Probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems

Sara Pérez-Vieites[*]
Department of Signal Theory and Communications, Universidad Carlos III de Madrid,
Avenida de la Universidad 30, 28911 Leganés, Madrid, Spain

Inés P. Mariño[†]
Department of Biology and Geology, Physics and Inorganic Chemistry, Universidad Rey Juan Carlos,
Calle Tulipán s/n, 28933 Móstoles, Madrid, Spain;
Institute for Women's Health, University College London, London WC1E 6BT, United Kingdom;
and Research Laboratory Systemic Medicine of Healthy Ageing, Institute of Biology and Medicine,
National Research Lobachevsky State University of Nizhny Novgorod, 603950 Nizhny Novgorod, Russia

Joaquín Míguez[‡]
Department of Signal Theory and Communications, Universidad Carlos III de Madrid,
Avenida de la Universidad 30, 28911 Leganés, Madrid, Spain

(Received 8 June 2018; published 5 December 2018)

Many problems in physics demand the ability to calibrate the parameters and predict the time evolution of complex dynamical models using sequentially collected data. Here we introduce a general methodology for the joint estimation of the static parameters and the forecasting of the state variables of nonlinear stochastic dynamical models. The proposed scheme is essentially probabilistic. It aims at recursively computing the sequence of joint posterior probability distributions of the unknown model parameters and its (time-varying) state variables conditional on the available observations. This framework combines two layers of inference: In the first layer, a grid-based scheme is used to approximate the posterior probability distribution of the fixed parameters; in the second layer, filtering (or data assimilation) techniques are employed to track and predict different conditional probability distributions of the state variables. Various types of procedures (deterministic grids, Monte Carlo, Gaussian filters, etc.) can be plugged into both layers, leading to a wealth of algorithms. For this reason, we refer to the proposed methodology as nested hybrid filtering. In this paper we specifically explore the combination of Monte Carlo and quasi–Monte Carlo (deterministic) approximations in the first layer with Gaussian filtering methods in the second layer, but other approaches fit naturally within the framework. We prove a general convergence result for a class of procedures that use sequential Monte Carlo in the first layer. Then we turn to an illustrative numerical example. In particular, we apply and compare different implementations of the methodology to the tracking of the state, and the estimation of the fixed parameters, of a stochastic two-scale Lorenz 96 system. This model is commonly used to assess data assimilation procedures in meteorology. We show estimation and forecasting results, obtained with a desktop computer, for up to 5000 dynamic state variables.

DOI: 10.1103/PhysRevE.98.063305

I. INTRODUCTION

A common feature to many problems in some of the most active fields of science is the need to calibrate (i.e., estimate the parameters) and then forecast the time evolution of high-dimensional dynamical systems using sequentially collected observations. One can find obvious examples in meteorology, where current models for global weather forecasting involve the tracking of millions of time-varying state variables [1], as well as in oceanography [2] or in climate modeling [3]. This problem is not constrained to geophysics though. In biochemistry and ecology it is often necessary to forecast the evolution of populations of interacting species (typically animal and/or vegetal species in ecology and different types of reacting molecules in biochemistry), which usually involves the estimation of the parameters that govern the interaction as well [4].

A. State of the art

Traditionally, model calibration (i.e., the estimation of the model static parameters) and the tracking and forecasting of the time-varying state variables have been addressed separately. The problem of tracking the state of the system using sequentially collected observations is often termed data assimilation in geophysics, while it is referred to as stochastic or Bayesian filtering by researchers in computational

[*] spvieites@tsc.uc3m.es
[†] ines.perez@urjc.es
[‡] joaquin.miguez@uc3m.es

statistics and applied probability. Carrying out both tasks jointly, parameter estimation and state forecasting, is a hard problem posing several practical and theoretical difficulties.

Many procedures have been suggested over the years (see, e.g., [5–8], as well as [9] for a survey), however they are subject to problems related to observability (i.e., ambiguities), lack of performance guarantees, or prohibitive computational demands. Some of the most relevant techniques can be classified into one or more of the categories below.

(i) State augmentation methods with artificial dynamics. The state vector, which contains the dynamical variables that describe the physical system, is extended with any static unknown parameters (commonly reinterpreted as slowly changing dynamical variables) in the model [5,10–12]. Standard filtering (or data assimilation) techniques are then used in order to track and forecast the extended state vector.

(ii) Particle learning techniques. For some models, the posterior probability distribution of the static parameters, conditional on the system state, can be computed in closed form and it depends only on a set of finite-dimensional statistics [8,13,14]. In a Monte Carlo setting, e.g., for particle filters, this means that the static parameters can be efficiently represented by sampling. Unfortunately, this approach is restricted to very specific models (an attempt to extend this idea to a more general setting can be found in [15]). The term particle learning was coined in [8], although the fundamental ideas were introduced earlier [13,14].

(iii) Classical importance resampling methods. Several authors have studied the performance of classical sequential importance sampling for static parameters [16–18]. Unfortunately, such algorithms tend to degenerate quickly over time unless certain conditions are met by the prior and posterior distributions [16,17] or computationally heavy interpolation schemes are adopted for the static parameters [18].

Only in the past few years have there been advances leading to well-principled probabilistic methods that solve the joint problem numerically and are supported by rigorous performance analyses [19–23]. They aim at calculating the posterior probability distribution of all the unknown variables and parameters of the model. From the viewpoint of Bayesian analysis, these conditional, or posterior, distributions contain all the information relevant for the estimation task. From them, one can compute point estimates of the parameters and states but also quantify the estimation error. However, state-of-the-art methods for Bayesian parameter estimation and stochastic filtering are batch techniques, i.e., they process the whole set of available observations repeatedly in order to produce numerical solutions. For this reason, they are not well suited to problems where observations are collected sequentially and have to be processed as they arrive (or simply when the sequence of observations is too long). The popular particle Markov chain Monte Carlo (PMCMC) [19] and the sequential Monte Carlo square (SMCS) [20] schemes are examples of such batch methods. The nested particle filter (NPF) of [21] is a purely recursive Monte Carlo method, more suitable than PMCMC and SMCS when long sequences of observations have to be processed. However, this technique is still computationally prohibitive in high-dimensional settings as it relies on two layers of intertwined Monte Carlo approximations.

While the schemes in [19–21] fully rely on Monte Carlo approximations in order to approximate the posterior probability distribution of the parameters and the states, there is an alternative class of schemes, often coined recursive maximum likelihood methods [9,10,24,25], that enable the sequential processing of the observed data as they are collected but do not yield full posterior distributions of the unknowns. They only output point estimates instead. Therefore, it is not possible to quantify the uncertainty of the estimates or forecasts. Moreover, they are subject to various convergence (and complexity) issues, e.g., when the posterior probability distribution is multimodal, when it contains singularities, or when the parameter likelihoods cannot be computed exactly.

In the physics literature, approximation schemes have been proposed that exploit the conditional dependences between the static parameters and the dynamic state variables in a way that resembles the SMCS or NPF schemes. The authors of [26] introduce a two-stage filter that alternates the estimation of static parameters (conditional on a fixed state estimate) and the tracking of the dynamic variables (conditional on a fixed estimate of the static parameters). Another alternating scheme, which combines Monte Carlo estimators with ensemble Kalman filters in order to handle the static parameters and dynamic states, can be found in [27].

In [28], an expectation-maximization algorithm is used to track a particle whose dynamics is governed by a hidden Markov model. The expectation step involves a (Monte Carlo–based) forward-filtering, backward-smoothing step that is computationally heavy and prevents the online application of the method. The authors of [29] investigate a variational scheme (based on the Laplace integral approximation) for data assimilation (including state and parameter estimation) and illustrate it with applications to the Lorenz 63 and Lorenz 96 models in a low dimensional setting. The same task of data assimilation with parameter estimation is tackled in [30]. In this case, the estimation of the states and parameter is reduced to an optimization problem that can be solved via an adjoint method for the estimation of a Hessian matrix. The schemes in [28–30] require one to process the data in batches, rather than recursively, and hence they are not well suited for online implementations. A sequential method, based on variational Bayes techniques, that admits an online (recursive) implementation can be found in [31]. However, the latter contribution is limited to the estimation of the time-varying states and does not deal with unknown static parameters.

B. Contribution

In this paper we propose a general probabilistic scheme to perform the joint task of parameter estimation and state tracking and forecasting. The methodology is Bayesian, i.e., it aims at the computation of the posterior probability distribution of the unknowns given the available data. It involves two layers of estimators, one for the static parameters and another one for the time-varying state variables. It can be interpreted that the state estimators and predictors are nested, or inserted, within a main algorithm that tackles the estimation of the parameters. The estimation of the static parameters and the dynamic variables is carried out in a purely sequential and recursive manner. This property makes the proposed method well suited for problems where long time series of data have to be handled.

It can be shown that a particular case of the proposed scheme is the NPF of [21], which relies on a sequential Monte Carlo sampler in the parameter space and bank of particle filters [32,33] in the space of the dynamic variables. However, the key feature and advantage of the general scheme that we advocate here is the ability to combine different types of algorithms in the two layers of inference (parameters and dynamic variables). Any grid-based method (where the probability distribution of the static parameters is represented by a set of points in the parameter space) can be employed in the first layer, while the computationally heavy particle filters in the second layer of the NPF can be replaced by simpler algorithms, easier to apply in practical problems.

We have investigated the use of sequential Monte Carlo and quasi–Monte Carlo [34] techniques in the parameter estimation layer. We note that the quasi–Monte Carlo scheme is a deterministic technique, although it formally resembles the Monte Carlo approach (hence the name). In the same vein, an unscented Kalman filter can be utilized in the parameter estimation layer, although we have left this for future research. For the second layer, we have assessed two Gaussian filters, namely, the extended Kalman filter (EKF) and the ensemble Kalman filter (EnKF). These two types of Gaussian filters have been well studied in the geophysics literature and there are a number of numerical techniques to ease their practical implementation for large-scale systems (e.g., covariance inflation [35,36] or localization [37–39]).

Because the flexibility to combine estimation techniques of different types within the same overall scheme is a key advantage of the proposed methodology, we refer to the resulting algorithms in general as nested hybrid filters (NHFs). Besides the numerical example described below, we provide a theoretical result on the asymptotic convergence of NHFs that use a sequential Monte Carlo scheme in the first layer (for the static parameters) and finite-variance estimators of the state variables in the second layer. Our analysis shows that the NHF can be biased if the filters in the second layer are (as it is the case in general with approximate Gaussian filters). However, it also ensures that the approximate posterior distribution of the parameters generated by the NHF, consisting of N samples in the parameter space, converges to a well-defined limit distribution with rate $O(N^{-1/2})$ under mild assumptions.

To illustrate the performance of the methodology, we present the results of computer simulations with a stochastic two-scale Lorenz 96 model [40] with underlying chaotic dynamics. In meteorology, the Lorenz 96 model is commonly used as a benchmark system for data assimilation [41,42] and parameter estimation techniques [43,44] because it displays the basic physical features of atmospheric dynamics [40] (e.g., convection and sensitivity to perturbations) and its dimension (number of state variables) can be selected arbitrarily, so it is possible to make the system as high dimensional as one desires. We have implemented, and compared numerically, four NHFs that combine Monte Carlo, quasi–Monte Carlo, EKF, and EnKF schemes in different ways. All the combinations that we have tried yield significant reductions of running times in comparison with the NPF for this model, without a significant loss of accuracy. We report simulation results for systems with up to 5000 dynamical variables to track and forecast.

C. Organization of the paper

The rest of the paper is organized as follows. After a brief comment on notation, we describe in Sec. II the class of (stochastic) dynamical systems of interest. Nested hybrid filters are introduced and explained in Sec. III. The asymptotic convergence theorem is stated and discussed in Sec. IV. In Sec. V the stochastic Lorenz 96 model, which is used in the simulations, is described. Some illustrative numerical results are presented in Sec. VI. Section VII provides a summary.

D. Notation

We denote vectors and matrices by boldface letters, either lowercase (for vectors) or uppercase (for matrices). Scalar magnitudes are denoted by lightface letters. For example, $d \in \mathbb{N}$ and $x \in \mathbb{R}$ are scalars, $\boldsymbol{x} \in \mathbb{R}^d$ is a vector, and $\boldsymbol{X} \in \mathbb{R}^{d\times d}$ is a matrix.

Most of the magnitudes of interest in this paper are random vectors (RVs). If $\boldsymbol{x}$ is a d-dimensional random variable taking values in $\mathbb{R}^d$, we use the generic notation $p(\boldsymbol{x})$ for its probability density function (PDF). This is an argumentwise notation. If we have two RVs $\boldsymbol{x}$ and $\boldsymbol{y}$, we write $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ for their respective PDFs, which are possibly different. In a similar vein, $p(\boldsymbol{x}, \boldsymbol{y})$ denotes the joint PDF of the two RVs and $p(\boldsymbol{x}|\boldsymbol{y})$ denotes the conditional PDF of $\boldsymbol{x}$ given $\boldsymbol{y}$. We find this simple notation convenient for the presentation of the model and methods and introduce a more specific terminology only for the analysis of convergence. We assume, for simplicity, that all random magnitudes can be described by PDFs with respect to the Lebesgue measure. The notation $\boldsymbol{x} \sim p(\boldsymbol{x})$ is read as the random variable $\boldsymbol{x}$ is distributed according to the PDF $p(\boldsymbol{x})$.

II. DYNAMICAL MODEL AND PROBLEM STATEMENT

A. State space models

We are interested in systems that can be described by a multidimensional stochastic differential equation (SDE) of the form

$$d\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})dt + \sigma d\boldsymbol{w}, \tag{1}$$

where t denotes continuous time, $\boldsymbol{x}(t) \in \mathbb{R}^{d_x}$ is the d_x-dimensional system state, $\boldsymbol{f} : \mathbb{R}^{d_x} \times \mathbb{R}^{d_\theta} \to \mathbb{R}^{d_x}$ is a nonlinear function parametrized by a fixed vector of unknown parameters, $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$, $\sigma > 0$ is a scale parameter that controls the intensity of the stochastic perturbation, and $\boldsymbol{w}(t)$ is a $d_x \times 1$ vector of independent standard Wiener processes. Very often, the underlying ordinary differential equation $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ describes some peculiar dynamics inherent to the system of interest (e.g., many of the systems of interest in geophysics are chaotic) and the addition of the perturbation $\boldsymbol{w}(t)$ accounts for model errors or other sources of uncertainty.

Equation (1) does not have a closed-form solution for a general nonlinear function $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ and therefore it has to be discretized for its numerical integration. A discretization scheme with fixed step size $h > 0$ yields in general a discrete-time stochastic dynamical system of the form

$$\bar{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_{k-1} + \boldsymbol{F}(\bar{\boldsymbol{x}}_{k-1}, \boldsymbol{\theta}, h, \sigma \boldsymbol{w}_k), \tag{2}$$

where $k \in \mathbb{N}$ denotes discrete time, $\bar{\boldsymbol{x}}_k \simeq \boldsymbol{x}(kh)$ is the system state at time $t = kh$, and $\boldsymbol{w}_k$ is an random variable of dimension $d_w \geqslant d_x$, which results from the integration of d_w independent Wiener processes. Since the integral of a Wiener process over an interval of length h is a Gaussian random variable with zero mean and variance h, the random variable $\boldsymbol{w}_k$ is also Gaussian, with mean $\boldsymbol{0}$ and covariance matrix $h\boldsymbol{I}_{d_w}$, where $\boldsymbol{I}_{d_w}$ is the $d_w \times d_w$ identity matrix. This was defined as $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{w}_k|\boldsymbol{0}, h\boldsymbol{I}_{d_w})$. The function $\boldsymbol{F}$ depends on the choice of discretization scheme. The simplest one is the Euler-Maruyama method, which yields [45]

$$\bar{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_{k-1} + h\boldsymbol{f}(\bar{\boldsymbol{x}}_{k-1}, \boldsymbol{\theta}) + \sigma \boldsymbol{w}_k, \tag{3}$$

i.e., the noise is additive, with $d_w = d_x$, and $\boldsymbol{F}(\bar{\boldsymbol{x}}_{k-1}, \boldsymbol{\theta}, h, \sigma \boldsymbol{w}_k) = h\boldsymbol{f}(\bar{\boldsymbol{x}}_{k-1}, \boldsymbol{\theta}) + \sigma \boldsymbol{w}_k$. For a Runge-Kutta method of order q, as a more sophisticated example, $d_w = qd_x$ and the function $\boldsymbol{F}$ results from applying $\boldsymbol{f}$, q times, with a Gaussian perturbation passing through the nonlinearity at each of these intermediate steps. See [45] for details on various integration methods for SDEs. In the following we work with the general equation (2).

We assume that the system of Eq. (2) can be observed every m discrete-time steps (i.e., every mh continuous-time units). The nth observation is an random variable $\boldsymbol{y}_n \in \mathbb{R}^{d_y}$ of dimension d_y, which we model as

$$\boldsymbol{y}_n = \boldsymbol{g}(\boldsymbol{x}_n, \boldsymbol{\theta}) + \boldsymbol{r}_n, \tag{4}$$

$n \in \mathbb{N}$, where $\boldsymbol{x}_n = \bar{\boldsymbol{x}}_{nm} \simeq \boldsymbol{x}(nmh)$ is the system state at the time of the nth observation (continuous time $t = nmh$), $\boldsymbol{g} : \mathbb{R}^{d_x} \times \mathbb{R}^{d_\theta} \to \mathbb{R}^{d_y}$ is a transformation that maps the state into the observation space, and $\boldsymbol{r}_n$ is a $\boldsymbol{0}$-mean observational-noise vector with covariance matrix $\sigma_0^2 \boldsymbol{I}_{d_y}$.

We can rewrite the state (2) in the timescale of the observations (i.e., discrete-time n rather than k) as

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \boldsymbol{F}^m(\boldsymbol{x}_{n-1}, \boldsymbol{\theta}, h, \sigma \boldsymbol{w}_n), \tag{5}$$

$n \in \mathbb{N}$, where the notation $\boldsymbol{F}^m$ indicates that Eq. (2) is applied m consecutive times in order to move from $\boldsymbol{x}_{n-1} = \bar{\boldsymbol{x}}_{(n-1)m}$ to $\boldsymbol{x}_n = \bar{\boldsymbol{x}}_{nm}$. Note that, as a consequence, the noise vector $\boldsymbol{w}_n$ in Eq. (5) has dimension $d_w = mqd_x$.

B. Probabilistic representation and problem statement

The state equation (5) together with the observation equation (4) describes a (stochastic) state space model. The goal is to design methods for the recursive estimation of both the static parameters $\boldsymbol{\theta}$ and the states $\boldsymbol{x}_n$, $n \in \mathbb{N}$. Note that the latter implies the estimation of the sequence $\bar{\boldsymbol{x}}_k \simeq \boldsymbol{x}(kh)$, $k \in \mathbb{N}$, i.e., the states between observation instants have to be estimated as well.

We adopt a Bayesian approach to this task. From this point of view, both the parameters $\boldsymbol{\theta}$ and the sequence of states $\boldsymbol{x}_n$ are random and we aim at computing their respective probability distributions conditional on the available data, i.e., the sequence of observations $\boldsymbol{y}_n$. The problem is best described if we replace the functional representation of the state space model in Eqs. (5) and (4) by an equivalent one in terms of PDFs.[1] To be specific, the probabilistic representation consists of the elements

$$\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0), \tag{6}$$

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}), \tag{7}$$

$$\boldsymbol{x}_n \sim p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}, \boldsymbol{\theta}), \tag{8}$$

$$\boldsymbol{y}_n \sim p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta}), \tag{9}$$

where $p(\boldsymbol{x}_0)$ and $p(\boldsymbol{\theta})$ are, respectively, the *a priori* PDFs of the system state and the parameter vector at time $n = 0$ ($t = 0$ as well), $p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}, \boldsymbol{\theta})$ is the conditional PDF of $\boldsymbol{x}_n$ given the state $\boldsymbol{x}_{n-1}$ and the parameters in $\boldsymbol{\theta}$, and $p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta})$ is the conditional PDF of the observation given the state and the parameters.

We note that the following.

(i) The priors $p(\boldsymbol{x}_0)$ and $p(\boldsymbol{\theta})$ can be understood as a probabilistic characterization of uncertainty regarding the system initial condition. If the initial condition $\boldsymbol{x}_0$ is known exactly, $p(\boldsymbol{x}_0)$ could be replaced by a Dirac δ allocating probability 1 at that point.

(ii) The PDF $p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}, \boldsymbol{\theta})$ does not have in general a closed-form expression because of the nonlinearity $\boldsymbol{F}^m$. However, it is usually straightforward to simulate $\boldsymbol{x}_n$ given $\boldsymbol{x}_{n-1}$ and $\boldsymbol{\theta}$ using Eq. (5) and this is sufficient for many methods to work.

(iii) The observations are conditionally independent given the states and the parameters. If the observational noise $\boldsymbol{v}_n$ is Gaussian, then $p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{g}(\boldsymbol{x}_n, \boldsymbol{\theta}), \sigma_0^2 \boldsymbol{I}_{d_y})$.

From a Bayesian perspective, all the information relevant for the characterization of $\boldsymbol{\theta}$ and $\boldsymbol{x}_n$ at discrete time n (corresponding to $t = nmh$) is contained in the joint posterior PDF $p(\boldsymbol{\theta}, \boldsymbol{x}_n|\boldsymbol{y}_{1,\ldots,n})$, where $\boldsymbol{y}_{1,\ldots,n} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n\}$. The latter density cannot be computed exactly in general and the goal of this paper is to describe a class of flexible and efficient recursive methods for its approximation.

We will show that one way to attain this goal is to tackle the approximation of the sequence of posterior PDFs of the parameters $p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n})$, $n \in \mathbb{N}$. This yields, in a natural way, approximations for $p(\boldsymbol{\theta}, \boldsymbol{x}_n|\boldsymbol{y}_{1,\ldots,n})$ and $p(\boldsymbol{x}_n|\boldsymbol{y}_{1,\ldots,n})$ for each n, as well as predictions for the densities of the intermediate states $p(\bar{\boldsymbol{x}}_{nm+k}|\boldsymbol{y}_{1,\ldots,n})$ for $k = 1, \ldots, m-1$.

III. NESTED HYBRID FILTERING

A. Importance sampling for parameter estimation

In order to introduce the proposed scheme of nested hybrid filters, let us consider the approximation of the nth posterior probability distribution of the parameters, with the

[1] The PDFs in the model (6)–(9) always exist if the functions $\boldsymbol{f}(\cdot, \cdot)$ and $\boldsymbol{g}(\cdot, \cdot)$ are differentiable. There are problems for which the conditional probability distribution of $\boldsymbol{x}_n$ conditional on $\boldsymbol{x}_{n-1}$, which we denote by τ_n, may not have a density with respect to the Lebesgue measure and the model (6)–(9) would not be well defined the way it is written. Even in that case, however, the proposed methodology could be applied using approximation filters in the second layer which do not depend on the differentiability of $\boldsymbol{f}$, such as particle filters or EnKFs.

PDF $p(\boldsymbol{\theta}_n|\boldsymbol{y}_{1,\ldots,n})$, using classical importance sampling [46]. In particular, let $q_n(\boldsymbol{\theta})$ be an arbitrary proposal PDF for the parameter vector $\boldsymbol{\theta}$ and assume that $q_n(\boldsymbol{\theta}) > 0$ whenever $p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n}) > 0$.

Assume that the posterior at time $n-1$, $p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n-1})$, is available. Then the posterior PDF at time n can be expressed, via Bayes' theorem, as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n}) \propto p(\boldsymbol{y}_n|\boldsymbol{\theta},\boldsymbol{y}_{1,\ldots,n-1})p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n-1}), \tag{10}$$

where the proportionality constant $p(\boldsymbol{y}_n|\boldsymbol{y}_{1,\ldots,n-1})$ is independent of $\boldsymbol{\theta}$. Expression (10) enables the application of the importance sampling method to approximate integrals with respect to the posterior PDF $p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n})$ (i.e., to approximate the statistics of this probability distribution). Specifically, if we draw N independent and identically distributed samples from $q_n(\boldsymbol{\theta})$, denoted by $\boldsymbol{\theta}_n^i$, $i=1,\ldots,N$, compute importance weights of the form

$$\tilde{w}_n^i = \frac{p\left(\boldsymbol{y}_n|\boldsymbol{\theta}_n^i,\boldsymbol{y}_{1,\ldots,n-1}\right)p\left(\boldsymbol{\theta}_n^i|\boldsymbol{y}_{1,\ldots,n-1}\right)}{q_n\left(\boldsymbol{\theta}_n^i\right)}$$

and normalize them to obtain

$$w_n^i = \frac{\tilde{w}_n^i}{\sum_{k=1}^N \tilde{w}_n^k}, \quad i=1,\ldots,N,$$

then it can be proved [46] that

$$\lim_{N\to\infty}\sum_{i=1}^N w_n^i f(\boldsymbol{\theta}_n^i) = \int f(\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n})d\boldsymbol{\theta} \quad \text{a.s.} \tag{11}$$

(where a.s. denotes almost surely) for any integrable function $f:\mathbb{R}^{d_\theta}\to\mathbb{R}$ under mild regularity assumptions. In this way one could estimate the value of $\boldsymbol{\theta}$, e.g.,

$$\boldsymbol{\theta}_n^N = \sum_{i=1}^N w_n^i\boldsymbol{\theta}_n^i \simeq \int \boldsymbol{\theta}p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n})d\boldsymbol{\theta} =: \mathbb{E}[\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n}],$$

where $\mathbb{E}[\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n}]$ denotes the expected value of $\boldsymbol{\theta}$ conditional on the observations $\boldsymbol{y}_{1,\ldots,n}$. We could also estimate the mean square error (MSE) of this estimator, as

$$\text{MSE}_n^N = \sum_{i=1}^N w_n^i\left\|\boldsymbol{\theta}_n^i - \hat{\boldsymbol{\theta}}_n\right\|^2 \simeq \int \left\|\boldsymbol{\theta}_n^i - \mathbb{E}[\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n}]\right\|^2 \times p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n})d\boldsymbol{\theta}. \tag{12}$$

The choice of $q_n(\boldsymbol{\theta})$ is of course key to the complexity and the performance of importance sampling schemes. One particularly simple choice is $q_n(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n-1})$, which reduces the importance sampling algorithm to (1) drawing N independent and identically distributed samples $\boldsymbol{\theta}_n^i$, $i=1,\ldots,N$, from $p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n-1})$, and (2) computing normalized importance weights $w_n^i \propto p(\boldsymbol{y}_n|\boldsymbol{\theta}_n^i,\boldsymbol{y}_{1,\ldots,n-1})$, $i=1,\ldots,N$. Unfortunately, this method is not practical because (i) it is not possible to draw exactly from $p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n})$, since this PDF is unknown, and (ii) the likelihood function $p(\boldsymbol{y}_n|\boldsymbol{\theta}_n^i,\boldsymbol{y}_{1,\ldots,n-1})$ cannot be evaluated exactly either. In the following we tackle the two issues above and in doing so obtain a general scheme for the approximation of the posterior distribution of the parameter vector $\boldsymbol{\theta}$ and the state vector $\boldsymbol{x}_n$, i.e., the distribution with PDF $p(\boldsymbol{\theta},\boldsymbol{x}_n|\boldsymbol{y}_{1,\ldots,n})$.

B. Sequential Monte Carlo hybrid filter

It is well known that the likelihood $u_n(\boldsymbol{\theta}) := p(\boldsymbol{y}_n|\boldsymbol{\theta},\boldsymbol{y}_{1,\ldots,n-1})$ can be approximated using filtering algorithms [19,47]. To be specific, the function $u_n(\boldsymbol{\theta})$ can be written as the integral

$$u_n(\boldsymbol{\theta}) = \int p(\boldsymbol{y}_n|\boldsymbol{x}_n,\boldsymbol{\theta})p(\boldsymbol{x}_n|\boldsymbol{\theta},\boldsymbol{y}_{1,\ldots,n-1})d\boldsymbol{\theta}, \tag{13}$$

where in turn the predictive density $p(\boldsymbol{x}_n|\boldsymbol{\theta},\boldsymbol{y}_{1,\ldots,n-1})$ is

$$p(\boldsymbol{x}_n|\boldsymbol{\theta},\boldsymbol{y}_{1,\ldots,n-1}) = \int p(\boldsymbol{x}_n|\boldsymbol{\theta},\boldsymbol{x}_{n-1})p(\boldsymbol{x}_{n-1}|\boldsymbol{\theta},\boldsymbol{y}_{1,\ldots,n-1})d\boldsymbol{x}_{n-1} \tag{14}$$

and

$$p(\boldsymbol{x}_{n-1}|\boldsymbol{\theta},\boldsymbol{y}_{1,\ldots,n-1}) \propto p(\boldsymbol{y}_{n-1}|\boldsymbol{\theta},\boldsymbol{x}_{n-1})p(\boldsymbol{x}_{n-1}|\boldsymbol{\theta},\boldsymbol{y}_{1,\ldots,n-2}). \tag{15}$$

Given a fixed parameter vector $\boldsymbol{\theta}$ and a prior PDF $p(\boldsymbol{x}_0|\boldsymbol{\theta})$, the sequence of likelihoods $u_n(\boldsymbol{\theta})$ can be computed by recursively applying Eqs. (13)–(15) for $n=1,2,\ldots$.

Let us now assume that we are given a sequence of parameter vectors $\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_{k-1},\boldsymbol{\theta}_k$ and we are interested in computing the likelihood of the last vector, $\boldsymbol{\theta}_k = \boldsymbol{\theta}'$. Following [21], one can compute a sequence of approximate likelihoods $\hat{u}_n(\boldsymbol{\theta}_n)$, $n=1,\ldots,k$, using the recursion

$$\hat{p}(\boldsymbol{x}_{n-1}|\boldsymbol{\theta}_{n-1},\boldsymbol{y}_{1,\ldots,n-1}) \propto p(\boldsymbol{y}_{n-1}|\boldsymbol{\theta}_{n-1},\boldsymbol{x}_{n-1})\hat{p}(\boldsymbol{x}_{n-1}|\boldsymbol{\theta}_{n-1},\boldsymbol{y}_{1,\ldots,n-2}), \tag{16}$$

$$\hat{p}(\boldsymbol{x}_n|\boldsymbol{\theta}_n,\boldsymbol{y}_{1,\ldots,n-1}) := \int p(\boldsymbol{x}_n|\boldsymbol{\theta}_n,\boldsymbol{x}_{n-1})\hat{p}(\boldsymbol{x}_{n-1}|\boldsymbol{\theta}_{n-1},\boldsymbol{y}_{1,\ldots,n-1})d\boldsymbol{x}_{n-1}, \tag{17}$$

$$\hat{u}_n(\boldsymbol{\theta}_n) := \int p(\boldsymbol{y}_n|\boldsymbol{\theta}_n,\boldsymbol{x}_n)\hat{p}(\boldsymbol{x}_n|\boldsymbol{\theta}_n,\boldsymbol{y}_{1,\ldots,n-1})d\boldsymbol{x}_n, \tag{18}$$

which starts with the initial density $\hat{p}(\boldsymbol{x}_0|\boldsymbol{\theta}_0) := p(\boldsymbol{x}_0|\boldsymbol{\theta}_0)$. It can be proved, using the same type of continuity arguments as in [21], that the approximation error

$$|u_k(\boldsymbol{\theta}') - \hat{u}_k(\boldsymbol{\theta}')| \tag{19}$$

can be kept bounded, for any k, provided some simple assumptions on the state space model and the sequence $\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n$ are satisfied. Note that, in expression (19), $u_k(\boldsymbol{\theta}')$ is the actual likelihood calculated by iterating (13)–(15) for $n=1,\ldots,k$, while $\hat{u}_k(\boldsymbol{\theta}')$ is the approximation computed using the sequence $\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_{k-1},\boldsymbol{\theta}_k=\boldsymbol{\theta}'$ and recursion (16)–(18).

The recursive approximation scheme for $\hat{u}_n(\boldsymbol{\theta})$ can be combined with the "naive" importance sampling procedure of Sec. III A to yield a general (and practical) method for the approximation of the sequence of *a posteriori* probability distributions of the parameter vector $\boldsymbol{\theta}$, defined as

$$\mu_n(d\boldsymbol{\theta}) := p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n})d\boldsymbol{\theta}.$$

We refer to the proposed scheme as a nested hybrid filter and provide a detailed outline in Algorithm 1.

```
Algorithm 1: Nested hybrid filter.
Inputs:
1. $N$ (number of Monte Carlo samples to generate).
2. A priori PDFs $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x}_0)$.
3. A Markov kernel $\kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}')$ which, given $\boldsymbol{\theta}' \in D$, generates jittered parameters $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$.
Procedure:
1. Initialization. Draw $\boldsymbol{\theta}_0^i$, $i = 1, \ldots, N$, independent and identically distributed samples from $\mu_0(d\boldsymbol{\theta}) = p(\boldsymbol{\theta})d\boldsymbol{\theta}$.
2. Recursive step.
  (a) For $i = 1, \ldots, N$ perform the following.
    (i) Draw $\bar{\boldsymbol{\theta}}_n^i$ from $\kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}_{n-1}^i)$.
    (ii) Approximate $\hat{p}(\boldsymbol{x}_n|\bar{\boldsymbol{\theta}}_n^i, \boldsymbol{y}_{1,\ldots,n-1})$ using a filtering algorithm.
    (iii) Use this approximation to compute the estimate
        $\hat{u}_n(\bar{\boldsymbol{\theta}}_n^i) = \int p(\boldsymbol{y}_n|\bar{\boldsymbol{\theta}}_n^i, \boldsymbol{x}_n)\hat{p}(\boldsymbol{x}_n|\bar{\boldsymbol{\theta}}_n^i, \boldsymbol{y}_{1,\ldots,n-1})d\boldsymbol{x}_n \tag{20}$
    and let $w_n^i \propto \hat{u}_n(\bar{\boldsymbol{\theta}}_n^i)$ be the normalized weight of $\bar{\boldsymbol{\theta}}_n^i$.
  (b) Resample the discrete distribution
        $\tilde{\mu}_n^N(d\boldsymbol{\theta}) = \sum_{i=1}^{N} w_n^i \delta_{\bar{\boldsymbol{\theta}}_n^i}(d\boldsymbol{\theta}) \tag{21}$
    $N$ times with replacement in order to obtain the particle set $\{\boldsymbol{\theta}_n^i\}_{i=1}^N$ and the approximate probability measure $\mu_n^N(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_n^i}(d\boldsymbol{\theta})$.
Outputs: A set of particles $\{\boldsymbol{\theta}_n^i\}_{i=1}^N$ and a probability measure $\mu_n^N(d\boldsymbol{\theta})$.
```

Algorithm 1 is essentially a sequential Monte Carlo (SMC) method, often known as a particle filter [32,48,49]. At each time step n, the output of the algorithm is an estimate of the posterior probability distribution $\mu_n(d\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n})d\boldsymbol{\theta}$. Specifically, we construct the discrete and random probability measure

$$\mu_n^N(d\boldsymbol{\theta}) = \frac{1}{N}\sum \delta_{\boldsymbol{\theta}_n^i}(d\boldsymbol{\theta}), \tag{22}$$

which can be used to approximate any integrals with respect to the true probability measure $\mu_n(d\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n})d\boldsymbol{\theta}$. For example, one can estimate any posterior expectations of the parameter vector $\boldsymbol{\theta}$ given the observations $\boldsymbol{y}_{1,\ldots,n}$, namely,

$$\begin{aligned}\mathbb{E}[\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n}] &= \int \boldsymbol{\theta}\mu_n(d\boldsymbol{\theta}) \simeq \int \boldsymbol{\theta}\mu_n^N(d\boldsymbol{\theta}) \\ &= \frac{1}{N}\sum_i \boldsymbol{\theta}_n^i =: \boldsymbol{\theta}_n^N.\end{aligned} \tag{23}$$

Since we have constructed a complete distribution, statistical errors can be estimated as well. The *a posteriori* covariance matrix of vector $\boldsymbol{\theta}$ can be approximated as

$$\begin{aligned}&\mathbb{E}[(\boldsymbol{\theta} - \mathbb{E}[\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n}])(\boldsymbol{\theta} - \mathbb{E}[\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n}])^\top|\boldsymbol{y}_{1,\ldots,n}] \\ &\simeq \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{\theta}_n^i - \boldsymbol{\theta}_n^N\right)\left(\boldsymbol{\theta}_n^i - \boldsymbol{\theta}_n^N\right)^\top =: \boldsymbol{P}_n^N.\end{aligned} \tag{24}$$

As a by-product, Algorithm 1 also yields an approximate predictive PDF for $\boldsymbol{x}_n$, namely,

$$\hat{p}(\boldsymbol{x}_n|\boldsymbol{y}_{1,\ldots,n-1}) = \sum_{i=1}^{N} w_n^i \hat{p}\left(\boldsymbol{x}_n|\boldsymbol{\theta}_n^i, \boldsymbol{y}_{1,\ldots,n-1}\right).$$

If one computes the approximate filter $\hat{p}(\boldsymbol{x}_n|\boldsymbol{\theta}_n^i, \boldsymbol{y}_{1,\ldots,n-1})$ as well, then the joint probability distribution of $\boldsymbol{\theta}$ and $\boldsymbol{x}_n$ conditioned on $\boldsymbol{y}_{1,\ldots,n}$ [denoted by $\pi_n(d\boldsymbol{\theta} \times d\boldsymbol{x}_n)$] can be approximated as

$$\pi_n^N(d\boldsymbol{\theta} \times d\boldsymbol{x}_n) = \sum_{i=1}^{N} w_n^i \hat{p}(\boldsymbol{x}_n|\boldsymbol{\theta}, \boldsymbol{y}_{1,\ldots,n})\delta_{\bar{\boldsymbol{\theta}}_n^i}(d\boldsymbol{\theta})d\boldsymbol{x}_n.$$

The scheme of Algorithm 1 is referred to as nested because the SMC algorithm generates, at each time step n, a set of samples $\{\boldsymbol{\theta}_n^1, \ldots, \boldsymbol{\theta}_n^N\}$ and for each sample $\boldsymbol{\theta}_n^i$ we embed a filter in the state space $\mathbb{R}^{d_x}$ in order to compute the PDF $\hat{p}(\boldsymbol{x}_n|\bar{\boldsymbol{\theta}}_n^i, \boldsymbol{y}_{1,\ldots,n-1})$ and the approximate likelihood $\hat{u}_n(\boldsymbol{\theta}_n^i)$. The term *hybrid* is used because the embedded filters need not be Monte Carlo methods; a variety of techniques can be used and in this paper we focus on Gaussian filters, which are attractive because of their (relative) computational simplicity. A scheme with nested particle filters was thoroughly studied in [21,50].

Let us finally remark that the NHF scheme relies on two approximations.

(i) Jittering of the parameters. The difficulty of drawing samples from $\mu_{n-1}(d\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{y}_{1,\ldots,n-1})d\boldsymbol{\theta}$ can be circumvented if we content ourselves with an approximate sampling step. In particular, if we have computed a Monte Carlo approximation $\mu_{n-1}^N(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_{n-1}^i}(d\boldsymbol{\theta})$ at time $n-1$ (with some of the samples replicated because of the resampling step) then we can generate new particles $\bar{\boldsymbol{\theta}}_n^i$, $i = 1, \ldots, N \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}_{n-1}^i)$, where $\kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}')$ is a Markov kernel, i.e., a probability distribution for $\boldsymbol{\theta}$ conditional on $\boldsymbol{\theta}'$. See Sec. IV for guidelines on the selection of this kernel. Intuitively, we can either jitter a few particles with arbitrary variance (while leaving most of them unperturbed) or jitter all particles with a controlled variance that decreases as N increases.

(ii) Estimation of likelihoods. The sequential approximation of Eqs. (16)–(18) yields biased estimates of the likelihoods $u_n(\boldsymbol{\theta}_n)$ [21]. This is discussed in Sec. IV. In Appendix A we provide details on the computation of the estimates $\hat{p}(\boldsymbol{x}_n|\bar{\boldsymbol{\theta}}_n^i, \boldsymbol{y}_{1,\ldots,n-1})$ and $\hat{u}_n(\boldsymbol{\theta}_n)$ using both the EnKF and the EKF. Other techniques (e.g., particle filters as in [21] or σ-point Kalman filters [51,52]) can be used as well.

C. Sequential quasi–Monte Carlo hybrid filter

The SMC method in the first layer of Algorithm 1 can be replaced by other schemes that rely on the point-mass representation of the posterior probability distribution $\mu_n(d\boldsymbol{\theta})$. It is possible to devise procedures based, for instance, on an unscented Kalman filter [53] or other σ-point Kalman methods [51,52] to obtain a Gaussian approximation of $\mu_n(\boldsymbol{\theta})$. Such Gaussian approximations, however, can be misleading when the posterior distribution is multimodal.

In this section we describe a NHF method (hence of the same class as Algorithm 1) where the SMC scheme is replaced by a sequential quasi–Monte Carlo (SQMC) procedure of the type introduced in [34]. The term quasi–Monte Carlo (QMC) refers to a class of deterministic methods for numerical integration [54] that employ low-discrepancy point sets (e.g., Halton sequences [55] or Sobol sequences [56]), instead of random sample sets, for the approximation of multidimensional integrals. In the context of QMC, *discrepancy* is defined to quantify how uniformly the points in a sequence are distributed into an arbitrary set S. Hence, the lowest discrepancy is attained when these points are equidistributed. The main advantage of (deterministic) QMC methods over (random) Monte Carlo schemes is that they can attain a faster rate of convergence relative to the number of points in the grid, N. Within a NHF, the use of QMC should lead to a better performance or complexity trade-off as long as the parameter dimension d_θ is relatively small. This is illustrated numerically for a stochastic two-scale Lorenz 96 model in Sec. VI.

The NHF based on the SQMC methodology of [34] can be obtained from Algorithm 1 if we replace the sampling and resampling steps typical of the SMC schemes by the generation of low-discrepancy point sets. Let $\{\boldsymbol{v}_n^i\}_{i=1}^N$ be a Halton sequence of low-discrepancy (deterministic) uniform samples [55]. These uniform samples can be used to generate low-discrepancy variates from other distributions via a number of methods.[2] For example, the Box-Muller transformation [58] can be used to generate pairs of independent, standard, normally distributed pseudorandom numbers. We explicitly indicate the use of low-discrepancy uniform numbers $\boldsymbol{v}_n^i$ in the generation of samples with general distributions by conditioning on $\boldsymbol{v}_n^i$. Hence, drawing the ith sample from the prior parameter PDF, $\boldsymbol{\theta}_0^i \sim p(\boldsymbol{\theta})$ is now replaced by $\boldsymbol{\theta}_0^i \sim p(\boldsymbol{\theta}|\boldsymbol{v}_0^i)$. In order to propagate the ith sample at time $n-1$, $\boldsymbol{\theta}_{n-1}^i$, into time n, we draw from the kernel $\kappa_N(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1}^i, \boldsymbol{v}_n^i)$. If sampling is needed in the second layer of filters [in order to compute the estimates $\hat{p}(\boldsymbol{x}_n|\bar{\boldsymbol{\theta}}_n^i, \boldsymbol{y}_{1,\ldots,n-1})$ and $\hat{u}_n(\bar{\boldsymbol{\theta}}_n^i)$] we use additional Halton sequences in a similar way.

In order to keep the low-discrepancy property across the resampling step, we additionally introduce the following functions (see [34] for details).

(i) One function is a discrepancy-preserving bijective map $\psi : \mathbb{R}^{d_x} \rightarrow [0,1]^{d_\theta}$. Several choices are possible for this function. Following [34], here we assume

$$\psi(\bar{\boldsymbol{\theta}}_n^i) = \left[1 + \exp\left(-\frac{\bar{\boldsymbol{\theta}}_n^i - \boldsymbol{\theta}_n^-}{\boldsymbol{\theta}_n^+ - \boldsymbol{\theta}_n^-}\right)\right]^{-1}, \tag{25}$$

where $\boldsymbol{\theta}_n^-$ and $\boldsymbol{\theta}_n^+$ are the d_θ-dimensional vectors whose jth components are, respectively,

$$[\boldsymbol{\theta}_n^-]_j = m_{n,j}^N - 2s_{n,j}^{2N}, \quad [\boldsymbol{\theta}_n^+]_j = m_{n,j}^N + 2s_{n,j}^{2N},$$

whereas $m_{n,j}^N = \sum_{i=1}^N w_n^i [\bar{\boldsymbol{\theta}}_n^i]_j$ and $s_{n,j}^{2N} = \sum_{i=1}^N w_n^i ([\bar{\boldsymbol{\theta}}_n^i]_j - m_{n,j}^N)^2$, $j = 1, \ldots, d_\theta$, are componentwise means and variances.

(ii) The other function is the inverse of the Hilbert curve, $\mathsf{h} : [0,1]^{d_\theta} \rightarrow [0,1]$, which is a continuous fractal space-filling curve that provides a locality-preserving map between a one-dimensional and a $d_{\boldsymbol{\theta}}$-dimensional space [59,60].

The SQMC-based NHF is outlined in the following algorithm.

Algorithm 2: Sequential quasi–Monte Carlo nested hybrid filter.

Inputs:

1. N (number of Monte Carlo samples to generate).
2. *A priori* PDFs $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x}_0)$.
3. A Markov kernel $\kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}')$ which, given $\boldsymbol{\theta}'$, generates jittered parameters $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$.

Procedure:

1. Initialization. Generate QMC uniform samples $\{\boldsymbol{v}_{-1}^i, \boldsymbol{v}_0^i\}_{i=1}^N$ in $[0,1)^{d_{\boldsymbol{\theta}}}$. Draw $\boldsymbol{\theta}_0^i \sim p(\boldsymbol{\theta}|\boldsymbol{v}_{-1}^i)$, $i = 1, \ldots, N$.
2. Recursive step $n \geqslant 1$.

(a) For $i = 1, \ldots, N$ perform the following.

(i) If $n = 1$, then draw $\bar{\boldsymbol{\theta}}_1^i \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}_0^i, \boldsymbol{v}_0^i)$; otherwise draw $\bar{\boldsymbol{\theta}}_n^i \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}_{n-1}^i, \tilde{\boldsymbol{v}}_{n-1}^{\mathsf{c}(i)})$ for $n \geqslant 2$.

(ii) Approximate $\hat{p}(\boldsymbol{x}_n|\bar{\boldsymbol{\theta}}_n^i, \boldsymbol{y}_{1,\ldots,n-1})$.

(iii) Use this approximation to compute the estimate

$$\hat{u}_n(\bar{\boldsymbol{\theta}}_n^i) = \int p(\boldsymbol{y}_n|\bar{\boldsymbol{\theta}}_n^i, \boldsymbol{x}_n)\hat{p}(\boldsymbol{x}_n|\bar{\boldsymbol{\theta}}_n^i, \boldsymbol{y}_{1,\ldots,n-1})d\boldsymbol{x}_n \tag{26}$$

and let $w_n^i \propto \hat{u}_n(\bar{\boldsymbol{\theta}}_n^i)$ be the normalized weight of $\bar{\boldsymbol{\theta}}_n^i$.

(b) Generate a QMC point set $\{\boldsymbol{v}_n^i\}_{i=1}^N$ in $[0,1)^{d_{\boldsymbol{\theta}}+1}$; let $\boldsymbol{v}_n^i = (v_n^i, \tilde{\boldsymbol{v}}_n^i) \in [0,1) \times [0,1)^{d_{\boldsymbol{\theta}}}$.

(c) Hilbert sort. Find a permutation b such that

$$(\mathsf{h} \circ \psi)\left(\bar{\boldsymbol{\theta}}_{\mathsf{n}}^{\mathsf{b}(1)}\right) \leqslant \cdots \leqslant (\mathsf{h} \circ \psi)\left(\bar{\boldsymbol{\theta}}_n^{\mathsf{b}(N)}\right) \quad \text{if } d_\theta = 1,$$

$$\bar{\boldsymbol{\theta}}_n^{\mathsf{b}(1)} \leqslant \cdots \leqslant \bar{\boldsymbol{\theta}}_n^{\mathsf{b}(N)} \quad \text{if } d_{\boldsymbol{\theta}} = 1.$$

(d) Resampling. Find a permutation c such that $v_n^{\mathsf{c}(1)} \leqslant \cdots \leqslant v_n^{\mathsf{c}(N)}$. For $i = 1, \ldots, N$, set $\boldsymbol{\theta}_n^i = \bar{\boldsymbol{\theta}}_n^j$ if and only if

$$\sum_{k=1}^{j-1} w_n^{\mathsf{b}(k)} < v_n^{\mathsf{c}(i)} \leqslant \sum_{k=1}^{j} w_n^{\mathsf{b}(k)}, \quad j \in \{1, \ldots, N\}.$$

Outputs: A set of particles $\{\boldsymbol{\theta}_n^i\}_{i=1}^N$ and a probability measure $\mu_n^N(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_n^i}(d\boldsymbol{\theta})$.

IV. CONVERGENCE ANALYSIS

The nested filtering schemes of Sec. III admit various implementations depending on how we choose to approximate the conditional PDF $p(\boldsymbol{x}_n|\boldsymbol{y}_{1,\ldots,n-1}, \boldsymbol{\theta})$, which in turn is needed to estimate the likelihood function and compute the importance weights $w_n^i \propto \hat{u}(\bar{\boldsymbol{\theta}}_n^i) \simeq u_n(\bar{\boldsymbol{\theta}}_n^i)$, $i = 1, \ldots, N$. For each choice of approximation method, the estimate $\hat{u}_n(\boldsymbol{\theta})$ may behave differently and yield different convergence properties. Here we assume that $\hat{u}_n(\boldsymbol{\theta})$ is a random variable with finite mean $\bar{u}_n(\boldsymbol{\theta}) = \mathbb{E}[\hat{u}_n(\boldsymbol{\theta})] < \infty$ and finite moments up to some prescribed order $p \geqslant 1$. Specifically, we make following assumption.

Assumption 1. Given $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$, the estimator $\hat{u}_n(\boldsymbol{\theta})$ is random and can be written as

$$\hat{u}_n(\boldsymbol{\theta}) = \bar{u}_n(\boldsymbol{\theta}) + m_n(\boldsymbol{\theta}), \tag{27}$$

[2] One can use a number of techniques used to produce random samples from a given uniform source. See [57] for a comprehensive description of the field, both for single and multivariate distributions.

where $m_n(\boldsymbol{\theta})$ is a zero-mean random variable satisfying $\mathbb{E}[m_n(\boldsymbol{\theta})^p] \leqslant \sigma^p < \infty$ for some prescribed $p \geqslant 1$. Furthermore, the mean $\bar{u}_n(\boldsymbol{\theta}) = \mathbb{E}[\hat{u}_n(\boldsymbol{\theta})]$ has the form

$$\bar{u}_n(\boldsymbol{\theta}) = u_n(\boldsymbol{\theta}) + b_n(\boldsymbol{\theta}), \tag{28}$$

where $b_n(\boldsymbol{\theta})$ is a deterministic and absolutely bounded bias function.

From here on, we use $D \subseteq \mathbb{R}^{d_\theta}$ to denote the support set of the parameter vector $\boldsymbol{\theta}$. Given a real function $a : D \to \mathbb{R}$, its absolute supremum is indicated as $\|a\|_\infty := \sup_{\boldsymbol{\theta}\in D} |a(\boldsymbol{\theta})|$. The set of absolutely bounded real functions on D is denoted by $B(D)$, i.e., $B(D) := \{(a : D \to \mathbb{R}) : \|a\|_\infty < \infty\}$. For our analysis we assume that $u_n \in B(D)$ and since we have also assumed the bias function b_n to be bounded, it follows that $\bar{u}_n \in B(D)$, i.e., $\|\bar{u}_n\|_\infty < \infty$. To be precise, we impose the following assumption.

Assumption 2. Given a fixed sequence of observations $y_{1,\ldots,n}$, the family of functions $\{\bar{u}_n(\boldsymbol{\theta}), \boldsymbol{\theta} \in D\}$ satisfies the following relations for each $n = 1, 2, \ldots$: (a) $\bar{u}_n \in B(D)$ and (b) $\bar{u}_n(\boldsymbol{\theta}) > 0$ for any $\boldsymbol{\theta} \in D$.

Since $\|u_n\|_\infty < \infty$, Assumption 2(a) follows from Assumption 1. Similarly, if $u_n(\boldsymbol{\theta}) > 0$ for all $\boldsymbol{\theta} \in D$ then Assumption 2(b) is a natural one [since $\hat{u}(\boldsymbol{\theta})$ is an estimator of a positive magnitude].

We will prove that, because of the bias $b_n(\boldsymbol{\theta})$, the approximation μ_n^N converges to the perturbed probability measure $\bar{\mu}_n$ induced by the mean function $\bar{u}_n$, instead of the true posterior probability measure μ_n induced by the true likelihood function u_n.

To be specific, the sequence of posterior measures $\mu_n, n \geqslant 1$, can be constructed recursively, starting from a prior μ_0, by means of the projective product operation [61], denoted $\mu_n = u_n \bullet \mu_{n-1}$. When u is a positive and bounded function and μ is a probability measure, the new measure $u \bullet \mu$ is defined in terms of its integrals. In particular, if $a \in B(D)$ then

$$\int a(\boldsymbol{\theta})(u \bullet \mu)(d\boldsymbol{\theta}) := \frac{\int a(\boldsymbol{\theta})u(\boldsymbol{\theta})\mu(d\boldsymbol{\theta})}{\int u(\boldsymbol{\theta})\mu(d\boldsymbol{\theta})}.$$

For conciseness, hereafter we use the shorthand

$$(a, \mu) := \int a(\boldsymbol{\theta})\mu(d\boldsymbol{\theta})$$

for the integral of a function $a(\boldsymbol{\theta})$ with respect to a measure $\mu(d\boldsymbol{\theta})$. With this notation, we can write

$$(a, \mu_n) = (a, u_n \bullet \mu_{n-1}) = \frac{(au_n, \mu_{n-1})}{(u_n, \mu_{n-1})}. \tag{29}$$

If instead of the true likelihood u_n we use the biased function $\bar{u}_n = u_n + b_n$ to update the posterior probability measure associated with the parameter vector $\boldsymbol{\theta}$ at each time n, then we obtain the new sequence of measures

$$\bar{\mu}_0 = \mu_0, \quad \bar{\mu}_n = \bar{u}_n \bullet \bar{\mu}_{n-1}, \quad n = 1, 2, \ldots,$$

where, according to the definition of the projective product,

$$(a, \bar{\mu}_n) = \frac{(a\bar{u}_n, \bar{\mu}_{n-1})}{(\bar{u}_n, \bar{\mu}_{n-1})}$$

for any integrable function $a(\boldsymbol{\theta})$. Note that the two sequences μ_n and $\bar{\mu}_n$ start from the same prior μ_0. Obviously, we recover the original sequence, i.e., $\bar{\mu}_n \to \mu_n$, when the bias vanishes, $b_n \to \mathbf{0}$.

In this section we prove that the approximation μ_n^N generated by a generic nested filter that satisfies Assumptions 1 and 2 converges to $\bar{\mu}_n$ in L_p, for each $n = 1, 2, \ldots$, under additional regularity assumption on the jittering kernel κ_N.

Assumption 3. The kernel κ_N used in the jittering step satisfies the inequality

$$\sup_{\boldsymbol{\theta}'\in D} \int |h(\boldsymbol{\theta}) - h(\boldsymbol{\theta}')|\kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}') \leqslant \frac{c_\kappa \|h\|_\infty}{\sqrt{N}} \tag{30}$$

for every $h \in B(D)$ and some constant $c_\kappa < \infty$ independent of N.

A simple kernel that satisfies Assumption 3 is [21]

$$\kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}') = (1 - \epsilon_N)\delta_{\boldsymbol{\theta}'}(d\boldsymbol{\theta}) + \epsilon_N \kappa(d\boldsymbol{\theta}|\boldsymbol{\theta}'),$$

where $0 < \epsilon_N \leqslant \frac{1}{\sqrt{N}}$ and $\kappa(d\boldsymbol{\theta}|\boldsymbol{\theta}')$ is an arbitrary Markov kernel with mean $\boldsymbol{\theta}'$ and finite variance, for example, $\kappa(d\boldsymbol{\theta}|\boldsymbol{\theta}') = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}', \tilde{\sigma}^2 \boldsymbol{I}_{d_\theta})$, where $\tilde{\sigma}^2 < \infty$ and $\boldsymbol{I}_{d_\theta}$ is the identity matrix. Intuitively, this kind of kernel changes each particle with probability ϵ_N and leaves it unmodified with probability $1 - \epsilon_N$.

Finally, we can state a general result on the convergence of Algorithm 1. For a real random variable x and $p \geqslant 1$, let $\|x\|_p$ denote the L_p-norm, i.e., $\|x\|_p := \mathbb{E}[|x|^p]^{1/p}$.

Theorem 1. Let the sequence of observations $y_{1,\ldots,n_0}$ be arbitrary but fixed, with $n_0 < \infty$, and choose an arbitrary function $h \in B(D)$. If Assumptions 1–3 hold, then

$$\left\|(h, \mu_n^N) - (h, \bar{\mu}_n)\right\|_p \leqslant \frac{c_n \|h\|_\infty}{\sqrt{N}}$$
$$\text{for } n = 0, 1, \ldots, n_0 \text{ and any } p \geqslant 1, \tag{31}$$

where $\{c_n\}_{0\leqslant n\leqslant n_0}$ is a sequence of finite constants independent of N.

For a proof of Theorem 1 see Appendix B.

We remark that Theorem 1 states that the approximate posterior probability measure output by Algorithm 1, μ_n^N, converges *not* to the true posterior measure μ_n, but to the biased version $\bar{\mu}_n$. Moreover, the latter depends on the choice of filters used in the second layer of Algorithm 1 (i.e., on the estimator of the likelihood $\hat{u}_n$). The value of this theorem is that it guarantees the numerical consistency of the nested hybrid filter: As we increase the computational effort (by increasing N), the random probability measure μ_n^N converges to a well-defined limit (and so do any point estimators that we may derive from it, e.g., the posterior mean estimator $\boldsymbol{\theta}_n^N$). The connection between this limit measure $\bar{\mu}_n$ and the true posterior measure μ_n is given by Assumption 1 and the projective product operation, namely,

$$\bar{\mu}_n = (u_n + b_n) \bullet \bar{\mu}_{n-1} \quad \text{while } \mu_n = u_n \bullet \mu_{n-1},$$

with both sequences starting with a common prior measure $\bar{\mu}_0 = \mu_0$. The practical performance of the proposed schemes (with finite N) is explored numerically in the following.

V. STOCHASTIC LORENZ 96 MODEL

In order to assess the proposed methods numerically, we have applied them to a stochastic discrete-time version of the two-scale Lorenz 96 model [40,43,62]. The latter is a deterministic system of nonlinear differential equations that displays some key features of atmospheric dynamics (including chaotic behavior) in a relatively simple model of arbitrary dimension (the number d_x of dynamic variables can be scaled as needed). The model consists of two sets of dynamic variables $\boldsymbol{x}$ and z. The system of stochastic differential equations takes the form

$$\begin{aligned} d\boldsymbol{x} &= \boldsymbol{f}_1(\boldsymbol{x}, z, \boldsymbol{\alpha})dt + \sigma d\boldsymbol{w}, \\ dz &= \boldsymbol{f}_2(\boldsymbol{x}, z, \boldsymbol{\alpha})dt + \bar{\sigma} d\bar{\boldsymbol{w}}, \end{aligned} \tag{32}$$

where $\boldsymbol{x}(t)$ and $z(t)$ represent the slow and fast variables, respectively, $\boldsymbol{w}$ and $\bar{\boldsymbol{w}}$ are Wiener processes, $\sigma, \bar{\sigma} > 0$ are known scale parameters, and $\boldsymbol{\alpha}$ is a parameter vector of dimension $d_\alpha = 4$. Let us assume there are d_x slow variables x_j, $j = 0, \ldots, d_x - 1$, and L fast variables per slow variable, i.e., z_l, $l = 0, \ldots, d_x L - 1$, overall. The maps $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are $\mathbb{R}^{d_x} \times \mathbb{R}^L \times \mathbb{R}^{d_\alpha} \to \mathbb{R}^{d_x}$ and $\mathbb{R}^L \times \mathbb{R}^{d_x} \times \mathbb{R}^{d_\alpha} \to \mathbb{R}^L$ functions, respectively, which can be written (skipping the time index t) as

$$\begin{aligned} \boldsymbol{f}_1 &= [f_{1,0}, \ldots, f_{1,d_x-1}]^\top, \\ f_{1,j}(\boldsymbol{x}, z, \boldsymbol{\alpha}) &= -x_{j-1}(x_{j-2} - x_{j+1}) - x_j \\ &\quad + F - \frac{HC}{B} \sum_{l=(j-1)L}^{Lj-1} z_l, \\ \boldsymbol{f}_2 &= [f_{2,0}, \ldots, f_{2,d_x L-1}]^\top, \\ f_{2,l}(z, \boldsymbol{\alpha}) &= -CBz_{l+1}(z_{l+2} - z_{l-1}) - Cz_l \\ &\quad + \frac{CF}{B} + \frac{HC}{B} x_{\lfloor (l-1)/L \rfloor}, \end{aligned} \tag{33}$$

where $\boldsymbol{\alpha}$ contains the parameters F, C, H, and B. The forcing parameter F controls the turbulence of the chaotic flow, C determines the timescale of the fast variables $\{z_l\}_{l\geqslant 0}$, H controls the strength of the coupling between the fast and slow variables, and B determines the amplitude of the fast variables [40]. The dynamic variables are assumed to be arranged on a circular structure, hence the operations on the j indices are modulo d_x and operations on the l indices are modulo L. This means that for any integer k, $j + k \equiv (j + k)\text{mod}d_x$ and $l + k \equiv (l + k)\text{mod}L$. The notation $\lfloor a \rfloor$ indicates the truncation of a positive real number a to the closest integer smaller than a.

We apply the fourth-order Runge-Kutta (RK4) method [45] to obtain a discrete-time version of the two-scale Lorenz 96 model. To be specific, we numerically integrate Eq. (32) by means of the stochastic difference equations

$$\begin{aligned} \boldsymbol{x}_n &= \boldsymbol{x}_{n-1} + \boldsymbol{F}_1^m(\boldsymbol{x}_{n-1}, z_{n-1}, \boldsymbol{\alpha}, h, \sigma \boldsymbol{w}_n), \\ z_n &= z_{n-1} + \boldsymbol{F}_2^m(\boldsymbol{x}_{n-1}, z_{n-1}, \boldsymbol{\alpha}, h, \bar{\sigma} \bar{\boldsymbol{w}}_n), \end{aligned} \tag{34}$$

where $h > 0$ is the integration step size and $\boldsymbol{w}_n$ and $\bar{\boldsymbol{w}}_n$ are sequences of independent and identically distributed standard Gaussian RVs.

We assume that the observations are linear but can only be collected from this system once every T discrete-time steps. Moreover, only one out of K slow variables can be observed. Therefore, the observation process has the form

$$\boldsymbol{y}_n = \begin{bmatrix} x_{K,nT} \\ x_{2K,nT} \\ \vdots \\ x_{d_y K,nT} \end{bmatrix} + \boldsymbol{r}_n, \tag{35}$$

where $n = 1, 2, \ldots$ and $\boldsymbol{r}_n$ is a sequence of independent and identically distributed RVs with common PDF $\mathcal{N}(\boldsymbol{r}_n | 0, \sigma_0^2 \mathbf{I}_{d_y})$. Note that fast variables are not observed.

In our computer experiments, the system (33) is often employed to generate both ground-truth values for the slow variables $\{\boldsymbol{x}_n\}_{n\geqslant 0}$ and synthetic observations $\{\boldsymbol{y}_n\}_{n\geqslant 1}$. However, since in real-world problems models are inherently imperfect (modeling errors always exist to some extent), a different and simpler version of the Lorenz 96 model is used in order to implement the NHFs. In this simpler version, the contribution of the fast variables to the jth equation is substituted by a polynomial function of the slow variable x_j. To be specific, we use the SDE

$$\begin{aligned} dx_j &= f_j(\boldsymbol{x}, \boldsymbol{\theta})dt + \sigma dw_j \\ &= [-x_{j-1}(x_{j-2} - x_{j+1}) - x_j + F - \ell(x_j, \mathsf{a})]dt + \sigma dw_j, \\ j &= 0, \ldots, d_x - 1, \end{aligned} \tag{36}$$

where $\mathsf{a} = [\mathsf{a}_1, \mathsf{a}_2]^\top$ is a (constant) parameter vector, $\boldsymbol{\theta} = [F, \mathsf{a}^\top]^\top$ contains all the parameters, the function $\ell(x_j, \mathsf{a}) \in \mathbb{R}$ is a polynomial ansatz for the coupling term $\frac{HC}{B} \sum_{l=(j-1)L}^{Lj-1} z_l$ in (33), and w_j is a standard Wiener process. Note that in this simplified model we have removed the fast variables completely. In this paper we assume that $\ell(x_j, \mathsf{a})$ is a polynomial in x_j of degree 2, characterized by the coefficients a_1 and a_2 as

$$\ell(x_j, \mathsf{a}) = \mathsf{a}_1 x_j^2 + \mathsf{a}_2 x_j.$$

Then the system (34) can be replaced by

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \bar{\boldsymbol{F}}^m(\boldsymbol{x}_{n-1}, \boldsymbol{\theta}, h, \sigma \boldsymbol{w}_n), \tag{37}$$

where $\bar{\boldsymbol{F}}^m$ is the RK4 approximation of the function $\boldsymbol{f} = [f_0, \ldots, f_{d_x-1}]^\top$ in Eq. (36). Assuming $\boldsymbol{r}_n$ is a sequence of independent and identically distributed noise terms with Gaussian probability distribution $p(\boldsymbol{r}) = \mathcal{N}(\boldsymbol{r} | 0, \sigma_0^2 \boldsymbol{I}_{d_y})$, then

$$p(\boldsymbol{y}_n | \boldsymbol{x}_n, \boldsymbol{\theta}) = \mathcal{N}\big(\boldsymbol{y}_n \big| \boldsymbol{x}_n, \sigma_0^2 \boldsymbol{I}_{d_y}\big), \tag{38}$$

which denotes a d_y-dimensional Gaussian density with zero mean and covariance matrix $\sigma_0^2 \boldsymbol{I}_{d_y}$, where $\boldsymbol{I}_{d_y}$ is the $d_y \times d_y$ identity matrix.

VI. NUMERICAL RESULTS

We have conducted computer simulations to illustrate the performance of the proposed NHF methods. In particular, we have carried out computer experiments for six different schemes: the NPF of [21], the two-stage filter of [26], and four NHFs that rely on the SQMC and the SMC, both in combination with EKFs or EnKFs. Then two different versions of Algorithm 1 (SMC EKF and SMC EnKF) and Algorithm 2 (SQMC EKF and SQMC EnKF) are simulated. The simulation setup is described below, followed by the discussion of our numerical results in Sec. VI B.

A. Simulation setup

For our computer experiments we have used the two-scale Lorenz 96 model of Eq. (32) in order to generate reference signals $\tilde{\boldsymbol{x}}_k$, $k = 0, 1, \ldots$, used as the ground truth for the assessment of the estimators, and sequences of observations $\boldsymbol{y}_n$, $n = 1, 2, \ldots$, as in Eq. (25). The model is integrated using the RK4 method with Gaussian perturbations [45] [as outlined in Eq. (34)]. The integration step is set to $h = 5 \times 10^{-3}$ continuous-time units through all experiments and the fixed model parameters are $F = 8$, $H = 0.75$, $C = 10$, and $B = 15$. For all experiments, we assume that there are $L = 10$ fast variables per slow variable, hence the total dimension of the model is $10d_x$ (with different values of d_x for different experiments). The noise scaling factors are $\sigma = \frac{h}{4} = 0.25 \times 10^{-3}$ and $\sigma_0 = 4$, both assumed known. We assume that half of the slow variables are observed in Gaussian noise, i.e., $K = 2$.

We assess the accuracy of the estimation algorithms in terms of the MSE of the predictors of the dynamic variables. For the NHFs, these estimators take the form

$$\hat{\boldsymbol{x}}_n = \sum_{i=1}^{N} w_n^i \hat{\boldsymbol{x}}_n^i, \tag{39}$$

where $\hat{\boldsymbol{x}}_n^i$ is the posterior-mean estimate obtained from the approximate filter $\hat{p}(\boldsymbol{x}_n | \boldsymbol{y}_{1,\ldots,n}, \bar{\boldsymbol{\theta}}_n^i)$, which can be expressed as $\mathcal{N}(\boldsymbol{x}_n | \hat{\boldsymbol{x}}_n^i, \boldsymbol{P}_n^i)$, since the approximation is Gaussian. In the plots, however, we show the empirical MSE per dimension resulting directly from the simulations,

$$\mathrm{MSE}_n = \frac{1}{d_x} \|\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n\|^2. \tag{40}$$

averaged over 100 independent simulation runs, all of them being of 40 continuous-time units of duration.

The simulations presented below include running times for the different methods. They have been coded in MATLAB R2016a and run on a computer with 64 GB of DRAM and equipped with two Intel Xeon E5-2680 processors (running at 2.80 GHz) with ten cores each and HyperThreading as well as an Intel Xeon Phi co-processor.

B. Results

Table I shows a comparison of the performance of the NPF, the two-stage filter, and the four NHFs, based on the use of SMC, SQMC, EKF, and EnKF schemes as described in Sec. III, in terms of their running times and the MSE of the state estimators (averaged over time and dimensions). We have carried out this computer simulation for a model with dimension $d_x = 40$ and a gap between observations of $hm = 0.05$ continuous-time units. All NHF algorithms work with $N = 100$ particles for the approximation of the posterior distributions of the fixed parameters, using $M = d_x = 40$ samples per each EnKF in the second layer. It can be seen that the highest error is achieved by the NPF, followed by the two-stage filter method. The NPF is also the algorithm that takes the longest running time. Both NHFs using EKF attain the least MSE with the smallest running time. In order to improve the performance of the NPF, the numbers of particles M and N would have to be considerably increased, but this would increase the running times correspondingly [the complexity of the NPF is $O(NM)$ [21]].

TABLE I. Running times and average MSE (over time and state dimensions) for the NPF, the two-stage filter, and four NHFs, based on the SQMC, the SMC, the EKF, and the EnKF, respectively.

Algorithm	Running time (min)	MSE
NHF: SQMC EKF	2.16	0.46
NHF: SMC EKF	2.27	0.49
NHF: SQMC EnKF	6.83	0.62
NHF: SMC EnKF	7.12	0.95
two-stage filter ($N = 600$, $M = 400$)	6.85	4.59
NPF ($N = M = 800$)	17.96	11.91

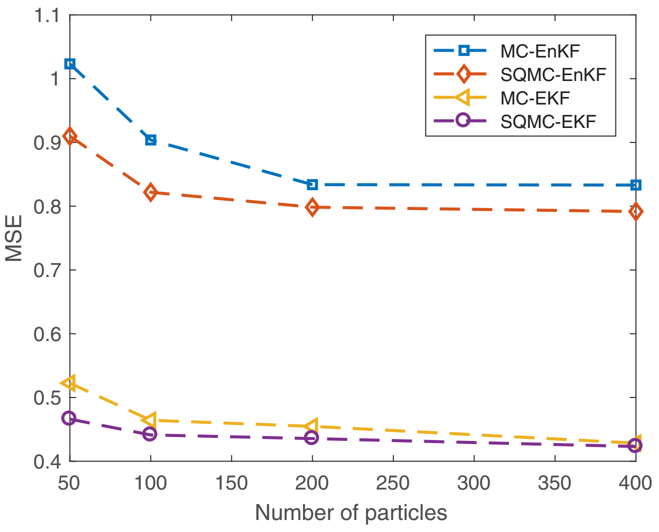

FIG. 1. MSE of the different NHFs depending on the number of particles used in the first-layer of the filter.

In the next experiment we assess the performance of the different NHFs depending on the number of particles used in the first layer of the filter, in order to choose appropriately this number to carry out the following computer experiments. For this purpose, we consider a model with dimension $d_x = 100$, a gap between observations of $hm = 0.05$ continuous-time units and a number of particles that ranges from 50 to 400. Figure 1 shows the numerical results for this experiment. We observe that the MSE for the four algorithms stabilizes quickly. At the sight of these results, we set $N = 100$ for all remaining experiments. Additionally, Fig. 1 also shows the difference between the NHFs. Specifically, we see that using SQMC in the first layer, we can slightly improve the performance. For this reason, in the following experiments we only simulate NHFs that rely on SQMC. Moreover, it is easy to observe that the filters that use EKFs in the second layer obtain better results.

In the next set of computer experiments we compare the SQMC EKF and the SQMC EnKF methods in terms of their average MSE and their running times for different values of the state dimension d_x and the gap between consecutive observations m (in discrete-time steps). For each combination

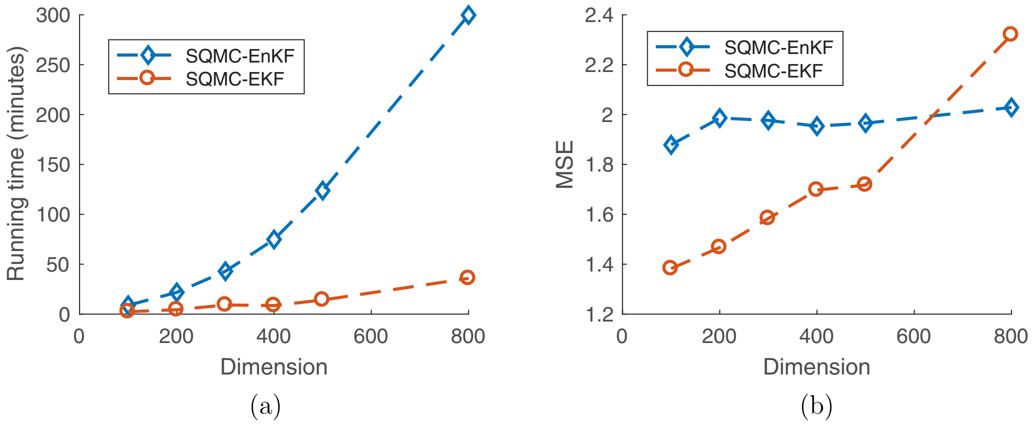

FIG. 2. Comparison of the SQMC EKF (red lines) and SQMC EnKF (blue lines) in terms of (a) their running time and (b) their MSE as the state dimension d_x increases, with a fixed gap between observations of $T = 20$ discrete time steps.

of d_x and m we have carried out 100 independent simulation runs. The number of particles in the parameter space is fixed, $N = 100$, for all simulations, but the size of the ensemble in the EnKFs is adjusted to the dimension, in particular, we set $M = d_x$.

Figure 2 shows the running times [Fig. 2(a)] and the average MSE [Fig. 2(b)] attained by the two SQMC NHFs when the state dimension d_x ranges from 100 to 800. The gap between observations is fixed to $m = 20$ (i.e., 0.1 time units versus 0.05 in Fig. 1). We observe that the SQMC EKF method attains significantly lower running times compared to the SQMC EnKF, since the former increases linearly with dimension while the latter increases its cost exponentially. However, the SQMC EKF obtains an MSE which increase with the dimension d_x, while the values of the MSE for the SQMC EnKF method are steady with respect to d_x.

Next, Fig. 3 displays the running times and the average MSEs attained by the two NHFs as we increase the gap between observations from $m = 10$ to $m = 100$ (hence, from $hm = 0.05$ to $hm = 0.50$ continuous-time units). The dimension of the state for this experiment is fixed to $d_x = 100$. Note that, as the gap m increases, fewer data points are effectively available for the estimation of both the parameters and the states. We observe again that the SQMC EnKF is computationally more costly than the SQMC EKF, however it attains a consistently smaller MSE when the gap between observations increases, suggesting that it may be a more efficient algorithm in data-poor scenarios.

Finally, we show results for a computer experiment in which we have used the SQMC EnKF method to estimate the parameters F and a and track the state variables of the two-scale Lorenz system with dimension $d_x = 5000$ and a gap between consecutive observations of $hm = 0.05$ continuous-time units. As in the rest of computer simulations, the number of particles used to approximate the sequence of parameter posterior distributions is $N = 100$.

Figure 4 shows the true state trajectories, together with their estimates, for the first two slow state variables of the two-scale Lorenz 96 model. We note that the first variable $x_1(t)$ is observed in Gaussian noise (with $\sigma_0 = 4$), while the second variable $x_2(t)$ is not observed. The accuracy of the estimation is similar though over the 20 continuous-time units of the simulation run (corresponding to 20×10^3 discrete-time steps), achieving MSE $\approxeq 0.87$. Taking into account the steadiness of the MSE with respect to dimension of the SQMC EnKF in Fig. 2(a) and the values of the MSE shown

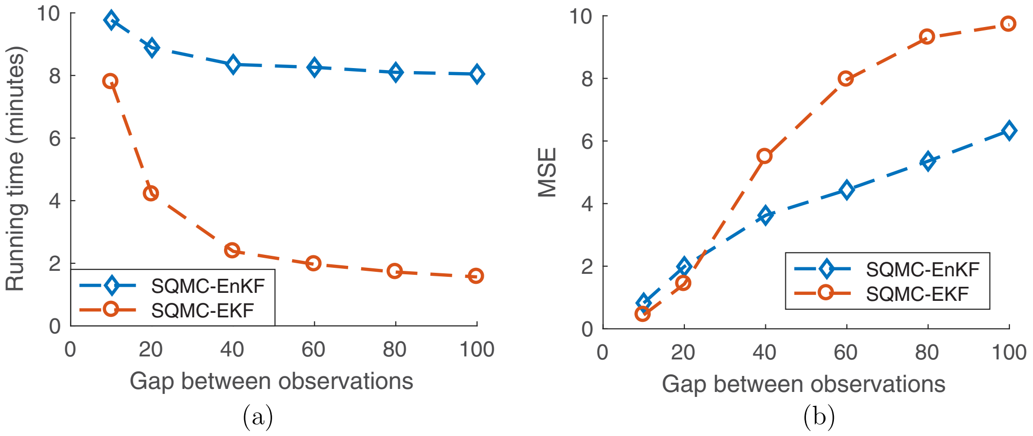

FIG. 3. Comparison of the SQMC EKF (red lines) and SQMC EnKF (blue lines) in terms of (a) their running time and (b) their MSE as the gap between observations m increases, with fixed state dimension $d_x = 100$.

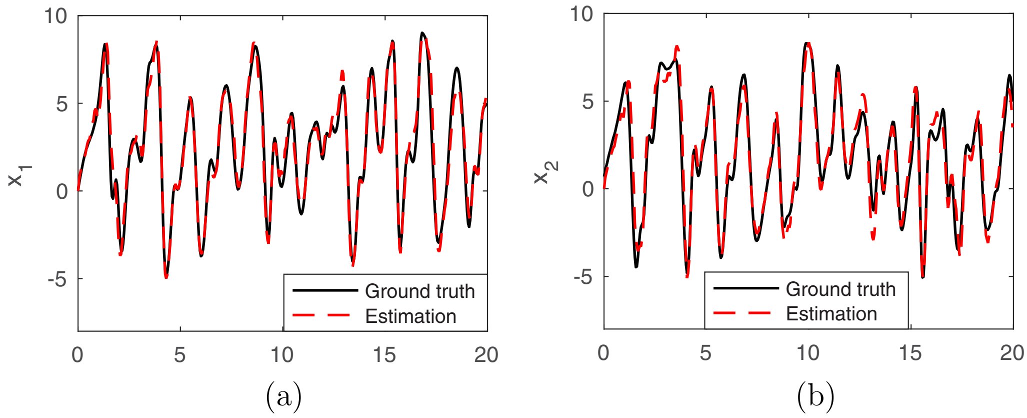

FIG. 4. Sequences of state values (black line) and estimates (dashed red line) in (a) x_1 and (b) x_2 over time. Variable x_1 is observed (in Gaussian noise), while x_2 is unobserved.

in Fig. 3(b) for the gap selected in this experiment ($m = 10$), the results obtained are within the expected range.

In Fig. 5 we observe the estimated posterior PDFs of the fixed parameters F, a_1, and a_2, together with the reference values. Note that the value $F = 8$ is the ground truth, but the values of a_1 and a_2 are genie-aided least-squares estimates obtained by observing directly the fast variables of the two-scale model. Figure 5(a) displays the true value $F = 8$ (vertical line) together with the approximate posterior PDF generated by the same Euler algorithm. We observe that nearly all probability mass is allocated close to the true value. In Fig. 5(b) we compare the approximate PDF of the coefficients $\mathsf{a} = [\mathsf{a}_1, \mathsf{a}_2]^\top$ produced by the NHF (dashed contour lines) with a kernel density estimator computed from the least-squares genie-aided estimates obtained from 100 independent simulations with the same setting (solid contour lines). The modes of the two PDFs are slightly shifted, but the two functions are otherwise similar. The genie-aided estimate of a is located in a high-probability region of the density function computed by the NHF.

VII. CONCLUSION

We have introduced a nested filtering methodology to recursively estimate the static parameters and the dynamic variables of nonlinear dynamical systems. The proposed framework combines a recursive Monte Carlo approximation method to compute the posterior probability distribution of the static parameters with a variety of filtering techniques to estimate the posterior distribution of the state variables of the system. In particular, we have investigated the use of Gaussian filters, as they admit fast implementations that can be well suited to high-dimensional systems. As a result, we have proposed a class of nested hybrid filters that combine Monte Carlo and quasi–Monte Carlo schemes for the (moderate-dimensional) unknown static parameters of the dynamical system with either extended Kalman filtering or ensemble Kalman filtering for the (higher-dimensional) time-varying states. Additionally, when sequential MC is applied in the first layer of the NHF scheme, we have proved that the algorithm converges with rate $O(N^{-1/2})$ to a well-defined limit distribution. We have presented numerical results for a two-scale

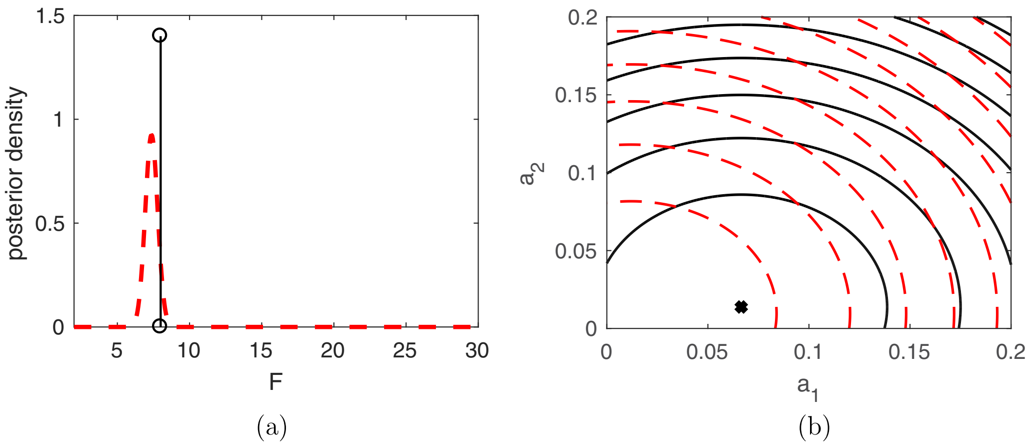

FIG. 5. Posterior density of the parameters $\mathsf{a} = [\mathsf{a}_1, \mathsf{a}_2]^\top$ and F at $t = 5$ in a 5000-dimensional Lorenz 96 model (red dashed lines). (a) The true value of F is indicated by a black vertical line, while (b) the location of reference values of a is marked by a black cross. Note that there is no ground truth for the parameters in a.

stochastic Lorenz 96 system, a model commonly used for the assessment of data assimilation methods in the geophysics. We illustrate the average performance of the methods in terms of estimation errors and running times and show numerical results for a 5000-dimensional system. This has been achieved with a relatively inefficient implementation of the method running on a desktop computer, hence we expect that the method can be applied to much larger-scale systems using adequate hardware and software.

ACKNOWLEDGMENTS

This research was partially supported by the Spanish Agencia Estatal de Investigación of Spain [Projects No. TEC2015-69868-C2-1-R (ADVENTURE) and No. TEC2017-86921-C2-1-R (CAIMAN)] and the Office of Naval Research Global (Grant Award No. N62909-15-1-2011). I.P.M. also acknowledges partial support from the Ministry of Education and Science of the Russian Federation, Grant Agreement No. 074-02-2018-330.

APPENDIX A: SUMMARY OF NHF ALGORITHMS

In this appendix we provide details of the four versions of the NHF used in the experiments of Sec. VI. The differences between the specific methods depend on whether the posterior distribution of the parameters is approximated using either an SMC or an SQMC scheme and whether the posterior (approximate) PDFs $\hat{p}(\boldsymbol{x}_n|\boldsymbol{y}_{1,\dots,n-1},\bar{\boldsymbol{\theta}}_n^i)$ and $\hat{p}(\boldsymbol{x}_n|\boldsymbol{y}_{1,\dots,n},\bar{\boldsymbol{\theta}}_n^i)$, which are needed to evaluate the importance weights $w_n^i \propto \hat{u}_n(\bar{\boldsymbol{\theta}}_n^i)$, are computed using either an EKF or an EnKF method.

In order to implement the EKF scheme, the state and the observation functions ($\boldsymbol{f}$ and $\boldsymbol{g}$) are assumed nonlinear and differentiable and therefore the mean $\boldsymbol{x}_n^i$ and the covariance matrix $\boldsymbol{P}_n^i$ can be directly calculated. This can be done by computing their respective Jacobian matrices ($\boldsymbol{J}_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{\theta}}$ and $\boldsymbol{J}_{\boldsymbol{g},\boldsymbol{x},\boldsymbol{\theta}}$) evaluated at the point $\boldsymbol{x}$ in the state space and $\boldsymbol{\theta}$ in the parameter space. In the EnKF scheme [63] the approximate filter $\hat{p}(\boldsymbol{x}_n|\boldsymbol{y}_{1,\dots,n},\bar{\boldsymbol{\theta}}_n^i)$ is represented by an ensemble of M Monte Carlo particles $\{\boldsymbol{x}_n^{i,j}\}_{j=1}^M$, which can be combined to yield an empirical covariance matrix $\boldsymbol{P}_n^{i,M}$, hence there is no need to assume that the state or observation functions are differentiable.

Each ensemble can be stored in a $d_x \times M$ matrix $\boldsymbol{X}_n^i = [\boldsymbol{x}_n^{i,1},\boldsymbol{x}_n^{i,2},\dots,\boldsymbol{x}_n^{i,M}]$. The ith mean and the ith covariance matrix can be computed as

$$\bar{\boldsymbol{x}}_n^i = \frac{1}{M}\boldsymbol{X}_n^i\boldsymbol{1}, \quad \bar{\boldsymbol{P}}_n^i = \frac{1}{M}\tilde{\boldsymbol{X}}_n^i\big(\tilde{\boldsymbol{X}}_n^i\big)^\top,$$

respectively, where $\boldsymbol{1} = [1,\dots,1]^\top$ is an M-dimensional column vector and $\tilde{\boldsymbol{X}}_n^i = \boldsymbol{X}_n^i - \bar{\boldsymbol{x}}_n^i\boldsymbol{1}^\top$ is an ensemble of deviations from $\bar{\boldsymbol{x}}_n^i$. We hence write $\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{X}_n^i)$ as shorthand for the Gaussian PDF $\mathcal{N}(\boldsymbol{x}_n|\bar{\boldsymbol{x}}_n^i,\bar{\boldsymbol{P}}_n^i)$.

We assume that the prior PDF of the state is Gaussian with known mean and covariance matrix, namely,

$$p(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0|\bar{\boldsymbol{x}}_0,\bar{\boldsymbol{P}}_0). \tag{A1}$$

The noise terms in the state space model are also assumed Gaussian, with zero mean and known covariance matrices, specifically,

$$\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{w}_k|\boldsymbol{0},\boldsymbol{Q}), \quad \boldsymbol{r}_k \sim \mathcal{N}(\boldsymbol{r}_k|\boldsymbol{0},\boldsymbol{T}) \tag{A2}$$

represent the state and observation noise respectively.

Below, we describe algorithms that rely on either Monte Carlo (MC) or QMC sampling schemes. Our simulations show that QMC can attain better performance, but this is only true if this sampling scheme is used in both the parameter space and the state space. To be specific, the methods that we have assessed are the following.

(i) The SMC EnKF (Algorithm 3) uses a sequential MC sampling procedure in the parameter space and a bank of EnKFs for the computation of the weights. Sampling inside each EnKF is standard MC.

(ii) The SQMC EnKF (Algorithm 4) uses sequential QMC sampling in the parameter space a QMC inside the EnKFs.

(iii) The SMC EKF (Algorithm 5) relies on sequential MC sampling for the parameters and uses EKFs for the weight computation. There is no need for sampling in the EKF algorithm.

(iv) The SQMC EKF (Algorithm 6) performs sequential QMC sampling in the parameter space and computes weights using EKFs.

Algorithm 3: The SMC-EnKF.

1. Initialization. Draw N independent and identically distributed particles $\boldsymbol{\theta}_0^i \sim \mu_0(d\boldsymbol{\theta})$ and $\{\boldsymbol{x}_0^{i,j}\} \sim p(\boldsymbol{x}_0)$, $i = 1,\dots,N$, $j = 1,\dots,M$. Let $\boldsymbol{X}_0^i = [\boldsymbol{x}_0^{i,1},\dots,\boldsymbol{x}_0^{i,M}]$, $i = 1,\dots,N$.

2. Recursive step. At time $n-1$, we have obtained $\mu_{n-1}^N(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_{n-1}^i}(d\boldsymbol{\theta})$ and, for each $i = 1,\dots,N$, $\hat{p}(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{1,\dots,n-1},\boldsymbol{\theta}_{n-1}^i) = \mathcal{N}(\boldsymbol{x}_{n-1}|\boldsymbol{X}_{n-1}^i)$.

(a) Prediction step.

(i) Draw $\bar{\boldsymbol{\theta}}_n^i \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}_{n-1}^i)$, $i = 1,\dots,N$.

(ii) For each $i = 1,\dots,N$ compute

$$\hat{\boldsymbol{X}}_n^i = \boldsymbol{X}_{n-1}^i + \boldsymbol{F}^m\big(\boldsymbol{X}_{n-1}^i,\bar{\boldsymbol{\theta}}_n^i,h,\sigma\boldsymbol{W}_n^i\big), \tag{A3}$$

where $\boldsymbol{W}_n^i = [\boldsymbol{w}_n^{i,1},\dots,\boldsymbol{w}_n^{i,M}]$, $i = 1,\dots,N$, is an $mqd_x \times M$ matrix of Gaussian perturbations (q denotes the order of the underlying RK integrator). This is done by generating $mqd_x \times M$ QMC random variates[3] and then applying the inversion method [57] to generate the Gaussian variates (one per uniform sample).

(iii) Set $\hat{p}(\boldsymbol{x}_n|\boldsymbol{y}_{1,\dots,n-1},\bar{\boldsymbol{\theta}}_n^i) = \mathcal{N}(\boldsymbol{x}_n|\hat{\boldsymbol{X}}_n^i)$.

(b) Update step.

(i) For $i = 1,\dots,N$, compute

$$\bar{\boldsymbol{M}}_n^i = \frac{1}{M}\tilde{\boldsymbol{X}}_n^i\big(\tilde{\boldsymbol{Z}}_n^i\big)^\top, \tag{A4}$$

$$\bar{\boldsymbol{S}}_n^i = \frac{1}{M}\tilde{\boldsymbol{Z}}_n^i\big(\tilde{\boldsymbol{Z}}_n^i\big)^\top + \boldsymbol{T}, \tag{A5}$$

$$\boldsymbol{K}_n^i = \bar{\boldsymbol{M}}_n^i\big(\bar{\boldsymbol{S}}_n^i\big)^{-1}, \tag{A6}$$

$$\check{\boldsymbol{X}}_n^i = \hat{\boldsymbol{X}}_n^i + \boldsymbol{K}_n^i\big(\boldsymbol{y}_n\boldsymbol{1}^\top - \bar{\boldsymbol{Y}}_n^i\big), \tag{A7}$$

[3] Using functions haltonset, scramble, and net in MATLAB, one can easily obtain a Halton quasirandom point set.

where $\boldsymbol{T} = \sigma_0^2 \boldsymbol{I}_{d_y}$ is the measurement noise covariance, $\bar{\boldsymbol{y}}_n = \frac{1}{M}\bar{\boldsymbol{Y}}_n^i \mathbf{1}$, and $\bar{\boldsymbol{x}}_n^i = \frac{1}{M}\hat{\boldsymbol{X}}_n^i\mathbf{1}$, with $\bar{\boldsymbol{Y}}_n^i = \boldsymbol{g}(\hat{\boldsymbol{X}}_n^i,\boldsymbol{\theta}) + \boldsymbol{R}_n^i$ and $\boldsymbol{R}_n^i = [\boldsymbol{r}_n^1,\dots,\boldsymbol{r}_n^M]$ a matrix of Gaussian perturbations. Further, $\tilde{\boldsymbol{X}}_n^i$ and $\tilde{\boldsymbol{Z}}_n^i$ are calculated as

$$\tilde{\boldsymbol{X}}_n^i = \hat{\boldsymbol{X}}_n^i - \bar{\boldsymbol{x}}_n^i \mathbf{1}^\top, \tag{A8}$$

$$\tilde{\boldsymbol{Z}}_n^i = \boldsymbol{g}\big(\hat{\boldsymbol{X}}_n^i,\boldsymbol{\theta}\big) - \bar{\boldsymbol{y}}_n^i\mathbf{1}^\top. \tag{A9}$$

(ii) Compute $\hat{u}(\bar{\boldsymbol{\theta}}_n^i) = \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{g}(\bar{\boldsymbol{x}}_n^i,\bar{\boldsymbol{\theta}}_n^i),\bar{\boldsymbol{S}}_n^i)$ and obtain the normalized weights

$$w_i = \frac{\hat{u}\big(\bar{\boldsymbol{\theta}}_n^i\big)}{\sum_{j=1}^N \hat{u}_n\big(\bar{\boldsymbol{\theta}}_n^j\big)}, \quad i = 1,\dots,N. \tag{A10}$$

(iii) Set the filter approximation

$$\hat{p}\big(\boldsymbol{x}_n\big|\boldsymbol{y}_{1,\dots,n},\bar{\boldsymbol{\theta}}_n^i\big) = \mathcal{N}\big(\boldsymbol{x}_n\big|\check{\boldsymbol{X}}_n^i\big). \tag{A11}$$

(c) Resampling. Draw indices $j_1,\dots,j_N$ from the multinomial distribution with probabilities $w_n^1,\dots,w_n^N$. Then set

$$\boldsymbol{\theta}_n^i = \bar{\boldsymbol{\theta}}_n^{j_i}, \quad \boldsymbol{X}_n^i = \check{\boldsymbol{X}}_n^{j_i} \tag{A12}$$

for $i = 1,\dots,N$. Hence

$$\hat{p}\big(\boldsymbol{x}_n\big|\boldsymbol{y}_{1,\dots,n},\boldsymbol{\theta}_n^i\big) = \mathcal{N}\big(\boldsymbol{x}_n\big|\boldsymbol{X}_n^i\big),$$

$$\mu_n^N(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_n^i}(d\boldsymbol{\theta}).$$

Algorithm 4: The SQMC EnKF.

1. Initialization. Generate QMC uniform samples $\{\boldsymbol{v}_{-1}^i,\boldsymbol{v}_0^i\}$ in $[0,1)^{d_\theta}$ and $\bar{\boldsymbol{v}}_{-1}^i$ in $[0,1)^{d_x}$. Draw $\boldsymbol{\theta}_0^i \sim \mu_0(d\boldsymbol{\theta}|\boldsymbol{v}_{-1}^i)$ and $\{\boldsymbol{x}_0^{i,j}\} \sim p(\boldsymbol{x}_0|\bar{\boldsymbol{v}}_{-1}^i)$, $i = 1,\dots,N$, $j = 1,\dots,M$. Let $\boldsymbol{X}_0^i = [\boldsymbol{x}_0^{i,1},\dots,\boldsymbol{x}_0^{i,M}]$, $i = 1,\dots,N$.

2. Recursive step $n \geqslant 1$. At time $n-1$, we have obtained $\mu_{n-1}^N(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_{n-1}^i}(d\boldsymbol{\theta})$ and, for each $i = 1,\dots,N$, $\hat{p}(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{1,\dots,n-1},\boldsymbol{\theta}_{n-1}^i) = \mathcal{N}(\boldsymbol{x}_{n-1}|\boldsymbol{X}_{n-1}^i)$.

(a) Prediction step.

(i) If $n = 1$, draw $\bar{\boldsymbol{\theta}}_n^i \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}_{n-1}^i,\boldsymbol{v}_0^i)$; otherwise draw $\bar{\boldsymbol{\theta}}_n^i \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}_{n-1}^i,\tilde{\boldsymbol{v}}_{n-1}^{\mathsf{c}(i)})$, $i = 1,\dots,N$, for $n \geqslant 2$.

(ii) For each $i = 1,\dots,N$ compute

$$\hat{\boldsymbol{X}}_n^i = \boldsymbol{X}_{n-1}^i + \boldsymbol{F}^m\big(\boldsymbol{X}_{n-1}^i,\bar{\boldsymbol{\theta}}_n^i,h,\sigma\boldsymbol{W}_n^i\big), \tag{A13}$$

where $\boldsymbol{W}_n^i = [\boldsymbol{w}_n^{i,1},\dots,\boldsymbol{w}_n^{i,M}]$, $i = 1,\dots,N$, is an $mqd_x \times M$ matrix of Gaussian perturbations (q denotes the order of the underlying RK integrator). This is done by generating $mqd_x \times M$ QMC random variates[3] and then applying the inversion method [57] to generate the Gaussian variates (one per uniform sample).

(iii) Set $\hat{p}(\boldsymbol{x}_n|\boldsymbol{y}_{1,\dots,n-1},\bar{\boldsymbol{\theta}}_n^i) = \mathcal{N}(\boldsymbol{x}_n|\hat{\boldsymbol{X}}_n^i)$.

(b) Update step.

(i) For $i = 1,\dots,N$, compute

$$\bar{\boldsymbol{M}}_n^i = \frac{1}{M}\tilde{\boldsymbol{X}}_n^i\big(\tilde{\boldsymbol{Z}}_n^i\big)^\top, \tag{A14}$$

$$\bar{\boldsymbol{S}}_n^i = \frac{1}{M}\tilde{\boldsymbol{Z}}_n^i\big(\tilde{\boldsymbol{Z}}_n^i\big)^\top + \boldsymbol{T}, \tag{A15}$$

$$\boldsymbol{K}_n^i = \bar{\boldsymbol{M}}_n^i\big(\bar{\boldsymbol{S}}_n^i\big)^{-1}, \tag{A16}$$

$$\check{\boldsymbol{X}}_n^i = \hat{\boldsymbol{X}}_n^i + \boldsymbol{K}_n^i\big(\boldsymbol{y}_n\mathbf{1}^\top - \bar{\boldsymbol{Y}}_n^i\big), \tag{A17}$$

where $\boldsymbol{T} = \sigma_0^2 \boldsymbol{I}_{d_y}$ is the measurement noise covariance, $\bar{\boldsymbol{y}}_n = \frac{1}{M}\bar{\boldsymbol{Y}}_n^i \mathbf{1}$, and $\bar{\boldsymbol{x}}_n^i = \frac{1}{M}\hat{\boldsymbol{X}}_n^i\mathbf{1}$, with $\bar{\boldsymbol{Y}}_n^i = \boldsymbol{g}(\hat{\boldsymbol{X}}_n^i,\boldsymbol{\theta}) + \boldsymbol{R}_n^i$ and $\boldsymbol{R}_n^i = [\boldsymbol{r}_n^1,\dots,\boldsymbol{r}_n^M]$ a matrix of Gaussian perturbations. Further, $\tilde{\boldsymbol{X}}_n^i$ and $\tilde{\boldsymbol{Z}}_n^i$ are calculated as

$$\tilde{\boldsymbol{X}}_n^i = \hat{\boldsymbol{X}}_n^i - \bar{\boldsymbol{x}}_n^i \mathbf{1}^\top, \tag{A18}$$

$$\tilde{\boldsymbol{Z}}_n^i = \boldsymbol{g}\big(\hat{\boldsymbol{X}}_n^i,\boldsymbol{\theta}\big) - \bar{\boldsymbol{y}}_n^i\mathbf{1}^\top. \tag{A19}$$

(ii) Compute $\hat{u}(\bar{\boldsymbol{\theta}}_n^i) = \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{g}(\bar{\boldsymbol{x}}_n^i,\bar{\boldsymbol{\theta}}_n^i),\bar{\boldsymbol{S}}_n^i)$ and obtain the normalized weights

$$w_i = \frac{\hat{u}\big(\bar{\boldsymbol{\theta}}_n^i\big)}{\sum_{j=1}^N \hat{u}_n\big(\bar{\boldsymbol{\theta}}_n^j\big)}, \quad i = 1,\dots,N. \tag{A20}$$

(iii) Set the filter approximation

$$\hat{p}\big(\boldsymbol{x}_n\big|\boldsymbol{y}_{1,\dots,n},\bar{\boldsymbol{\theta}}_n^i\big) = \mathcal{N}\big(\boldsymbol{x}_n\big|\check{\boldsymbol{X}}_n^i\big). \tag{A21}$$

(c) Generate a QMC point set $\{\boldsymbol{v}_n^i\}_{i=1}^N$ in $[0,1)^{d_\theta+1}$; let $\boldsymbol{v}_n^i = (v_n^i,\tilde{\boldsymbol{v}}_n^i) \in [0,1)\times[0,1)^{d_\theta}$.

(d) Hilbert sort. Find a permutation b such that

$$(\mathsf{h}\circ\psi)\big(\bar{\boldsymbol{\theta}}_{\mathsf{n}}^{\mathsf{b}(1)}\big) \leqslant \cdots \leqslant (\mathsf{h}\circ\psi)\big(\bar{\boldsymbol{\theta}}_{\mathsf{n}}^{\mathsf{b}(\mathsf{N})}\big) \quad \text{if } d_{\boldsymbol{\theta}} \geqslant 2,$$
$$\bar{\boldsymbol{\theta}}_n^{\mathsf{b}(1)} \leqslant \cdots \leqslant \bar{\boldsymbol{\theta}}_n^{\mathsf{b}(N)} \quad \text{if } d_{\boldsymbol{\theta}} = 1.$$

(e) Resampling. Find a permutation c such that $\boldsymbol{v}_n^{\mathsf{c}(1)} \leqslant \cdots \leqslant \boldsymbol{v}_n^{\mathsf{c}(N)}$. For $i = 1,\dots,N$ set $\boldsymbol{\theta}_n^i = \bar{\boldsymbol{\theta}}_n^j$ and $\boldsymbol{X}_n^i = \check{\boldsymbol{X}}_n^j$ if and only if

$$\sum_{k=1}^{j-1} w_n^{\mathsf{b}(k)} < v_n^{\mathsf{c}(i)} \leqslant \sum_{k=1}^{j} w_n^{\mathsf{b}(k)}, \quad j \in \{1,\dots,N\}.$$

Hence $\hat{p}(\boldsymbol{x}_n|\boldsymbol{y}_{1,\dots,n},\boldsymbol{\theta}_n^i) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{X}_n^i)$ and $\mu_n^N(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_n^i}(d\boldsymbol{\theta})$.

Algorithm 5: The SMC EKF.

1. Initialization. Draw N independent and identically distributed particles $\boldsymbol{\theta}_0^i \sim \mu_0(d\boldsymbol{\theta})$ and $\{\boldsymbol{x}_0^i\} \sim p(\boldsymbol{x}_0)$, $i = 1,\dots,N$. Let $\boldsymbol{P}_0^i = \boldsymbol{P}_0$.

2. Recursive step. At time $n-1$, we have obtained $\mu_{n-1}^N(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_{n-1}^i}(d\boldsymbol{\theta})$ and, for each $i = 1,\dots,N$, $\hat{p}(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{1,\dots,n-1},\boldsymbol{\theta}_{n-1}^i) = \mathcal{N}(\boldsymbol{x}_{n-1}|\boldsymbol{x}_{n-1}^i,\boldsymbol{P}_{n-1}^i)$.

(a) Prediction step.

(i) Draw $\bar{\boldsymbol{\theta}}_n^i \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}_{n-1}^i)$, $i = 1,\dots,N$.

(ii) For each $i = 1,\dots,N$ compute

$$\hat{\boldsymbol{x}}_n^i = \boldsymbol{x}_{n-1}^i + \boldsymbol{F}^m\big(\boldsymbol{x}_{n-1}^i,\bar{\boldsymbol{\theta}}_n^i,h\big), \tag{A22}$$

$$\hat{\boldsymbol{P}}_n^i = \boldsymbol{G}^m\big(\boldsymbol{P}_{n-1}^i,\sigma,\boldsymbol{Q}\big), \tag{A23}$$

where $\boldsymbol{Q}$ is the covariance matrix of the state and $\boldsymbol{G}^m$ denotes the composition of function $\boldsymbol{G}$ m times $(\boldsymbol{G} \circ \cdots \circ \boldsymbol{G})$. The function $\boldsymbol{G}$ in turn is described as

$$\boldsymbol{G}\big(\boldsymbol{P}^i_{k-1}, \sigma, \boldsymbol{Q}\big) = \boldsymbol{J}_{\boldsymbol{F},\hat{\boldsymbol{x}}^i_{k-1},\bar{\boldsymbol{\theta}}^i_n} \boldsymbol{P}^i_{k-1} \boldsymbol{J}^{\top}_{\boldsymbol{F},\hat{\boldsymbol{x}}^i_{k-1},\bar{\boldsymbol{\theta}}^i_n} + \sigma^2 \boldsymbol{Q}. \tag{A24}$$

(iii) Set $\hat{p}(\boldsymbol{x}_n|\boldsymbol{y}_{1,\ldots,n-1}, \bar{\boldsymbol{\theta}}^i_n) = \mathcal{N}(\boldsymbol{x}_n|\hat{\boldsymbol{x}}^i_n, \hat{\boldsymbol{P}}^i_n)$.

(b) Update step.

(i) For $i = 1, \ldots, N$, compute

$$\bar{\boldsymbol{S}}^i_n = \boldsymbol{J}_{\boldsymbol{g},\hat{\boldsymbol{x}}^i_n,\bar{\boldsymbol{\theta}}^i_n} \hat{\boldsymbol{P}}^i_n \boldsymbol{J}^{\top}_{\boldsymbol{g},\hat{\boldsymbol{x}}^i_n,\bar{\boldsymbol{\theta}}^i_n} + \sigma^2 \boldsymbol{T}, \tag{A25}$$

$$\boldsymbol{K}^i_n = \hat{\boldsymbol{P}}^i_n \boldsymbol{J}^{\top}_{\boldsymbol{g},\hat{\boldsymbol{x}}^i_n,\bar{\boldsymbol{\theta}}^i_n} \big(\bar{\boldsymbol{S}}^i_n\big)^{-1}, \tag{A26}$$

$$\check{\boldsymbol{x}}^i_n = \hat{\boldsymbol{x}}^i_n + \boldsymbol{K}^i_n\big(\boldsymbol{y}_n - \boldsymbol{g}\big(\hat{\boldsymbol{x}}^i_n, \bar{\boldsymbol{\theta}}^i_n\big)\big), \tag{A27}$$

$$\check{\boldsymbol{P}}^i_n = \big(\boldsymbol{I}_{d_x} - \boldsymbol{K}^i_n \boldsymbol{J}_{\boldsymbol{g},\hat{\boldsymbol{x}}^i_n,\bar{\boldsymbol{\theta}}^i_n}\big)\hat{\boldsymbol{P}}^i_n, \tag{A28}$$

where $\boldsymbol{T} = \sigma_0^2 \boldsymbol{I}_{d_y}$ is the measurement noise covariance.

(ii) Compute $\hat{u}(\bar{\boldsymbol{\theta}}^i_n) = \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{g}(\hat{\boldsymbol{x}}^i_n, \bar{\boldsymbol{\theta}}^i_n), \bar{\boldsymbol{S}}^i_n)$ and obtain the normalized weights

$$w_i = \frac{\hat{u}\big(\bar{\boldsymbol{\theta}}^i_n\big)}{\sum_{j=1}^N \hat{u}_n\big(\bar{\boldsymbol{\theta}}^j_n\big)}, \quad i = 1, \ldots, N. \tag{A29}$$

(iii) Set the filter approximation

$$\hat{p}\big(\boldsymbol{x}_n\big|\boldsymbol{y}_{1,\ldots,n}, \bar{\boldsymbol{\theta}}^i_n\big) = \mathcal{N}\big(\boldsymbol{x}_n\big|\check{\boldsymbol{x}}^i_n, \check{\boldsymbol{P}}^i_n\big). \tag{A30}$$

(c) Resampling. Draw indices $j_1, \ldots, j_N$ from the multinomial distribution with probabilities $w^1_n, \ldots, w^N_n$. Then set

$$\boldsymbol{\theta}^i_n = \bar{\boldsymbol{\theta}}^{j_i}_n, \quad \boldsymbol{x}^i_n = \check{\boldsymbol{x}}^{j_i}_n, \quad \boldsymbol{P}^i_n = \check{\boldsymbol{P}}^{j_i}_n \tag{A31}$$

for $i = 1, \ldots, N$. Hence

$$\hat{p}\big(\boldsymbol{x}_n\big|\boldsymbol{y}_{1,\ldots,n}, \boldsymbol{\theta}^i_n\big) = \mathcal{N}\big(\boldsymbol{x}_n\big|\boldsymbol{x}^i_n, \boldsymbol{P}^i_n\big),$$

$$\mu^N_n(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}^i_n}(d\boldsymbol{\theta}).$$

Algorithm 6: The SQMC EKF.

1. Initialization. Generate QMC uniform samples $\{\boldsymbol{v}^i_{-1}, \boldsymbol{v}^i_0\}$ in $[0, 1)^{d_{\boldsymbol{\theta}}}$ and $\bar{\boldsymbol{v}}^i_{-1}$ in $[0, 1)^{d_x}$. Draw $\boldsymbol{\theta}^i_0 \sim \mu_0(d\boldsymbol{\theta}|\boldsymbol{v}^i_{-1})$ and $\{\boldsymbol{x}^i_0\} \sim p(\boldsymbol{x}_0|\bar{\boldsymbol{v}}^i_{-1})$, $i = 1, \ldots, N$. Let $\boldsymbol{P}^i_0 = \boldsymbol{P}_0$.

2. Recursive step $n \geqslant 1$. At time $n-1$, we have obtained $\mu^N_{n-1}(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}^i_{n-1}}(d\boldsymbol{\theta})$ and, for each $i = 1, \ldots, N$, $\hat{p}(\boldsymbol{x}_{n-1}|\boldsymbol{y}_{1,\ldots,n-1}, \boldsymbol{\theta}^i_{n-1}) = \mathcal{N}(\boldsymbol{x}_{n-1}|\boldsymbol{x}^i_{n-1}, \boldsymbol{P}^i_{n-1})$.

(a) Prediction step.

(i) If $n = 1$, draw $\bar{\boldsymbol{\theta}}^i_n \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}^i_{n-1}, \boldsymbol{v}^i_0)$; otherwise draw $\bar{\boldsymbol{\theta}}^i_n \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}^i_{n-1}, \tilde{\boldsymbol{v}}^{\mathsf{c}(i)}_{n-1})$, $i=1, \ldots, N$, for $n \geqslant 2$.

(ii) For each $i = 1, \ldots, N$ compute

$$\hat{\boldsymbol{x}}^i_n = \boldsymbol{x}^i_{n-1} + \boldsymbol{F}^m\big(\boldsymbol{x}^i_{n-1}, \bar{\boldsymbol{\theta}}^i_n, h\big), \tag{A32}$$

$$\hat{\boldsymbol{P}}^i_n = \boldsymbol{G}^m\big(\boldsymbol{P}^i_{n-1}, \sigma, \boldsymbol{Q}\big), \tag{A33}$$

where $\boldsymbol{Q}$ is the covariance matrix of the state and $\boldsymbol{G}^m$ denotes the composition of function $\boldsymbol{G}$ m times $(\boldsymbol{G} \circ \cdots \circ \boldsymbol{G})$. The function $\boldsymbol{G}$ in turn is described as

$$\boldsymbol{G}\big(\boldsymbol{P}^i_{k-1}, \sigma, \boldsymbol{Q}\big) = \boldsymbol{J}_{\boldsymbol{F},\hat{\boldsymbol{x}}^i_{k-1},\bar{\boldsymbol{\theta}}^i_n} \boldsymbol{P}^i_{k-1} \boldsymbol{J}^{\top}_{\boldsymbol{F},\hat{\boldsymbol{x}}^i_{k-1},\bar{\boldsymbol{\theta}}^i_n} + \sigma^2 \boldsymbol{Q}. \tag{A34}$$

(iii) Set $\hat{p}(\boldsymbol{x}_n|\boldsymbol{y}_{1,\ldots,n-1}, \bar{\boldsymbol{\theta}}^i_n) = \mathcal{N}(\boldsymbol{x}_n|\hat{\boldsymbol{x}}^i_n, \hat{\boldsymbol{P}}^i_n)$.

(b) Update step.

(i) For $i = 1, \ldots, N$, compute

$$\bar{\boldsymbol{S}}^i_n = \boldsymbol{J}_{\boldsymbol{g},\hat{\boldsymbol{x}}^i_n,\bar{\boldsymbol{\theta}}^i_n} \hat{\boldsymbol{P}}^i_n \boldsymbol{J}^{\top}_{\boldsymbol{g},\hat{\boldsymbol{x}}^i_n,\bar{\boldsymbol{\theta}}^i_n} + \sigma^2 \boldsymbol{T}, \tag{A35}$$

$$\boldsymbol{K}^i_n = \hat{\boldsymbol{P}}^i_n \boldsymbol{J}^{\top}_{\boldsymbol{g},\hat{\boldsymbol{x}}^i_n,\bar{\boldsymbol{\theta}}^i_n} \big(\bar{\boldsymbol{S}}^i_n\big)^{-1}, \tag{A36}$$

$$\check{\boldsymbol{x}}^i_n = \hat{\boldsymbol{x}}^i_n + \boldsymbol{K}^i_n\big(\boldsymbol{y}_n - \boldsymbol{g}\big(\hat{\boldsymbol{x}}^i_n, \bar{\boldsymbol{\theta}}^i_n\big)\big), \tag{A37}$$

$$\check{\boldsymbol{P}}^i_n = \big(\boldsymbol{I}_{d_x} - \boldsymbol{K}^i_n \boldsymbol{J}_{\boldsymbol{g},\hat{\boldsymbol{x}}^i_n,\bar{\boldsymbol{\theta}}^i_n}\big)\hat{\boldsymbol{P}}^i_n, \tag{A38}$$

where $\boldsymbol{T} = \sigma_0^2 \boldsymbol{I}_{d_y}$ is the measurement noise covariance.

(ii) Compute $\hat{u}(\bar{\boldsymbol{\theta}}^i_n) = \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{g}(\hat{\boldsymbol{x}}^i_n, \bar{\boldsymbol{\theta}}^i_n), \bar{\boldsymbol{S}}^i_n)$ and obtain the normalized weights

$$w_i = \frac{\hat{u}\big(\bar{\boldsymbol{\theta}}^i_n\big)}{\sum_{j=1}^N \hat{u}_n\big(\bar{\boldsymbol{\theta}}^j_n\big)}, \quad i = 1, \ldots, N. \tag{A39}$$

(iii) Set the filter approximation

$$\hat{p}\big(\boldsymbol{x}_n\big|\boldsymbol{y}_{1,\ldots,n}, \bar{\boldsymbol{\theta}}^i_n\big) = \mathcal{N}\big(\boldsymbol{x}_n\big|\check{\boldsymbol{x}}^i_n, \check{\boldsymbol{P}}^i_n\big). \tag{A40}$$

(c) Generate a QMC point set $\{\boldsymbol{v}^i_n\}_{i=1}^N$ in $[0, 1)^{d_{\boldsymbol{\theta}}+1}$. Let $\boldsymbol{v}^i_n = (v^i_n, \tilde{\boldsymbol{v}}^i_n) \in [0, 1) \times [0, 1)^{d_{\boldsymbol{\theta}}}$.

(d) Hilbert sort. Find a permutation b such that

$$(\mathsf{h} \circ \psi)\big(\bar{\boldsymbol{\theta}}^{\mathsf{b}(1)}_{\mathsf{n}}\big) \leqslant \cdots \leqslant (\mathsf{h} \circ \psi)\big(\bar{\boldsymbol{\theta}}^{\mathsf{b}(\mathsf{N})}_{\mathsf{n}}\big) \quad \text{if } d_{\boldsymbol{\theta}} \geqslant 2,$$
$$\bar{\boldsymbol{\theta}}^{\mathsf{b}(1)}_n \leqslant \cdots \leqslant \bar{\boldsymbol{\theta}}^{\mathsf{b}(N)}_n \quad \text{if } d_{\boldsymbol{\theta}} = 1.$$

(e) Resampling. Find a permutation c such that $\boldsymbol{v}^{\mathsf{c}(1)}_n \leqslant \cdots \leqslant \boldsymbol{v}^{\mathsf{c}(N)}_n$. For $i = 1, \ldots, N$ set $\boldsymbol{\theta}^i_n = \bar{\boldsymbol{\theta}}^j_n$, $\boldsymbol{x}^i_n = \check{\boldsymbol{x}}^j_n$ and $\boldsymbol{P}^i_n = \check{\boldsymbol{P}}^j_n$ if and only if

$$\sum_{k=1}^{j-1} w^{\mathsf{b}(k)}_n < v^{\mathsf{c}(i)}_n \leqslant \sum_{k=1}^{j} w^{\mathsf{b}(k)}_n, \quad j \in \{1, \ldots, N\}.$$

Hence $\hat{p}(\boldsymbol{x}_n|\boldsymbol{y}_{1,\ldots,n}, \boldsymbol{\theta}^i_n) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{x}^i_n, \boldsymbol{P}^i_n)$ and $\mu^N_n(d\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}^i_n}(d\boldsymbol{\theta})$.

The computational complexity of sampling in the parameter space increases linearly with N using either the SMC or the SQMC methods. However, the complexity of SQMC increases also with $N \log N$ due to the resampling step since it requires the computation of the Hilbert inverse as well as two permutations in steps 2(d) and 2(e), in Algorithms 4 and 6. The computational cost of the EKF method increases with rates d_y^3 and d_x^2. The former is due to the inversion of the d_y-dimensional matrix $\bar{\boldsymbol{S}}^i_n$ in Eqs. (A26) and (A36) of Algorithms 5 and 6, respectively, while the product in Eqs. (A28) and (A38), used to obtain the predictive state covariance matrix $\check{\boldsymbol{P}}^i_n$, justifies the latter. Using the EnKF method, the complexity is also cubic with d_y because of the

TABLE II. Summary of the complexity of the different NHFs.

Algorithm	SMC	SQMC
EKF	$O(d_x^2 d_y^3 N)$	$O(d_x^2 d_y^3 N \log N)$
EnKF	$O(d_x d_y^3 M N)$	$O(d_x d_y^3 M N \log N)$

inversion of $\bar{S}_n^i$ in Eqs. (A6) and (A16) of Algorithms 3 and 4 and it increases linearly with M and d_x because Algorithms 3 and 4 do not require the computation of the predictive state covariance matrices. In Table II the complexity is summarized depending on the filters we choose. In order to alleviate the computational cost of the inversion of the observation covariance matrix $\bar{S}_n^i$, in practice we use the approximation described in Appendix C.

APPENDIX B: PROOF OF THEOREM 1

1. Outline of the proof

We need to prove that the approximation μ_n^N generated by a generic nested filter that satisfies Assumptions 1–3 converges to $\bar{\mu}_n$ in L_p for each $n = 1, 2, \ldots, n_0 < \infty$. We split the analysis of the nested filter into three steps: jittering, weight computation, and resampling. The approximation μ_{n-1}^N of $\bar{\mu}_{n-1}$ is available at the beginning of the nth time step. After jittering, we obtain a new approximation

$$\check{\mu}_{n-1}^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\bar{\boldsymbol{\theta}}_n^i} \tag{B1}$$

that can be proved to converge to $\bar{\mu}_{n-1}$ using an auxiliary result from [21]. After the computation of the weights, the measure

$$\tilde{\mu}_n^N = \sum_{i=1}^{N} w_n^i \delta_{\bar{\boldsymbol{\theta}}_n^i} \tag{B2}$$

is obtained and its convergence towards $\bar{\mu}_n$ has to be established. Finally, after the resampling step, a standard piece of analysis proves the convergence of

$$\mu_n^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{\theta}_n^i} \tag{B3}$$

to $\bar{\mu}_n$. Below, we provide three lemmas for the conditional convergence of $\check{\mu}_{n-1}^N$, $\tilde{\mu}_n^N$, and μ_n^N, respectively. Then we combine them in order to prove Theorem 1 by an induction argument.

2. Jittering

In the jittering step, a new cloud of particles $\{\bar{\boldsymbol{\theta}}_n^i\}_{i=1}^N$ is generated by propagating the existing samples across the kernels $\kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}_{n-1}^i)$, $i = 1, \ldots, N$. This step has been analyzed in [21] in the context of the NPF. Several types of kernels can be used. In general, there is a trade-off between the number of particles that are changed using this kernel and the "amount of perturbation" that can be applied to each particle. For this reason, we let the jittering kernel κ_N depend explicitly on N. For our analysis, Assumption 3 is sufficient.

The convergence results to be given in this appendix are presented in terms of upper bounds for the L_p-norms of the approximation errors. For a random variable z, its L_p-norm is $\|z\|_p = \mathbb{E}[|z|^p]^{1/p}$. The approximate measures generated by the nested filter, e.g., μ_n^N, are measured-valued random variables. Therefore, integrals of the form (h, μ_n^N), for some $h \in B(D)$, are real random variables and it makes sense to evaluate the L_p-norm of the random error $(h, \mu_n^N) - (h, \bar{\mu}_n)$. We start with the approximation $\check{\mu}_{n-1}^N$ produced after the jittering step at time n.

Lemma 1. Let the sequence of observations $y_{1,\ldots,n}$ be arbitrary but fixed. If $h \in B(D)$, Assumption 3 holds, and

$$\left\|\left(h, \mu_{n-1}^N\right) - (h, \bar{\mu}_{n-1})\right\|_p \leqslant \frac{c_{n-1}\|h\|_\infty}{\sqrt{N}} \tag{B4}$$

for some $p \geqslant 1$ and a constant $c_{n-1} < \infty$ independent of N, then

$$\left\|\left(h, \check{\mu}_{n-1}^N\right) - (h, \bar{\mu}_{n-1})\right\|_p \leqslant \frac{c_{1,n}\|h\|_\infty}{\sqrt{N}}, \tag{B5}$$

where the constant $c_{1,n} < \infty$ is also independent of N.

The proof of this Lemma 1 is identical to the proof of Lemma 3 in [21].

3. Computation of the weights

In order to analyze the errors at the weight computation step we rely on Assumption 2. An upper bound for the error in the weight computation step is established next.

Lemma 2. Let the sequence of observations $y_{1,\ldots,n}$ be arbitrary but fixed. Choose any $h \in B(D)$ and some $p \geqslant 1$. If Assumptions 1 and 2 hold and

$$\left\|\left(h, \check{\mu}_{n-1}^N\right) - (h, \bar{\mu}_{n-1})\right\|_p \leqslant \frac{c_{1,n}\|h\|_\infty}{\sqrt{N}} \tag{B6}$$

for some constant $c_{1,n} < \infty$ independent of N, then

$$\left\|\left(h, \tilde{\mu}_n^N\right) - (h, \bar{\mu}_n)\right\|_p \leqslant \frac{c_{2,n}\|h\|_\infty}{\sqrt{N}}, \tag{B7}$$

where the constant $c_{2,n} < \infty$ is independent of N.

Proof. We address the characterization of the weights and therefore of the approximate measure $\tilde{\mu}_n^N = \sum_{i=1}^N w_n^i \delta_{\bar{\theta}_n^i}$. From the definition of the projective product in (29), the integrals of h with respect to $\bar{\mu}_n$ and $\tilde{\mu}_n^N$ can be written as

$$(h, \bar{\mu}_n) = \frac{(\bar{u}_n h, \bar{\mu}_{n-1})}{(\bar{u}_n, \bar{\mu}_{n-1})}, \quad \left(h, \tilde{\mu}_n^N\right) = \frac{\left(\hat{u}_n h, \check{\mu}_{n-1}^N\right)}{\left(\hat{u}_n, \check{\mu}_{n-1}^N\right)}, \tag{B8}$$

respectively. From (B8) we can write the difference $(h, \tilde{\mu}_n^N) - (h, \bar{\mu}_n)$ as

$$\left(h, \tilde{\mu}_n^N\right) - (h, \bar{\mu}_n) = \frac{\left(h\hat{u}_n, \check{\mu}_{n-1}^N\right) - (h\bar{u}_n, \bar{\mu}_{n-1})}{(\bar{u}_n, \bar{\mu}_{n-1})} + \left(h, \tilde{\mu}_n^N\right)\frac{(\bar{u}_n, \bar{\mu}_{n-1}) - \left(\hat{u}_n, \check{\mu}_{n-1}^N\right)}{(\bar{u}_n, \bar{\mu}_{n-1})},$$

which readily yields the inequality

$$\begin{aligned}\left|\left(h,\tilde{\mu}_n^N\right)-(h,\bar{\mu}_{n-1})\right| &\leqslant \frac{\left|\left(h\hat{u}_n,\check{\mu}_{n-1}^N\right)-(h\bar{u}_n,\bar{\mu}_{n-1})\right|}{(\bar{u}_n,\bar{\mu}_{n-1})}\\ &\quad+\frac{\|h\|_\infty\left|\left(\hat{u}_n,\check{\mu}_{n-1}^N\right)-(\bar{u}_n,\bar{\mu}_{n-1})\right|}{(\bar{u}_n,\bar{\mu}_{n-1})}\end{aligned} \qquad \text{(B9)}$$

by simply noting that $|(h,\tilde{\mu}_n^N)|\leqslant\|h\|_\infty$, since $\tilde{\mu}_n^N$ is a probability measure. From (B9) and Minkowski's inequality we easily obtain the bound

$$\begin{aligned}\left\|\left(h,\tilde{\mu}_n^N\right)-(h,\bar{\mu}_{n-1})\right\|_p &\leqslant \frac{1}{(\bar{u}_n,\bar{\mu}_{n-1})}\left[\|h\|_\infty\left\|\left(\hat{u}_n,\check{\mu}_{n-1}^N\right)-(\bar{u}_n,\bar{\mu}_{n-1})\right\|_p\right.\\ &\quad\left.+\left\|\left(h\hat{u}_n,\check{\mu}_{n-1}^N\right)-(h\bar{u}_n,\bar{\mu}_{n-1})\|_p\right],\right.\end{aligned} \qquad \text{(B10)}$$

where $(\bar{u}_n,\bar{\mu}_{n-1})>0$ from Assumption 2(b).

We need to find upper bounds for the two terms on the right-hand side of (B10). Consider first the term $\|(h\hat{u}_n,\check{\mu}_{n-1}^N)-(h\bar{u}_n,\bar{\mu}_{n-1})\|_p$. A simple triangle inequality yields

$$\begin{aligned}\left\|\left(h\hat{u}_n,\check{\mu}_{n-1}^N\right)-(h\bar{u}_n,\bar{\mu}_{n-1})\right\|_p &\leqslant \left\|\left(h\hat{u}_n,\check{\mu}_{n-1}^N\right)-\left(h\bar{u}_n,\check{\mu}_{n-1}^N\right)\right\|_p+\left\|\left(h\bar{u}_n,\check{\mu}_{n-1}^N\right)\right.\\ &\quad\left.-(h\bar{u}_t,\bar{\mu}_{n-1})\right\|_p.\end{aligned} \qquad \text{(B11)}$$

On one hand, since $\sup_{\boldsymbol{\theta}\in D}|h(\boldsymbol{\theta})\bar{u}_n(\boldsymbol{\theta})|\leqslant\|h\|_\infty\|\bar{u}_n\|_\infty<\infty$ [see Assumption 2(a)], it follows from the assumption in Eq. (B6) that

$$\left\|\left(h\bar{u}_n,\check{\mu}_{n-1}^N\right)-(h\bar{u}_n,\bar{\mu}_{n-1})\right\|_p\leqslant\frac{c_{1,n}\|h\|_\infty\|\bar{u}_n\|_\infty}{\sqrt{N}}, \qquad \text{(B12)}$$

where $c_{1,n}<\infty$ is a constant independent of N.

On the other hand, we may note that

$$\begin{aligned}\left|\left(h\hat{u}_n,\check{\mu}_{n-1}^N\right)-\left(h\bar{u}_n,\check{\mu}_{n-1}^N\right)\right|^p &= \left|\frac{1}{N}\sum_{i=1}^N\left[h\left(\bar{\boldsymbol{\theta}}_n^i\right)\hat{u}_n\left(\bar{\boldsymbol{\theta}}_n^i\right)-h\left(\bar{\boldsymbol{\theta}}_n^i\right)\bar{u}_n\left(\bar{\boldsymbol{\theta}}_n^i\right)\right]\right|^p.\end{aligned} \qquad \text{(B13)}$$

Let $\mathcal{G}_n$ be the σ algebra generated by the random particles $\{\bar{\boldsymbol{\theta}}^i_{1,\ldots,n-1},\boldsymbol{\theta}^i_{0,\ldots,n-1}\}_{1\leqslant i\leqslant N}$ and assume that p is even. Then we can apply conditional expectations on both sides of (B13) to obtain

$$\begin{aligned}\mathbb{E}\left[\left|\left(h\hat{u}_n,\check{\mu}_{n-1}^N\right)-\left(h\bar{u}_n,\check{\mu}_{n-1}^N\right)\right|^p\middle|\mathcal{G}_n\right] &= \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^N h\left(\bar{\boldsymbol{\theta}}_n^i\right)m_n\left(\bar{\boldsymbol{\theta}}_n^i\right)\right)^p\middle|\mathcal{G}_n\right],\end{aligned}$$

where the expression on the right-hand side has been simplified by using the assumption $\hat{u}_n(\boldsymbol{\theta})=\bar{u}_n(\boldsymbol{\theta})+m_n(\boldsymbol{\theta})$ in Assumption 1. Also from Assumption 1, the random variables $m_n(\bar{\boldsymbol{\theta}}_n^i)$ are conditionally independent (given $\mathcal{G}_n$) and have zero mean and finite moments of order p, $\mathbb{E}[m_n(\bar{\boldsymbol{\theta}}_n^i)^p]\leqslant\sigma^p<\infty$. If we realize that

$$\mathbb{E}\left[h\left(\bar{\boldsymbol{\theta}}_n^i\right)m_n\left(\bar{\boldsymbol{\theta}}_n^i\right)\middle|\mathcal{G}_n\right]=h\left(\bar{\boldsymbol{\theta}}_n^i\right)\mathbb{E}\left[m_n\left(\bar{\boldsymbol{\theta}}_n^i\right)\middle|\mathcal{G}_n\right]=0$$

and bear in mind the conditional independence of the $m_n(\bar{\boldsymbol{\theta}}_n^i)$, then it is an exercise in combinatorics to show that the number of nonzero terms in

$$\begin{aligned}&\mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^N h(\bar{\boldsymbol{\theta}}_n^i)m_n(\bar{\boldsymbol{\theta}}_n^i)\right)^p\middle|\mathcal{G}_n\right]\\ &\quad=\sum_{i_1}\cdots\sum_{i_p}\mathbb{E}\left[h\left(\bar{\boldsymbol{\theta}}_n^{i_1}\right)m_n\left(\bar{\boldsymbol{\theta}}_n^{i_1}\right)\cdots h\left(\bar{\boldsymbol{\theta}}_n^{i_p}\right)m_n\left(\bar{\boldsymbol{\theta}}_n^{i_p}\right)\middle|\mathcal{G}_n\right]\end{aligned}$$

is at most $\tilde{c}^pN^{p/2}$ for some constant $\tilde{c}^p<\infty$ independent of N and h. Since each of the nonzero terms is upper bounded by $\mathbb{E}[[h(\bar{\boldsymbol{\theta}}_n^i)m_n(\bar{\boldsymbol{\theta}}_n^i)]^p|\mathcal{G}_n]\leqslant\|h\|_\infty^p\sigma^p<\infty$ (using Assumption 1 again), then it follows that

$$\begin{aligned}&\mathbb{E}\left[\left|\left(h\hat{u}_n,\check{\mu}_{n-1}^N\right)-\left(h\bar{u}_n,\check{\mu}_{n-1}^N\right)\right|^p\right]\\ &\quad=\mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^N h(\bar{\boldsymbol{\theta}}_n^i)m_n(\bar{\boldsymbol{\theta}}_n^i)\right)^p\middle|\mathcal{G}_n\right]\leqslant\frac{\tilde{c}^p\sigma^p\|h\|_\infty^p}{N^{p/2}}\end{aligned} \qquad \text{(B14)}$$

for even p. Given (B14), it is straightforward to show that the same result holds for every $p\geqslant 1$ using Jensen's inequality. Finally, since the bound on the right-hand side of (B14) is independent of $\mathcal{G}_n$, we can take expectations on both sides of the inequality and obtain that

$$\left\|\left(h\hat{u}_n,\check{\mu}_{n-1}^N\right)-\left(h\bar{u}_n,\check{\mu}_{n-1}^N\right)\right\|_p\leqslant\frac{\tilde{c}\sigma\|h\|_\infty}{\sqrt{N}}. \qquad \text{(B15)}$$

Substituting (B15) and (B12) into (B11) yields

$$\left\|\left(h\hat{u}_n,\check{\mu}_{n-1}^N\right)-(h\bar{u}_n,\bar{\mu}_{n-1})\right\|_p\leqslant\frac{c_n'\|h\|_\infty^p\|\bar{u}_n\|_\infty}{\sqrt{N}}, \qquad \text{(B16)}$$

where $c_n'=c_{1,n}+\tilde{c}\sigma$ is a constant independent of N. The same argument leading to the bound in (B16) can be repeated, step by step, on the norm $\|(\hat{u}_n,\check{\mu}_{n-1}^N)-(\bar{u}_n,\bar{\mu}_{n-1})\|_p$ [simply taking $h(\boldsymbol{\theta})=1$], to arrive at

$$\left\|\left(\hat{u}_n,\check{\mu}_{n-1}^N\right)-(\bar{u}_n,\bar{\mu}_{n-1})\right\|_p\leqslant\frac{c_n'\|\bar{u}_n\|_\infty}{\sqrt{N}}. \qquad \text{(B17)}$$

To complete the proof, we substitute (B16) and (B17) back into (B10) and so obtain

$$\left\|\left(h,\tilde{\mu}_n^N\right)-(h,\bar{\mu}_{n-1})\|_p\leqslant\frac{c_{2,n}\|h\|_\infty}{\sqrt{N}},\right.$$

where the constant $c_{2,n}=\|\bar{u}_n\|_\infty(2c_n')/(\bar{u}_t,\bar{\mu}_{t-1})<\infty$ is independent of N. ■

4. Resampling

The quantification of the error in the resampling step of the nested filter is a standard piece of analysis, well known from the particle filtering literature (see, e.g., [61]). We can state the following result.

Lemma 3. Let the sequence of observations $y_{1,\ldots,n}$ be arbitrary but fixed. If $h\in B(D)$ and

$$\left\|\left(h,\tilde{\mu}_n^N\right)-(h,\bar{\mu}_n)\right\|_p\leqslant\frac{c_{2,n}\|h\|_\infty}{\sqrt{N}} \qquad \text{(B18)}$$

for a constant $c_{2,n}<\infty$ independent of N, then

$$\left\|\left(h,\mu_n^N\right)-(h,\bar{\mu}_n)\right\|_p\leqslant\frac{c_{3,n}\|h\|_\infty}{\sqrt{N}},$$

where the constant $c_{3,n} < \infty$ is independent of N as well.

See, e.g., the proof of Lemma 1 in [64].

5. An induction proof for Theorem 1

Finally, we can put Lemmas 1–3 together in order to prove the inequality (31) by induction in n. At time $n = 0$, we draw $\boldsymbol{\theta}_0^i, i = 1, \dots, N$, independently from the prior $\mu_0 = \bar{\mu}_0$ and it is straightforward to show that $\|(h, \mu_0^N) - (h, \bar{\mu}_0)\|_p \leqslant \frac{c_0\|h\|_\infty}{\sqrt{N}}$, where c_0 does not depend on N.

Assume that, at time $n-1$,

$$\left\|\left(h, \mu_{n-1}^N\right) - (h, \bar{\mu}_{n-1})\right\|_p \leqslant \frac{c_{n-1}\|h\|_\infty}{\sqrt{N}},$$

where $c_{n-1} < \infty$ is independent of N. Then we simply apply Lemmas 1–3 in sequence to obtain

$$\left\|\left(h, \mu_n^N\right) - (h, \bar{\mu}_n)\right\|_p \leqslant \frac{c_n\|h\|_\infty}{\sqrt{N}}$$

for a constant $c_n = c_{3,n} < \infty$ independent of N. ■

APPENDIX C: SIMPLIFICATION OF THE INVERSE $(S^i)^{-1}$

The predictive covariance of the observation vector $\boldsymbol{y}_n$ is a $d_y \times d_y$ matrix $\boldsymbol{S}_n$. Inverting $\boldsymbol{S}_n$ has a cost $O(d_y^3)$, which can become intractable. Assuming that variables located "far away" in the circumference of the Lorenz 96 model have small correlation, we can approximate $\boldsymbol{S}_n$ as a block diagonal matrix, namely, $\hat{\boldsymbol{S}}_n = \boldsymbol{S}_n \odot \boldsymbol{M}$, where $\odot$ denotes an elementwise product,

$$\boldsymbol{M} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \end{bmatrix} \tag{C1}$$

is a mask matrix, and $\mathbf{0}$ and $\mathbf{1}$ are, respectively, matrices of zeros and ones of dimension $d_q \times d_q$. There are Q blocks in the diagonal of $\boldsymbol{M}$, hence $d_y = Qd_q$. The original matrix could contain some nonzero values where the zero blocks of $\boldsymbol{M}$ are placed, however their values are assumed close to zero. The resulting matrix

$$\hat{\boldsymbol{S}} = \begin{bmatrix} \bar{\boldsymbol{S}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{S}}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\boldsymbol{S}}_Q \end{bmatrix}$$

is easily inverted as

$$\hat{\boldsymbol{S}}_n^{-1} = \begin{bmatrix} \bar{\boldsymbol{S}}_1^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{S}}_2^{-1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\boldsymbol{S}}_Q^{-1} \end{bmatrix}$$

with a computational cost $O(Qd_q^3) = O(\frac{d_y^3}{Q^2})$.

[1] A. M. Clayton, A. Lorenc, and D. M. Barker, Q. J. R. Meteorol. Soc. **139**, 1445 (2013).

[2] P. J. V. Leeuwen, Mon. Weather Rev. **131**, 2071 (2003).

[3] D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, and P. Bauer, Q. J. R. Meteorol. Soc. **137**, 553 (2011).

[4] A. Golightly and D. J. Wilkinson, Interface Focus **1**, 807 (2011).

[5] J. Liu and M. West, in *Sequential Monte Carlo Methods in Practice*, edited by A. Doucet, N. de Freitas, and N. Gordon (Springer, Berlin, 2001), Chap. 10, pp. 197–223.

[6] A. Aksoy, F. Zhang, and J. W. Nielsen-Gammon, Mon. Weather Rev. **134**, 2951 (2006).

[7] G. Evensen, IEEE Control Syst. **29**, 83 (2009).

[8] C. M. Carvalho, M. S. Johannes, H. F. Lopes, and N. G. Polson, Stat. Sci. **25**, 88 (2010).

[9] N. Kantas, A. Doucet, S. S. Singh, J. M. Maciejowski, and N. Chopin, Stat. Sci. **30**, 328 (2015).

[10] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić, Proc. IEEE **92**, 423 (2004).

[11] G. Kitagawa, J. Am. Stat. Assoc. **93**, 1203 (1998).

[12] H. Zhang, H.-J. Hendricks-Franssen, X. Han, J. A. Vrugt, and H. Vereecken, Hydrol. Earth Syst. Sci. **21**, 4927 (2017).

[13] G. Storvik, IEEE Trans. Signal Process. **50**, 281 (2002).

[14] P. M. Djurić and J. Míguez, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, Piscataway, 2002), Vol. 3.

[15] P. M. Djurić, M. F. Bugallo, and J. Míguez, *Proceedings of the 29th IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, Piscataway, 2004).

[16] A. Papavasiliou, J. Appl. Probab. **42**, 1053 (2005).

[17] A. Papavasiliou, Stoch. Process. Appl. **116**, 1048 (2006).

[18] J. Olsson, O. Cappé, R. Douc, and E. Moulines, Bernoulli **14**, 155 (2008).

[19] C. Andrieu, A. Doucet, and R. Holenstein, J. R. Stat. Soc. B **72**, 269 (2010).

[20] N. Chopin, P. E. Jacob, and O. Papaspiliopoulos, J. R. Stat. Soc. B **75**, 397 (2013).

[21] D. Crisan and J. Míguez, Bernoulli **24**, 3039 (2018).

[22] I. P. Mariño, A. Zaikin, and J. Miguez, PLoS One **12**, e0182015 (2017).

[23] J. Míguez, I. P. Mariño, and M. A. Vázquez, Signal Process. **142**, 281 (2018).

[24] C. Andrieu, A. Doucet, and V. B. Tadić, IFAC Proc. Vol. **45**, 500 (2012).

[25] V. B. Tadic, IEEE Trans. Inf. Theory **56**, 6406 (2010).

[26] N. Santitissadeekorn and C. Jones, Mon. Weather Rev. **143**, 2028 (2015).

[27] M. Frei and H. R. Künsch, Mon. Weather Rev. **140**, 1476 (2012).

[28] T. T. Ashley and S. B. Andersson, Phys. Rev. E **92**, 052707 (2015).

[29] J. Ye, D. Rey, N. Kadakia, M. Eldridge, U. I. Morone, P. Rozdeba, H. D. I. Abarbanel, and J. C. Quinn, Phys. Rev. E **92**, 052901 (2015).

[30] S.-i. Ito, H. Nagao, A. Yamanaka, Y. Tsukada, T. Koyama, M. Kano, and J. Inoue, Phys. Rev. E **94**, 043307 (2016).

[31] M. D. Vrettas, M. Opper, and D. Cornford, Phys. Rev. E **91**, 012148 (2015).

[32] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, In IEE Proc. F-radar and signal processing **140**, 107 (1993).

[33] A. Doucet, S. Godsill, and C. Andrieu, Stat. Comput. **10**, 197 (2000).

[34] M. Gerber and N. Chopin, J. R. Stat. Soc. B Met. **77**, 509 (2015).

[35] J. L. Anderson, Tellus A **61**, 72 (2009).

[36] H. Li, E. Kalnay, and T. Miyoshi, Q. J. R. Meteorol. Soc. **135**, 523 (2009).

[37] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. A. Yorke, Tellus A **56**, 415 (2004).

[38] P. L. Houtekamer and H. L. Mitchell, Mon. Weather Rev. **129**, 123 (2001).

[39] F. C. Vossepoel and P. Jan van Leeuwen, Mon. Weather Rev. **135**, 1006 (2007).

[40] H. M. Arnold, Stochastic parametrisation and model uncertainty, Ph.D. thesis, University of Oxford, 2013.

[41] P. J. V. Leeuwen, Q. J. R. Meteorol. Soc. **136**, 1991 (2010).

[42] D. Pazó, A. Carrassi, and J. M. López, Q. J. R. Meteorol. Soc. **142**, 1290 (2016).

[43] J. Hakkarainen, A. Ilin, A. Solonen, M. Laine, H. Haario, J. Tamminen, E. Oja, and H. Järvinen, Nonlinear Proc. Geophys. **19**, 127 (2012).

[44] D. Rey, M. Eldridge, M. Kostuk, H. D. Abarbanel, J. Schumann-Bischoff, and U. Parlitz, Phys. Lett. A **378**, 869 (2014).

[45] T. C. Gard, *Introduction to Stochastic Differential Equations* (Dekker, New York, 1988).

[46] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods* (Springer, Berlin, 2004).

[47] E. Koblents and J. Míguez, Stat. Comput. **25**, 407 (2015).

[48] J. S. Liu and R. Chen, J. Am. Stat. Assoc. **93**, 1032 (1998).

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

[50] D. Crisan and J. Miguez, Adv. Appl. Probab. **49**, 1170 (2017).

[51] J. T. Ambadan and Y. Tang, J. Atmos. Sci. **66**, 261 (2009).

[52] I. Arasaratnam and S. Haykin, IEEE Trans. Autom. Control **54**, 1254 (2009).

[53] S. J. Julier and J. Uhlmann, Proc. IEEE **92**, 401 (2004).

[54] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods* (SIAM, Philadelphia, 1992), Vol. 63.

[55] J. H. Halton, Commun. ACM **7**, 701 (1964).

[56] P. Bratley and B. L. Fox, ACM Trans. Math. Software **14**, 88 (1988).

[57] L. Martino, D. Luengo, and J. Míguez, *Independent Random Sampling Methods* (Springer, Berlin, 2018).

[58] G. E. P. Box and M. E. Muller, Ann. Math. Stat. **29**, 610 (1958).

[59] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, IEEE Trans. Knowl. Data Eng. **13**, 124 (2001).

[60] C. H. Hamilton and A. Rau-Chaplin, Inf. Process. Lett. **105**, 155 (2008).

[61] A. Bain and D. Crisan, *Fundamentals of Stochastic Filtering* (Springer, Berlin, 2008).

[62] Y. Hirata, Phys. Rev. E **89**, 052916 (2014).

[63] G. Evensen, Ocean Dyn. **53**, 343 (2003).

[64] J. Míguez, D. Crisan, and P. M. Djurić, Stat. Comput. **23**, 91 (2013).