = 100, 200, 400, 800$ and 1000 and computed the empirical MSE and the average running time for the parallel scheme and each combination of M and N . Correspondingly, we have also run the centralised BF with $K = MN$ particles, hence for $K = 2 \times 10^3, 4 \times 10^3, 8 \times 10^3, 16 \times 10^3$ and 20×10^3 .

Figure 2 displays the resulting empirical MSE versus the running time for the two methods. If we qualify an algorithm as more efficient than another one when it is capable of attaining a lower MSE in the same amount of time, then this set of simulations shows that the independent ensemble scheme is more efficient than the centralised BF. Indeed, a close look at Fig. 2 reveals that the ensemble of $M = 20$ NIBFs with $N = 1000$ particles per filter achieves an empirical MSE of $\approx 6 \times 10^{-4}$ with a running time of ≈ 2.9 s, while the centralised BF attains the same performance with $K = 20 \times 800$ particles and a running time of ≈ 27.2 s (as shown by the dashed horizontal line in the plot).

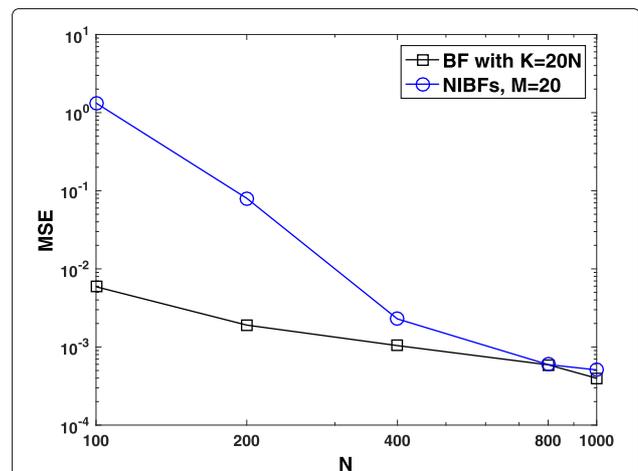


Fig. 2 Empirical mean of the MSE versus the running time for the centralised BF and the independent ensemble of BFs. The parallel scheme is run with $M = 20$ constant and $N = 100, 200, 400, 600, 800$ and 1000 . The centralised BF is run with $K = 20N$ particles, where N takes values in the same way as for the parallel algorithm. The dashed horizontal lines indicate where the mean of the MSE match for the independent ensemble and the centralised BF. The running times for the two algorithms at that MSE level are shown as labels on the horizontal axis

5.2 Example: Lorenz 96 model

5.2.1 The J -dimensional Lorenz 96 system

The Lorenz 96 model is a deterministic system of non-linear differential equations that displays chaotic dynamics [33, 34]. The system dimension, i.e. the number of dynamic variables, can be scaled arbitrarily. A stochastic version of the model can be easily obtained by converting each differential equation into a stochastic differential equation driven by an independent and additive Wiener process. In particular, a model with J variables, Z_j , $j = 0, \dots, J - 1$, can be written down as the system of stochastic differential equations

$$dZ_j = -Z_{j-1} (Z_{j-2} - Z_{j+1}) - Z_j + F + \sigma dW_j, \quad j = 1, \dots, J, \quad (34)$$

where $F = 8$ is a constant forcing parameter⁵, the Wiener processes $\{W_j(s)\}_{l,j \geq 0}$ are assumed independent and the scale parameter σ is known.

A straightforward application of the Euler-Maruyama integration method yields a discrete-time version of the stochastic, two-scale Lorenz 96 model. If we let $T_d > 0$ denote the discretisation period and n denotes discrete time, then we readily obtain

$$Z_{j,n} = Z_{j,n-1} - T_d [Z_{j-1,n-1} (Z_{j-2,n-1} - Z_{j+1,n-1}) - F + Z_{j,n-1}] + \sqrt{T_d} \sigma U_{j,n}, \quad (35)$$

where $j = 0, \dots, J - 1$ and $\{U_{j,n}\}_{l,j,n \geq 0}$ are independent and identically distributed (i.i.d.) standard Gaussian r.v.s.

We assume that observations can only be collected from this system once every n_0 discrete time steps. Moreover, only the variables with even indices ($j = 0, 2, 4, \dots, J$, for even J) are measured. Therefore, the observation process has the form

$$Y_t = [Z_{0,n_0 t}, Z_{2,n_0 t}, \dots, Z_{J,n_0 t}]^\top + V_t, \quad (36)$$

where $t = 1, 2, \dots$ and $\{V_t\}_{t \geq 1}$ is a sequence of i.i.d. r.v.s with common pdf $\mathcal{N}(v_t; 0, \sigma_y^2 \mathcal{I}_{\frac{J}{2}})$, which denotes a $\frac{J}{2}$ -dimensional Gaussian distribution with 0 mean and covariance matrix $\sigma_y^2 \mathcal{I}_{\frac{J}{2}}$.

Equations 35 and (36) describe a state space model that can be expressed in terms of the general notation in Section 2. The state process at time t is $\tilde{X}_n = [Z_{0,n}, \dots, Z_{J-1,n}]^\top$ and the transition kernel from time $n - 1$ to time n is

$$\tilde{\tau}_n(d\tilde{x}_n | \tilde{x}_{n-1}) = \mathcal{N}(\tilde{x}_n; \Psi(\tilde{x}_{n-1}), \sigma_x^2 \mathcal{I}_J) d\tilde{x}_n, \quad (37)$$

where $\mathcal{N}(x; \mu, \Sigma)$ is the Gaussian density with argument x , mean μ and covariance matrix Σ , $\sigma_x^2 = T_d \sigma^2$ and $\Psi : \mathbb{R}^J \rightarrow \mathbb{R}^J$ is the deterministic transformation that accounts for all the terms on the right hand side of (35)

except the noise contribution $\sqrt{T_d} \sigma U_{j,n}$. Since we only collect observations every $n_0 T_d$ continuous-time units, we need to put the dynamics of the states on the same time scale as the observation process $\{Y_t\}_{t \geq 1}$ in Eq. (36). If we define $X_t = \tilde{X}_{n_0 t}$ then the transition kernel from X_{t-1} to X_t follows readily from (37),

$$\tau_t(dx_t | x_{t-1}) = \int \cdots \int \tilde{\tau}_{n_0 t}(d\tilde{x}_t | \tilde{x}_{n_0 t-1}) \prod_{i=1}^{n_0-2} \tilde{\tau}_{n_0 t-i}(\tilde{x}_{n_0 t-i} | \tilde{x}_{n_0 t-i-1}) \tilde{\tau}_{(t-1)n_0+1}(d\tilde{x}_{(t-1)n_0+1} | x_{t-1}). \quad (38)$$

While $\tau_t(x_t | x_{t-1})$ cannot be evaluated in closed form, it is straightforward to draw a sample from $X_t | x_{t-1}$ by simply running Eq. (35) n_0 times, with starting point x_{t-1} . The likelihood function is

$$g_t^{y_t}(x_t) \propto \exp \left\{ -\frac{1}{2\sigma_y^2} \sum_{r=0}^{\frac{t}{2}} (y_{r,t} - x_{2r,t})^2 \right\}.$$

5.2.2 Simulation setup

We have run 100 independent simulations of the discretised Lorenz 96 model described in Section 5.2.1 above over 20 continuous-time units, with integration step $T_d = 2 \times 10^{-4}$ (which amounts to 10^5 discrete-time steps) and, for each simulation, we have obtained noisy observations, with $\sigma_y^2 = \frac{1}{2}$ and $n_0 = 10$, according to Eq. (36). The noise-scale parameter σ in the state Eq. (35) is set as $\sigma = \frac{1}{\sqrt{2}}$, so that the noise variance becomes $\sigma_x^2 = \frac{T_d}{2}$.

The computer experiments are similar to Section 5.1. For each simulation, we have run $M = 10$ iNIBFs with N particles each versus a centralised BF with $K = 10N$ particles and used them to compute one-step-ahead predictions of the observations. In particular, at discrete-time t , we have computed predictions of the observation vector y_t , using the measures

$$\begin{aligned} \xi_t^K &= \frac{1}{K} \sum_{k=1}^K \delta_{\tilde{x}_t^{(k)}} \quad \text{and} \quad \xi_t^{M \times N} = \frac{1}{M} \sum_{m=1}^M \xi_t^{m,N} \\ &= \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N \delta_{\tilde{x}_t^{(m,n)}}, \end{aligned}$$

for the centralised BF and the NIBFs, respectively. To be specific, if $y_t = (y_{0,t}, y_{1,t}, \dots, y_{\frac{J}{2},t})$, we have computed estimates

$$y_{r,t}^K = \frac{1}{K} \sum_{k=1}^K \tilde{x}_{2r,t}^{(k)}, \quad y_{r,t}^{M \times N} = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N \tilde{x}_{2r,t}^{(m,n)};$$

and then we have averaged the quadratic errors $(y_{r,t} - y_{r,t}^K)^2$ and $(y_{r,t} - y_{r,t}^{M \times N})^2$ over r, t and 100 independent simulation runs. Finally, we have normalised the

resulting empirical MSE with respect to the observation power $\frac{2}{J} \sum_{r=0}^{\frac{J}{2}} E[y_{r,t}^2]$. Note that, in this case, we have used the actual observations generated in the simulations to obtain the errors, instead of the proxy values in Section 5.1.

The experiments have been carried out for a Lorenz 96 model with $J = 20$ variables first and then for the same model with $J = 50$ variables.

The simulations have been coded using Matlab version R2016b (64 bits), with the parallel computing toolbox enabled, on an 8-core Intel(R) Xeon(R) CPU E5-2680 v2 server, with clock frequency 2.80 GHz and 64 GB of RAM. All the results reported are averaged over 100 independent simulation runs as described above.

5.2.3 Numerical results

Figure 3 plots the normalised MSE attained by the centralised BF and the ensemble of $M = 10$ NIBFs versus N . The centralised BF is run with $K = 10N$ particles, while each one of the $M = 10$ NIBFs is run with N particles. The figure shows results for two different state-space dimensions. The solid lines correspond to a stochastic Lorenz 96 model with $J = 20$ variables. In this case, the outcome of the simulations is similar to the experiments with the Lorenz 63 system: for $K = MN$, the centralised BF attains a smaller MSE than the NIBFs, with the gap closing as N increases. The result of the experiment is different when the state space dimension is incremented to $J = 50$. In this case, the normalised MSE of the NIBFs is slightly smaller than the error of the centralised BF for $N < 1600$,

with both estimators attaining the same performance for $N = 1600$. Hence, for this example, the averaging of the NIBFs has a beneficial effect on the accuracy of the estimators, at least for certain combinations of the number of particles N and the dimension J . We have verified that the bias of the centralised estimator y_t^K is lesser than the bias of the estimator $y_t^{M \times N}$, as predicted by the theoretical analysis, while $y_t^{M \times N}$ attains a smaller empirical variance than y_t^K (at least for $N < 1600$ and $40 \leq J \leq 100$).

Figure 4 displays the results of the same computer experiment as in Fig. 3, except that instead of averaging the MSE over the 100 independent simulation runs, we display the maximum MSE, both for the centralised BF and the ensembles of NIBFs, out of the 100 simulations for each one of the values of N . We observe that the ensemble of NIBFs is more robust than the centralised BF. While for dimension $J = 20$ the centralised BF attains a clearly lower average MSE than the NIBFs, the maximum MSE turns out to be similar for both algorithms. For dimension $J = 50$, the average MSE of the NIBFs is already lower (as shown in Fig. 3) than the average MSE of the BF, and the advantage of the parallelised algorithm increases when we look at the maximum MSE.

Figure 5 plots the same normalised MSE values of Fig. 3 versus the running times of the algorithms, given in seconds, for a complete simulation with 10^5 discrete time steps. As in the experiments of Section 5.1.3, the NIBFs can attain the same MSE as the centralised BF in just a fraction of the running time. While the improvement can be, ideally, of a factor M (with $M = 10$ in this case), in practice it depends on the efficiency of the computing software. With the version of Matlab (R2016b, with the

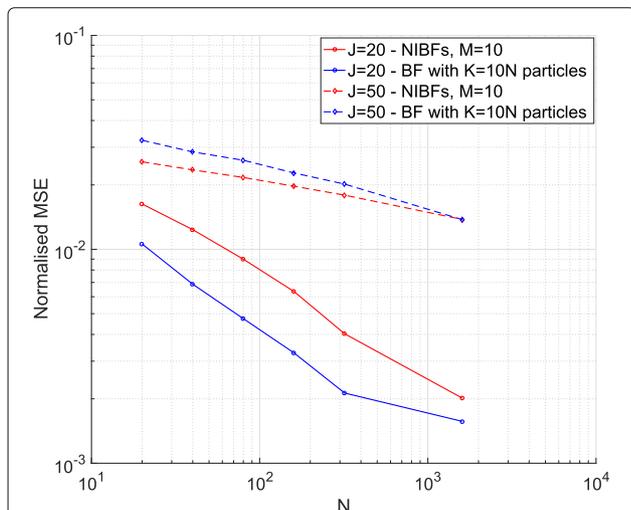


Fig. 3 Empirical mean of the normalised MSE for the centralised BF, with $K = 10N$ particles, and the average of $M = 10$ NIBFs with $N = 20, 40, 80, 160, 320, 1600$. Results are shown for a Lorenz 96 system with $J = 20$ variables (solid lines) and a Lorenz 96 model with $J = 50$ variables (dashed lines). All curves have been obtained from a set of 100 independent simulation trials

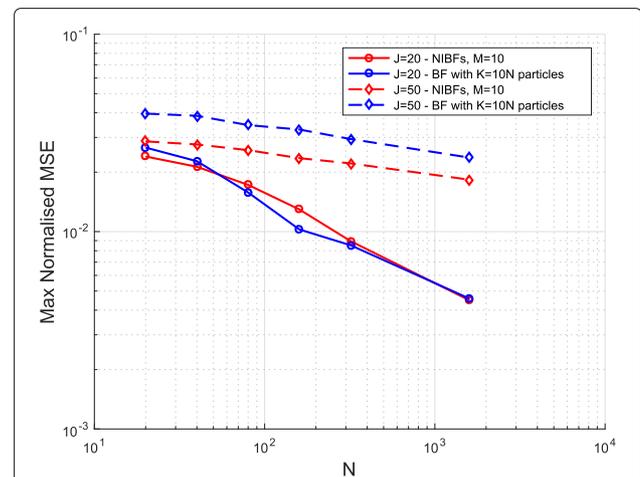
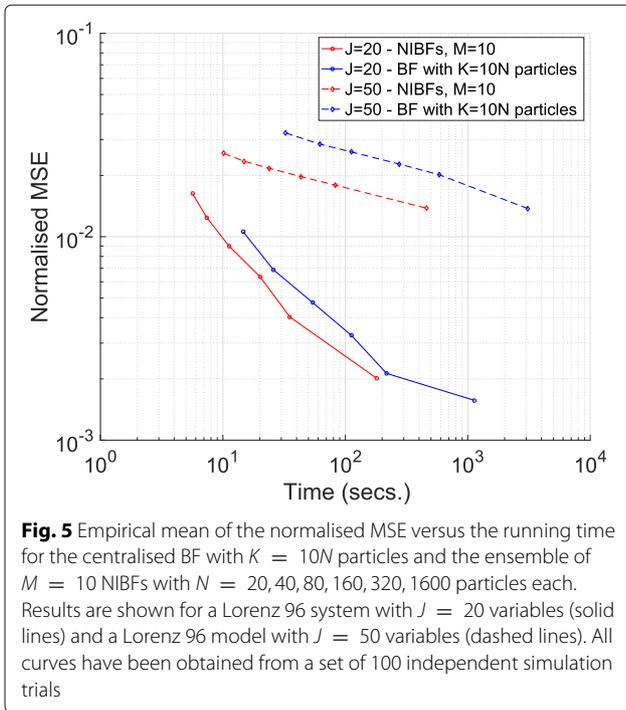


Fig. 4 Empirical maximum (out of 100 independent simulation runs) of the normalised MSE for the centralised BF, with $K = 10N$ particles, and the average of $M = 10$ NIBFs with $N = 20, 40, 80, 160, 320, 1600$. Results are shown for a Lorenz 96 system with $J = 20$ variables (solid lines) and a Lorenz 96 model with $J = 50$ variables (dashed lines)



parallelisation toolbox) and the 8-core Intel Xeon processor used in these experiments, the running time of the centralised BF with $K = 10N$ particles was reduced by a modest factor of 2.6 for $J = 20$ when using $M = 10$ parallel NIBFs with N particles each. For $J = 50$, however, the running time was reduced by a factor of 6.6. The difference is due to the ability of the Matlab software to parallelise more efficiently when handling larger vectors. From this figure, we observe that, for $J = 50$, the NIBFs attain the same minimum error as the centralised BF (a normalised MSE of ≈ 0.0138) with a running time that is 6.6 times smaller (464 versus 3,082 s).

6 Conclusions

We have presented a survey of methods for the parallelisation of particle filters. Specifically, we have described the basic parallelisation scheme based on ensembles of statistically independent PFs and then discussed three alternatives which introduce different degrees of interaction among the concurrently running filters. We have placed emphasis on the theoretical guarantees of the algorithms, and, hence, we have stated conditions for the convergence of all the techniques, including the DRNA-based PF of [2], the particle island model of [13] and the α -SMC method of [12].

In the second half of the paper, we have focused on the theoretical properties of the ensemble of non-interacting PFs. For this method, we have shown, both numerically and through the definition of time–error indices, that

the averaging of statistically independent PFs should be preferred when N , the number of particles per independent filter, can be made sufficiently large to reduce the bias. This is often the case when using many-core computers (or computing clusters). When parallelisation is implemented using many low-power devices (such as GPUs), parallelisation with interaction is more efficient. Our numerical experiments for the stochastic Lorenz 96 model also show that the averaging of independent estimators can lead to lower estimation errors, compared to a centralised bootstrap filter with the same number of particles, as the dimension of the state space is increased.

Endnotes

¹Note that G_t^N is an estimate of the normalising constant for π_t (namely, the integral $(\mathbf{1}, \rho_t)$) which can be shown to be unbiased under mild assumptions [4]. In Bayesian model selection, this constant is termed ‘model evidence’, while in parameter estimation problems, it is often referred to as the likelihood (of the unknown parameters) [27].

²Other particle filtering algorithms can be applied in a straightforward way; however, we assume bootstrap filters (i.e. the procedure of Algorithm 1) for the sake of clarity and notational simplicity.

³Note the difference in notation between the continuous time s and the parameter s .

⁴Chosen from a typical trajectory of the *deterministic* Lorenz 63 model.

⁵The deterministic Lorenz 96 system is chaotic for $F > 6$, with increasing turbulence of the chaotic flow as F is made larger.

Appendix 1 Proof of Lemma 2

We proceed by induction in the time index t . For $t = 0$, $\rho_0 = \tau_0 = \pi_0$ and, since $x_0^{(i)}$, $i = 1, \dots, N$, are drawn from π_0 , the equality $E[(f, \rho_0^N)] = (f, \rho_0)$ is straightforward.

Let us assume that

$$E[(f, \rho_{t-1}^N)] = (f, \rho_{t-1}) \tag{39}$$

for some $t > 0$ and any $f \in B(\mathcal{X})$. If we use $\bar{\mathcal{F}}_t$ to denote the σ -algebra generated by the set of random variables $\{x_{0:t-1}^{(i)}, \bar{x}_{1:t}^{(i)} : 1 \leq i \leq N\}$ then we readily find that

$$E[(f, \rho_t^N) | \bar{\mathcal{F}}_t] = E[G_t^N(f, \pi_t^N) | \bar{\mathcal{F}}_t] = G_t^N(f, \bar{\pi}_t^N), \tag{40}$$

since G_t^N is measurable w.r.t. $\bar{\mathcal{F}}_t$ and $E[(f, \pi_t^N) | \bar{\mathcal{F}}_t] = (f, \bar{\pi}_t^N)$. Moreover, if we recall that

$$\begin{aligned} (f, \bar{\pi}_t^N) &= \sum_{i=1}^N w_t^{(i)} f(\bar{x}_t^{(i)}) = \sum_{i=1}^N \frac{g_t^{y_t}(\bar{x}_t^{(i)}) f(\bar{x}_t^{(i)})}{\sum_{j=1}^N g_t^{y_t}(\bar{x}_t^{(j)})} \\ &= \frac{(fg_t^{y_t}, \xi_t^N)}{(g_t^{y_t}, \xi_t^N)} \end{aligned}$$

then it is apparent from the definition of G_t^N in (9) that

$$G_t^N(f, \bar{\pi}_t^N) = G_{t-1}^N(fg_t^{y_t}, \xi_t^N). \quad (41)$$

Taking together (40) and (41), we have

$$E[(f, \rho_t^N) | \bar{\mathcal{F}}_t] = G_{t-1}^N(fg_t^{y_t}, \xi_t^N). \quad (42)$$

Let \mathcal{F}_{t-1} be the σ -algebra generated by the set of variables $\{x_{0:t-1}^{(i)}, \bar{x}_{0:t-1}^{(i)} : 1 \leq i \leq N\}$. Since $\mathcal{F}_{t-1} \subseteq \bar{\mathcal{F}}_t$, Eq. (42) yields

$$\begin{aligned} E[(f, \rho_t^N) | \mathcal{F}_{t-1}] &= E[G_{t-1}^N(fg_t^{y_t}, \xi_t^N) | \mathcal{F}_{t-1}] \\ &= G_{t-1}^N E[(fg_t^{y_t}, \xi_t^N) | \mathcal{F}_{t-1}], \end{aligned} \quad (43)$$

since G_{t-1}^N is measurable w.r.t. \mathcal{F}_{t-1} . Moreover, for any $h \in B(\mathcal{X})$, it is straightforward to show that

$$E[(h, \xi_t^N) | \mathcal{F}_{t-1}] = (h, \tau_t \pi_{t-1}^N) = ((h, \tau_t), \pi_{t-1}^N),$$

hence, as $fg_t^{y_t} \in B(\mathcal{X})$, we readily obtain

$$E[(fg_t^{y_t}, \xi_t^N) | \mathcal{F}_{t-1}] = ((fg_t^{y_t}, \tau_t), \pi_{t-1}^N). \quad (44)$$

Substituting (44) into (43), we arrive at

$$\begin{aligned} E[(f, \rho_t^N) | \mathcal{F}_{t-1}] &= G_{t-1}^N(((fg_t^{y_t}, \tau_t), \pi_{t-1}^N)) \\ &= ((fg_t^{y_t}, \tau_t), \rho_{t-1}^N), \end{aligned} \quad (45)$$

where (45) follows from the definition of the estimate of ρ_{t-1} , namely $\rho_{t-1}^N = G_{t-1}^N \pi_{t-1}^N$. If we take unconditional expectations on both sides of Eq. (45), we obtain

$$\begin{aligned} E[(f, \rho_t^N)] &= E[(((fg_t^{y_t}, \tau_t), \rho_{t-1}^N))] \\ &= ((fg_t^{y_t}, \tau_t), \rho_{t-1}) \end{aligned} \quad (46)$$

$$= (f, g_t^{y_t} \cdot \tau_t \rho_{t-1}) \quad (47)$$

$$= (f, \rho_t), \quad (48)$$

where equality (46) follows from the induction hypothesis (39), (47) is obtained by simply re-ordering (46) and Eq. (48) follows from the recursive definition of ρ_t in (5).

Appendix 2 Proof of Lemma 3

For $t = 0$, $\rho_0^N = \pi_0^N$, hence the result follows from Lemma 1. At any time $t > 0$, since $\rho_t^N = G_t^N \pi_t^N$, we readily have

$$\begin{aligned} E[|(f, \rho_t^N) - (f, \rho_t)|^p] &= E\left[\left|\frac{1}{N} \sum_{i=1}^N G_t^N f(x_t^{(i)}) - (f, \rho_t)\right|^p\right] \\ &= E\left[\left|\frac{1}{N} \sum_{i=1}^N Z_t^{(i)}\right|^p\right], \end{aligned} \quad (49)$$

where $Z_t^{(i)} = G_t^N f(x_t^{(i)}) - (f, \rho_t)$, $i = 1, \dots, N$. It is apparent that the random variables $Z_t^{(i)}$, $i = 1, \dots, N$, are conditionally independent given the σ -algebra $\bar{\mathcal{F}}_t$ generated by the set $\{x_{0:t-1}^{(j)}, \bar{x}_{0:t-1}^{(j)} : 1 \leq j \leq N\}$. It can also be proved that every $Z_t^{(i)}$ is centred and bounded, as explicitly shown in the sequel.

To see that $Z_t^{(i)}$ has zero mean, let us note first that

$$E[G_t^N f(x_t^{(i)}) | \bar{\mathcal{F}}_t] = G_t^N(f, \bar{\pi}_t^N),$$

since G_t^N is measurable w.r.t. $\bar{\mathcal{F}}_t$. Moreover, by the same argument as in the proof of Lemma 2, one can show that $G_t^N(f, \bar{\pi}_t^N) = G_{t-1}^N(fg_t^{y_t}, \xi_t^N)$ and, therefore,

$$\begin{aligned} E[G_t^N f(x_t^{(i)}) | \mathcal{F}_{t-1}] &= E[G_{t-1}^N(fg_t^{y_t}, \xi_t^N) | \mathcal{F}_{t-1}] \\ &= G_{t-1}^N((fg_t^{y_t}, \tau_t), \pi_{t-1}^N), \end{aligned} \quad (50)$$

where we have used the fact that, for any $h \in B(\mathcal{X})$, $E[(h, \xi_t^N) | \mathcal{F}_{t-1}] = ((h, \tau_t), \pi_{t-1}^N)$. However, since $\rho_{t-1}^N = G_{t-1}^N \pi_{t-1}^N$, Eq. (50) amounts to

$$E[G_t^N f(x_t^{(i)}) | \mathcal{F}_{t-1}] = ((fg_t^{y_t}, \tau_t), \rho_{t-1}^N)$$

and taking (unconditional) expectations on both sides of the equation above yields

$$\begin{aligned} E[G_t^N f(x_t^{(i)})] &= E[(((fg_t^{y_t}, \tau_t), \rho_{t-1}^N))] \\ &= ((fg_t^{y_t}, \tau_t), \rho_{t-1}) \end{aligned} \quad (51)$$

$$= (f, \rho_t), \quad (52)$$

where (51) follows from Lemma 2 (i.e. ρ_{t-1}^N is unbiased) and (52) is a straightforward consequence of the definition of ρ_t in (5). Eq. 52 states that $E[Z_t^{(i)}] =$

$$E[G_t^N f(x_t^{(i)}) - (f, \rho_t)] = 0.$$

To see that (every) $Z_t^{(i)}$ is bounded, note that

$$G_t^N \leq \prod_{k=1}^t \|g_k^{y_k}\|_\infty < \infty, \quad \text{for any } t < \infty, \quad (53)$$

whereas

$$\begin{aligned} (f, \rho_t) &= ((fg_t^{y_t}, \tau_t), \rho_{t-1}) \\ &= (((fg_t^{y_t}, \tau_t) g_{t-1}^{y_{t-1}}, \tau_{t-1}) g_{t-2}^{y_{t-2}}, \dots, \tau_1), \pi_0) \\ &\leq \|f\|_\infty \prod_{k=1}^t \|g_k^{y_k}\|_\infty < \infty. \end{aligned} \quad (54)$$

Taking (53) and (54) together, we arrive at

$$|Z_t^{(i)}| \leq 2\|f\|_\infty \prod_{k=1}^t \|g_k^{y_k}\|_\infty \quad (55)$$

which is finite for any finite t (indeed, for every $t \leq T$).

Since the variables $Z_t^{(i)}$, $i = 1, \dots, N$, in (49) are bounded, with zero mean and conditionally independent given $\tilde{\mathcal{F}}_t$, it is not difficult to show (see, e.g. [23] (Lemma A.1)) that

$$E \left[|(f, \rho_t^N) - (f, \rho_t)|^p \right] \leq \frac{2^p \tilde{c}_t^p \|f\|_\infty^p \prod_{k=1}^t \|g_k^{y_k}\|_\infty^p}{N^{\frac{p}{2}}}, \quad (56)$$

where the constant \tilde{c}_t is finite and independent of N . From (56), we easily obtain the inequality (16) in the statement of Lemma 3, with $\tilde{c}_t = 2\tilde{c}_t \|f\|_\infty \prod_{k=1}^t \|g_k^{y_k}\|_\infty < \infty$ for any $t \leq T < \infty$.

Acknowledgements

The authors thank Dr. Katrin Achutegui for her valuable assistance in obtaining and plotting the numerical results in Section 4.

Funding

This work was partially supported by *Ministerio de Economía y Competitividad* of Spain (TEC2012-38883-C02-01 COMPREHENSION and TEC2015-69868-C2-1-R ADVENTURE) and the Office of Naval Research Global (N62909-15-1-2011). D. C. and J. M. acknowledge the support of the Isaac Newton Institute through the program *Monte Carlo Inference for High-Dimensional Statistical Models*.

Authors' contributions

DC and JM carried out the analysis and obtained the theoretical results. JM and GR-M coded the algorithms and run the computer experiments. All authors collaborated in the composition of the manuscript. The authors are listed in alphabetical order. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹Department of Mathematics of Imperial College London, London, UK. ²Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Madrid, Spain. ³Instituto de Investigación Sanitaria Gregorio Marañón (IiSGM), Madrid, Spain.

Received: 2 March 2017 Accepted: 1 May 2018

Published online: 25 May 2018

References

1. G Hendeby, R Karlsson, F Gustafsson, Particle filtering: the need for speed. *EURASIP J. Adv. Sig. Process.* **2010**, 22 (2010)
2. M Bolić, PM Djurić, S Hong, Resampling algorithms and architectures for distributed particle filters. *IEEE Trans. Sig. Process.* **53**(7), 2442–2450 (2005)
3. A Doucet, N de Freitas, N Gordon, *Sequential Monte Carlo Methods in Practice*. (Springer, New York, 2001)
4. P Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. (Springer-Verlag, New York, 2004)
5. O Cappé, SJ Godsill, E Moulines, An overview of existing methods and recent advances in sequential Monte Carlo. *Proc. IEEE.* **95**(5), 899–924 (2007)
6. A Bain, D Crisan, *Fundamentals of Stochastic Filtering*. (Springer-Verlag, New York, 2008)
7. A Gelencsér-Horváth, G Tornai, A Horváth, G Cserey, Fast, parallel implementation of particle filtering on the gpu architecture. *EURASIP J. Adv. Sig. Process.* **2013**(1), 1–16 (2013)
8. J Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization. *Digit. Sig. Process.* **17**, 787–807 (2007)
9. Hlinka O, Sluciak O, Hlawatsch F, Djurić P, Rupp M, Likelihood consensus and its application to distributed particle filtering. *IEEE Trans. Sig. Process.* **60**(8), 4334–4349 (2012)

10. J Miguez, MA Vázquez, A proof of uniform convergence over time for a distributed particle filter. *Sig. Process.* **122**, 152–163 (2016)
11. K Heine, N Whiteley, Fluctuations, stability and instability of a distributed particle filter with local exchange. *Stoch. Process. Appl.* **127.8**(2017), 2508–2541 (2016)
12. N Whiteley, A Lee, K Heine, On the role of interaction in sequential Monte Carlo algorithms. *Bernoulli.* **22**(1), 494–529 (2016)
13. C Vergé, C Dubarry, P Del Moral, E Moulines, On parallel implementation of sequential Monte Carlo methods: the island particle model. *Stat. Comput.* **25**(2), 243–260 (2015)
14. P Del Moral, E Moulines, J Olsson, C Vergé, Convergence properties of weighted particle islands with application to the double bootstrap algorithm. *Stoch. Syst.* **6**(2), 367–419 (2016)
15. W Han, *On the Numerical Solution of the Filtering Problem*. (Ph.D. Thesis. Department of Mathematics, Imperial College London, 2013)
16. N Gordon, D Salmond, AFM Smith, Novel approach to nonlinear and non-Gaussian Bayesian state estimation. *IEE Proc.-F.* **140**(2), 107–113 (1993)
17. A Doucet, N de Freitas, N Gordon, in *Sequential Monte Carlo Methods in Practice*, ed. by A Doucet, N de Freitas, and N Gordon. An introduction to sequential Monte Carlo methods (Springer-Verlag, New York, 2001), pp. 4–14. chapter 1
18. A Doucet, S Godsill, C Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput.* **10**(3), 197–208 (2000)
19. R Douc, O Cappé, in *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*. Comparison of resampling schemes for particle filtering (IEEE, 2005)
20. D Crisan, A Doucet, A survey of convergence results on particle filtering. *IEEE Trans. Sig. Process.* **50**(3), 736–746 (2002)
21. N Chopin, A sequential particle filter method for static models. *Biometrika.* **89**(3), 539–552 (2002)
22. XL Hu, TB Schon, L Ljung, A basic convergence result for particle filtering. *IEEE Trans. Sig. Process.* **56**(4), 1337–1348 (2008)
23. D Crisan, J Miguez, Particle-kernel estimation of the filter density in state-space models. *Bernoulli.* **20**(4), 1879–1929 (2014)
24. J Miguez, D Crisan, PM Djurić, On the convergence of two sequential Monte Carlo methods for maximum a posteriori sequence estimation and stochastic global optimization. *Stat. Comput.* **23**(1), 91–107 (2013)
25. P Del Moral, A Guionnet, On the stability of interacting processes with applications to filtering and genetic algorithms. *Ann. l'Institut Henri Poincaré, (B) Probab. Stat.* **37**(2), 155–194 (2001)
26. J Miguez, in *IEEE 8th Sensor Array and Multichannel Sig. Process. Workshop (SAM)*. On the uniform asymptotic convergence of a distributed particle filter (IEEE, 2014), pp. 241–244
27. C Andrieu, A Doucet, R Holenstein, Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. B.* **72**, 269–342 (2010)
28. N Chopin, PE Jacob, O Papaspiliopoulos, SMC2: an efficient algorithm for sequential analysis of state space models. *J. R. Stat. Soc. Ser. B Stat Methodol.* **75.3**(2013), 397–426 (2012)
29. J Miguez, IP Mariño, MA Vázquez, Analysis of a nonlinear importance sampling scheme for Bayesian parameter estimation in state-space models. *Sig. Process.* **142**, 281–291 (2018)
30. J Olsson, O Cappé, R Douc, E Moulines, Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli.* **14**(1), 155–179 (2008)
31. EN Lorenz, Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**(2), 130–141 (1963)
32. AJ Chorin, P Krause, Dimensional reduction for a Bayesian filter. *PNAS.* **101**(42), 15013–15017 (2004)
33. EN Lorenz, in *Proceedings of the Seminar on Predictability, vol. 1. Predictability: a problem partly solved* (European Centre on Medium Range Weather Forecasting, Reading, UK, 1996)
34. J Hakkarainen, A Ilin, A Solonen, M Laine, H Haario, J Tamminen, E Oja, H Järvinen, On closure parameter estimation in chaotic systems. *Nonlinear Proc. Geoph.* **19**(1), 127–143 (2012)