Signal Processing 189 (2021) 108295

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Nested Gaussian filters for recursive Bayesian inference and nonlinear tracking in state space models

Sara Pérez-Vieites, Joaquín Míguez*

Department of Signal Theory & Communications, Universidad Carlos III de Madrid. Avenida de la Universidad 30, 28911 Leganés, Madrid, Spain

ARTICLE INFO

Article history:
Received 23 December 2020
Revised 29 June 2021
Accepted 17 August 2021
Available online 19 August 2021

Keywords:
Filtering
Kalman
Monte Carlo
Bayesian inference
Parameter estimation

ABSTRACT

We introduce a new sequential methodology to calibrate the fixed parameters and track the stochastic dynamical variables of a state-space system. The proposed method is based on the nested hybrid filtering (NHF) framework of [1], that combines two layers of filters, one inside the other, to compute the joint posterior probability distribution of the static parameters and the state variables. In particular, we explore the use of deterministic sampling techniques for Gaussian approximation in the first layer of the algorithm, instead of the Monte Carlo methods employed in the original procedure. The resulting scheme reduces the computational cost and so makes the algorithms potentially better-suited for high-dimensional state and parameter spaces. We describe a specific instance of the new method and then study its performance and efficiency of the resulting algorithms for a stochastic Lorenz 63 model and for a stochastic volatility model with real data.

1. Introduction

State-space models are a popular tool in many fields of science and engineering where researchers and practitioners deal with uncertainty in dynamical systems. A typical state space model consists of:

- A random sequence of state vectors, $\boldsymbol{x}_t$, that contain the variables of interest for the description of the real-world system at hand, but cannot be observed (at least completely).
- A random sequence of noisy observation vectors, $\boldsymbol{y}_t$, where each $\boldsymbol{y}_t$ can be related to the state $\boldsymbol{x}_t$ through some conditional probability distribution.
- A vector $\boldsymbol{\theta}$ of static model parameters that determine the model behaviour and, typically, have to be estimated from the available data.

Classical filtering methods [2–7], including both Kalman-based algorithms and Monte Carlo schemes (particle filters [6]) tackle the problem of predicting and tracking the states $\boldsymbol{x}_t$ using the observations $\boldsymbol{y}_t$, while assuming that the parameters $\boldsymbol{\theta}$ are given. This is hardly ever the case in practice, though, and the fixed parameters $\boldsymbol{\theta}$ have to be estimated from the data $\boldsymbol{y}_t$ as well. The joint tracking of $\boldsymbol{x}_t$ and estimation of $\boldsymbol{\theta}$ involves several practical and theoretical difficulties. The straightforward approach is the use of state augmentation [8–13], where an extended state is introduced that includes both the static parameters $\boldsymbol{\theta}$ and the dynamical variables $\boldsymbol{x}_t$. This methodology can be applied with any standard filtering technique such as Kalman-like methods (either extended [8] or sigma-point-based [9] approximations) and particle filters (PFs) [10,12,14]. In the case of PFs, artificial dynamics are usually introduced for the fixed parameters, reinterpreting them as slow-changing dynamical variables in order to avoid the degeneracy of the Monte Carlo approximation. Both with Kalman and particle filtering techniques, state augmentation is easy to apply but the resulting methods are often inefficient and lack theoretical guarantees. When the posterior distribution of the static parameters can be represented by a set of finite statistics, classical state-augmentation can be replaced by a two-stage procedure where one first samples the posterior of the parameters and then the states (conditional on the parameters). Such methods are often referred to as particle learning [15–18]. The assumption of having a description of the posterior distribution of the parameters is rather restrictive, though. A more general strategy is the use of recursive maximum likelihood methods [11,19–22]. These techniques are well-principled and can be applied to a broad class of models. However, they provide point estimates of the unknowns as the observations are collected and not full posterior distributions. Therefore, the uncertainty is not quantified.

On the other hand, some major advances in the last few years have led to well-principled algorithms that can solve (numerically) the joint model inference (parameter estimation) and state track-

[*] Corresponding author.
E-mail addresses: saperezv@pa.uc3m.es (S. Pérez-Vieites), joaquin.miguez@uc3m.es (J. Míguez).

https://doi.org/10.1016/j.sigpro.2021.108295

ing problems. They are fundamentally Bayesian methods that aim at computing the posterior probability distribution of the unknown states and parameters given sequentially collected observations. This approach not only provides point estimates but also information about the uncertainty of those estimates. Some examples are the sequential Monte Carlo square (SMC^2) [23], the particle Markov chain Monte Carlo (PMCMC) [24] and the nested particle filter (NPF) [25] methods. Both SMC^2 and PMCMC are popular methods that have been used by many authors in different types of problems [26–30]. However, they are batch (non recursive) techniques. Therefore, every time a new observation is introduced, the whole sequence of observations may need to be processed in order to compute the new Bayes estimator. A closely-related methodology that is better suited for long sequences of observed data is the NPF [25,31]. It applies the same principles as SMC^2, but NPFs are purely recursive. It is a scheme with two intertwined layers of Monte Carlo methods, one inside the other, using the "inner" layer to track the dynamic state variables and the "outer" layer for parameter estimation. However, since the NPF uses Monte Carlo in both layers of filters, its computational cost becomes prohibitive in high-dimensional problems. With the aim of reducing this cost, the class of nested hybrid filters (NHFs) was introduced in [1]. NHFs are recursive algorithms with the same multi-layer structure as NPFs but they enable the use of non-Monte Carlo filtering techniques in the "inner" layer (state tracking). Therefore, an NHF is a general scheme that can approximate the posterior probability distribution of the static parameters with a recursive Monte Carlo or quasi-Monte Carlo [32] method and combine it with different filtering (Monte Carlo or Kalman-based) techniques in order to approximate the posterior probability distribution of the dynamical state variables of the system. The tools and techniques in [1] *do not* enable the use of non-Monte Carlo approximations (e.g., Kalman filters) in the outer layer.

In this paper, we introduce an extension of the NHF methodology of [1] that enables the use of Gaussian schemes in both layers of the nested filtering algorithm. The new scheme, therefore, is a methodological generalization of the algorithms in [1,25]. The class of nested filters for joint parameter estimation and state tracking that can be devised using the approach in this paper is broad and it includes combinations of Gaussian (Kalman-like) and particle filters in any of the two layers. The resulting algorithms remain recursive and yield numerical approximations of the posterior probability of the unknown state variables and parameters using the sequentially collected observations. In particular, it is possible to design nested Kalman filters which are computationally much faster than the methods in [25] or [1] and yield Gaussian approximations of the posterior distributions of interest.

To be specific, in this work we explain in detail the use of deterministic Gaussian approximations (such as the unscented Kalman filter (UKF) [5] or the cubature Kalman filter (CKF) [33]) in the outer layer of the nested filtering scheme. Either particle or Gaussian (Kalman) filters can be easily plugged into the inner layer but, since our main aim is the reduction of computational costs, we implement extended Kalman filters in our experiments.

The key difficulty to be tackled when using non-Monte Carlo methods in the outer layer of a nested filter is to keep the algorithm recursive. This was achieved for the Monte Carlo methods in [25] and [1] using a "jittering" procedure that cannot be extended to Gaussian filters in a practical way. Instead, we place a condition on the update of the filter in the outer layer that depends on a distance defined on the parameter space. When the distance between consecutive parameter updates falls below a prescribed threshold the algorithm operates in a purely recursive manner. This approach can work adequately when the posterior probability distributions of the state variables are continuous with respect to (w.r.t.) the unknown parameters, and we prove that this is the case under regularity assumptions on the state-space model.

In order to assess the performance of the proposed nested methods we have implemented a recursive scheme that employs a UKF in the outer layer (for parameter estimation) and a bank of extended Kalman filters (EKFs) in the inner layer (for state tracking). We have carried out a simulation study to compare the performance of this algorithm with two state-augmented Gaussian filters (a UKF and an ensemble Kalman filter (EnKF) [34]) as well as nested hybrid algorithm that combines a particle filter in the outer layer with EKFs in the inner layer in the manner described in [1]. The methods are applied in two examples: the first one consists in tracking a stochastic Lorenz 63 model with three unknown parameters in the state equation, while in the second one we perform inference on a stochastic volatility model using real-world time-series data (namely, euro-to-USD exchange rates between 2014 and 2016).

The rest of the paper is organized as follows. In Section 2 we describe the class of state-space models with unknown parameters to be studied through the paper. In Section 3 we derive the family of nested Gaussian filters with sigma-point approximations in the outer layer. Computer simulation results are presented in Section 4, while in Section 5 we describe an example with real data. Finally, Section 6 is devoted to the conclusions.

2. Problem statement

2.1. State space models

We are interested in the class of Markov state-space dynamical systems with additive noise that can be described by the pair of equations

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) + \boldsymbol{v}_t, \tag{1}$$

$$\boldsymbol{y}_t = g(\boldsymbol{x}_t, \boldsymbol{\theta}) + \boldsymbol{r}_t, \tag{2}$$

where $t \in \mathbb{N}$ denotes discrete time, $\boldsymbol{x}_t \in \mathbb{R}^{d_x}$ is the d_x-dimensional system state, $f : \mathbb{R}^{d_x} \times \mathbb{R}^{d_\theta} \longrightarrow \mathbb{R}^{d_x}$ and $g : \mathbb{R}^{d_x} \times \mathbb{R}^{d_\theta} \longrightarrow \mathbb{R}^{d_y}, d_x \geq d_y$, are possibly nonlinear functions parameterized by a (random but fixed) vector of unknown parameters, $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}, \boldsymbol{y}_t \in \mathbb{R}^{d_y}$ is the observation vector at time t and $\boldsymbol{v}_t$ and $\boldsymbol{r}_t$ are zero-mean random vectors playing the roles of state and observations noises.

The system of Eqs. (1) and (2) can be described in terms of a set of relevant probability density functions (pdfs)[1], specifically

$$\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0), \quad \boldsymbol{\theta} \sim p(\boldsymbol{\theta}), \tag{3}$$

$$\boldsymbol{x}_t \sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\theta}), \tag{4}$$

$$\boldsymbol{y}_t \sim p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta}), \tag{5}$$

where $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x}_0)$ are the a *priori* pdfs of the parameters and the state, respectively, $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})$ is the conditional density of the state $\boldsymbol{x}_t$ given $\boldsymbol{x}_{t-1}$ and the parameter vector $\boldsymbol{\theta}$, and $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$ is the conditional pdf of the observation $\boldsymbol{y}_t$ given $\boldsymbol{x}_t$ and $\boldsymbol{\theta}$. We assume that $\boldsymbol{y}_t$ is conditionally independent of all other observations (given $\boldsymbol{x}_t$ and $\boldsymbol{\theta}$) and the prior pdfs of the state, $p(\boldsymbol{x}_0)$, and the parameters, $p(\boldsymbol{\theta})$, are known and the corresponding probability distributions are independent.

[1] We adopt an argument-wise notation for pdfs. If we have two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, we write $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ for their respective pdfs, which are possibly different. In a similar way, $p(\boldsymbol{x}, \boldsymbol{y})$ denotes the joint pdf of the two random variables and $p(\boldsymbol{x}|\boldsymbol{y})$ denotes the conditional pdf of $\boldsymbol{x}$ given $\boldsymbol{y}$.

2.2. Model inference

The key difficulty in this class of models is the Bayesian estimation of the parameter vector $\boldsymbol{\theta}$, since its calibration is necessary in order to track the state variables and predict the evolution of the system. From the viewpoint of Bayesian analysis, we aim at computing the posterior pdf $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$ as it contains all the relevant information for the estimation task at discrete time t. However, this pdf can be written as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) = \int p(\boldsymbol{\theta}, \boldsymbol{x}_t|\boldsymbol{y}_{1:t})d\boldsymbol{x}_t, \tag{6}$$

leading naturally to approximations for $p(\boldsymbol{\theta}, \boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ for each t. This means that when computing $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$ we may not only estimate the parameter vector $\boldsymbol{\theta}$, but we may also implicitly track the state dynamical variables. The main aim of this paper is to obtain a Gaussian approximation of $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$ within a nested Gaussian filtering scheme, whose second layer of filters will provide, in addition, Gaussian approximations for $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})$.

3. Nested Gaussian filters

In this section, we introduce a class of nested filter for state-space models with unknown parameters that combine different types of Gaussian approximations in the inner and outer layers. We outline the methodology used to obtain the Gaussian approximations of $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$ (in the outer layer) and $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})$ (in the inner layer).

In the sequel we keep using $p(\cdot)$ to denote the actual pdfs. We aim, however, at constructing Gaussian approximations of the posterior pdfs induced by the state-space model (3)–(5) and the sequence of observations. For this purpose, we introduce notation $\mathcal{N}(\boldsymbol{x}|\bar{\boldsymbol{x}}, \boldsymbol{C})$ to denote the Gaussian pdf with mean $\bar{\boldsymbol{x}}$ and covariance matrix $\boldsymbol{C}$. We will show how to recursively compute approximations $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}_t, \hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}})$, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{x}_t|\hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}}, \hat{\boldsymbol{C}}_{t|t,\boldsymbol{\theta}}^{\boldsymbol{x}})$ and $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t|\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{C}}_t^{\boldsymbol{x}})$.

3.1. Sequential Gaussian approximation

Let us aim at computing expectations of the form $\mathbb{E}[f(\boldsymbol{\theta})|\boldsymbol{y}_{1:t}] = \int f(\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})d\boldsymbol{\theta}$ for some test function of the parameters, $f(\boldsymbol{\theta})$. Different choices of $f(\cdot)$ enable us to compute different statistics, e.g., for $f(\boldsymbol{\theta}) = \boldsymbol{\theta}$ we obtain the posterior mean $\bar{\boldsymbol{\theta}}_t = \mathbb{E}[\boldsymbol{\theta}|\boldsymbol{y}_{1:t}]$ while if we let

$$f(\boldsymbol{\theta}) = \left(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_t\right)^\top \left(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_t\right)$$

then we obtain the posterior covariance matrix of $\boldsymbol{\theta}$. Using Bayes' rule, we can express the posterior pdf of $\boldsymbol{\theta}$ as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} \times p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1}), \tag{7}$$

hence, we can rewrite the posterior expectation as

$$\mathbb{E}[f(\boldsymbol{\theta})|\boldsymbol{y}_{1:t}] = \int \psi(\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})d\boldsymbol{\theta}, \tag{8}$$

where the function $\psi(\boldsymbol{\theta})$ is constructed as

$$\psi(\boldsymbol{\theta}) := \frac{f(\boldsymbol{\theta})p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})}. \tag{9}$$

If we assume that $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$ is Gaussian, then we can approximate (8) using cubature rules [33] or the unscented transform (UT) [5]. Specifically, a Gaussian approximation $\mathcal{N}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}_{t-1}, \boldsymbol{C}_{t-1}^{\boldsymbol{\theta}}) \approx p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$ can be represented at time t by a set of reference points and weights, $\{\boldsymbol{\theta}_{t-1}^i, w_{t-1}^i\}_{0\le i\le M-1}, M = 2d_\theta + 1$ (see Appendix B), which in turn we may use to approximate the integral in (8) as

$$\int \psi(\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})d\boldsymbol{\theta} \approx \sum_{i=1}^{M} \psi(\boldsymbol{\theta}_{t-1}^i)w_{t-1}^i. \tag{10}$$

On the other hand, the pdf in the denominator of expression (9), $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})$, can be written as

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{y}_t, \boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})d\boldsymbol{\theta}, \tag{11}$$

where the joint pdf of $\boldsymbol{y}_t$ and $\boldsymbol{\theta}$ given all previous observations can be decomposed as

$$p(\boldsymbol{y}_t, \boldsymbol{\theta}|\boldsymbol{y}_{1:t-1}) = p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1}). \tag{12}$$

Then, the integral in (11) can also be approximated using the same set of reference points and weights as

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}) \approx \sum_{i=0}^{M-1} p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)w_{t-1}^i. \tag{13}$$

Finally, we can approximate the pdf $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i), i = 0, \ldots, M-1$, using a bank of M Gaussian filters placed in the second layer of the nested filter [1]. Once these densities are computed, we can approximate $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})$ as in (13).

The argument above enables us to approximate any integral $\int f(\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})d\boldsymbol{\theta}$. In particular, we can compute the mean vector and covariance matrix of $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}_t, \hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}})$ by taking $f(\boldsymbol{\theta}) = \boldsymbol{\theta}$ and $f(\boldsymbol{\theta}) = (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t)(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t)^\top$, respectively, where

$$\hat{\boldsymbol{\theta}}_t = \int \boldsymbol{\theta} p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})d\boldsymbol{\theta}. \tag{14}$$

Specifically, we obtain the formulation for approximating the mean parameter vector, $\hat{\boldsymbol{\theta}}_t$, and its covariance matrix, $\hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}}$, sequentially as

$$\hat{\boldsymbol{\theta}}_t \approx \sum_{i=0}^{M-1} \boldsymbol{\theta}_{t-1}^i \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} w_{t-1}^i \quad \text{and} \tag{15}$$

$$\hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}} \approx \sum_{i=0}^{M-1} (\boldsymbol{\theta}_{t-1}^i - \hat{\boldsymbol{\theta}}_t)(\boldsymbol{\theta}_{t-1}^i - \hat{\boldsymbol{\theta}}_t)^\top \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} w_{t-1}^i. \tag{16}$$

We outline the procedure for the sequential computation of the Gaussian approximations $\mathcal{N}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}_t, \hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}}) \approx p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}), t = 1, 2, \ldots,$ in Algorithm 1. The calculations done in the second layer of filters are summarized in 2(a). Notice that, at any time $t \ge 1$, we update the reference points $\boldsymbol{\theta}_{t-1}^i, i = 0, \ldots, M-1$, and, therefore, we

Algorithm 1 Nested Gaussian filters.

```
Inputs:
  - Prior pdfs $p(\boldsymbol{x}_0)$ and $p(\boldsymbol{\theta})$. Assume that either $p(\boldsymbol{x}_0)$ is Gaussian
    or a Gaussian approximation is available.

Procedure:
  1 Initialization
    (a) Generate $M$ reference points, $\boldsymbol{\theta}_0^i$, from $p(\boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_0, \boldsymbol{C}_0^{\boldsymbol{\theta}})$
        for $i = 0, \ldots, M-1$, with weights $w_0^i$.
  2 Sequential step, $t \ge 1$.
    (a) For each $i = 0, \ldots, M-1$, use a Gaussian filter to approximately compute $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$.
    (b) Compute $\hat{\boldsymbol{\theta}}_t$ and $\hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}}$ via (15) and (16).
    (c) Generate new reference points $\boldsymbol{\theta}_t^i$ and weights $w_t^i, i =$
        $0, \ldots, M-1$, from $\hat{\boldsymbol{\theta}}_t$ and $\hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}}$.

Outputs: $\hat{\boldsymbol{\theta}}_t$ and $\hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}}$.
```

need to run the M Gaussian filters in the second layer from scratch (i.e., from $n = 0$ to $n = t$) in order to (approximately) evaluate the densities $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$. Thus, Algorithm 1 is sequential but not recursive and, as a consequence, not well suited to handle long sequences of observations.

3.2. Recursive algorithm

For every new observation vector $\boldsymbol{y}_t$, in Algorithm 1 the pdfs $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$ are computed by running the nested filters from time 0 until the current time t, which makes the computational cost increase with t^2.

However, the entries of the covariance matrix, $\hat{\boldsymbol{C}}_t^{\boldsymbol{\theta}}$, also tend to stabilize over time, which makes the difference between consecutive reference points, $\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i$, decrease. If we also assume that the function $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ is continuous in $\boldsymbol{\theta}$, then we can make the computation recursive by assuming that $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i) \approx p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-2}^i)$ when $\boldsymbol{\theta}_{t-1}^i \approx \boldsymbol{\theta}_{t-2}^i$. For the sake of clarity we summarize the steps for computing $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$ in Algorithm 2, relying on a bank of EKFs. Let us remark that the second layer of the nested algorithm can be implemented using a variety of filters, e.g., particle filters as in [25] or Gaussian filters as in [1], including UKFs as we have done for the first layer. We choose a bank of EKFs simply because it is the computationally less demanding alternative.

Algorithm 3 outlines a recursive nested Gaussian filter with a UKF/CKF in the first layer and EKFs in the second layer. It can be seen as a recursive and explicit implementation of Algorithm 1. The initialization remains the same (step 1(a)), computing M reference points $\boldsymbol{\theta}_0^i$ and weights $w_0^i, i = 0, \ldots, M-1$, from the prior $p(\boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_0, \boldsymbol{C}_0^{\boldsymbol{\theta}})$. Also, we initialize the state and its covariance matrix in every Gaussian filter of the second layer (step 1(b)) by setting $\hat{\boldsymbol{x}}_0^i = \hat{\boldsymbol{x}}_0$ and $\hat{\boldsymbol{C}}_0^{\boldsymbol{x},i} = \boldsymbol{C}_0^{\boldsymbol{x}}, i = 0, \ldots, M-1$, from the prior $p(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0|\hat{\boldsymbol{x}}_0, \boldsymbol{C}_0^{\boldsymbol{x}})$.

The sequential procedure starts by approximating $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$ with the second layer of Gaussian filters (step 2(a)i). This is done differently depending on whether we assume $\boldsymbol{\theta}_{t-1}^i \approx \boldsymbol{\theta}_{t-2}^i$ or not. To be specific, the norm[2] $\|\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i\|_p$ is computed and compared against a prescribed relative threshold $\lambda > 0$ in order to determine whether the prediction and update steps in the second layer of filters can be performed recursively or not. Specifically:

- If $\|\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i\|_p < \lambda\|\boldsymbol{\theta}_{t-2}^i\|_p$ is not satisfied for $\boldsymbol{\theta}_{t-1}^i$, the ith filter runs from scratch following the scheme in Algorithm 2.
- When $\|\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i\|_p < \lambda\|\boldsymbol{\theta}_{t-2}^i\|_p$ is satisfied for $\boldsymbol{\theta}_{t-1}^i$, only one prediction and update step (from time $t-2$ to time $t-1$) is needed. In particular, we make the approximation $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i) \approx p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-2}^i)$.

Algorithm 2 Extended Kalman filter conditional on $\boldsymbol{\theta}_{t-1}^i$, used in the second layer of the nested filter.

Inputs:

- Prior pdf $p(\boldsymbol{x}_0)$ and parameter vector $\boldsymbol{\theta}_{t-1}^i$.
- State-space model described in Eqs. (1) and (2). In particular, $f(\cdot)$ denotes the drift function in the state eq. (1) and $g(\cdot)$ is the observation function in eq. (2). The covariance of the state noise is denoted $\boldsymbol{V}$ and the covariance of the observation noise is denoted $\boldsymbol{R}$.

Procedure:

1 Initialization

(a) Assume $p(\boldsymbol{x}_0)$ is Gaussian with mean $\hat{\boldsymbol{x}}_0$ and covariance $\hat{\boldsymbol{C}}_0^{\boldsymbol{x}}$, i.e., $p(\boldsymbol{x}_0) \approx \mathcal{N}(\boldsymbol{x}_0|\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{C}}_0^{\boldsymbol{x}})$.

2 Sequential step, $t \geq 1$.

(a) **Prediction step.** Compute

$$\begin{aligned} \hat{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i} &= f(\hat{\boldsymbol{x}}_{t-1|t-1,\boldsymbol{\theta}_{t-1}^i}, \boldsymbol{\theta}_{t-1}^i), \\ \hat{\boldsymbol{C}}^{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i} &= \boldsymbol{J}_{f,\hat{\boldsymbol{x}}_{t-1|t-1,\boldsymbol{\theta}_{t-1}^i}} \hat{\boldsymbol{C}}^{\boldsymbol{x}}_{t-1|t-1,\boldsymbol{\theta}_{t-1}^i} \boldsymbol{J}^{\top}_{f,\hat{\boldsymbol{x}}_{t-1|t-1,\boldsymbol{\theta}_{t-1}^i}} + \boldsymbol{V}, \end{aligned} \tag{20}$$

where

$\boldsymbol{J}_{f,\boldsymbol{x}}$ is the Jacobian matrix of $f(\cdot)$ evaluated at $\hat{\boldsymbol{x}}_{t-1|t-1,\boldsymbol{\theta}_{t-1}^i}$.

(b) Approximate $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i) \approx \mathcal{N}(\boldsymbol{x}_t|\hat{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i}, \hat{\boldsymbol{C}}^{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i})$ and compute

$$\begin{aligned} & p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i) \\ &= \int p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}_{t-1}^i) p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i) d\boldsymbol{x}_t \\ &\approx \int p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}_{t-1}^i) \mathcal{N}(\boldsymbol{x}_t|\hat{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i}, \hat{\boldsymbol{C}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i}) d\boldsymbol{x}_t. \end{aligned} \tag{21}$$

(c) **Update step.** Compute

$$\hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i} = \hat{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i} + \boldsymbol{K}_t(\boldsymbol{y}_t - g(\hat{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i}, \boldsymbol{\theta}_{t-1}^i)), \tag{22}$$

$$\hat{\boldsymbol{C}}^{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i} = (\boldsymbol{I}_{d_x} - \boldsymbol{K}_t\boldsymbol{J}_g)\hat{\boldsymbol{C}}^{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i}, \tag{23}$$

$$\begin{aligned} \boldsymbol{K}_t &= \hat{\boldsymbol{C}}^{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i} \boldsymbol{J}^{\top}_{g,\hat{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i}} \times \\ &\times (\boldsymbol{J}_{g,\hat{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i}} \hat{\boldsymbol{C}}^{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i} \boldsymbol{J}^{\top}_{g,\hat{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i}} + \boldsymbol{R})^{-1}, \end{aligned}$$

where

$\boldsymbol{J}_{g,\boldsymbol{x}}$ is the Jacobian matrix of $g(\cdot)$ evaluated at $\hat{\boldsymbol{x}}_{t|t-1,\boldsymbol{\theta}_{t-1}^i}$. Approximate $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta}_{t-1}^i) \approx \mathcal{N}(\boldsymbol{x}_t|\hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i}, \hat{\boldsymbol{C}}^{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i})$.

Outputs: $\hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i}, \hat{\boldsymbol{C}}^{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i}$ and $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$.

In either case, we use $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$ in order to compute $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$ as in step 2(a) of Algorithm 2. Finally, we can compute the mean vector $\hat{\boldsymbol{\theta}}_t$ and the covariance matrix $\hat{\boldsymbol{C}}_t^{\boldsymbol{\theta},i}$ at time t in step 2(b), by using (15) and (16). We prepare the new reference points $\boldsymbol{\theta}_t^i$ and their weights w_t^i from $\mathcal{N}(\hat{\boldsymbol{\theta}}_t, \hat{\boldsymbol{C}}_t^{\boldsymbol{\theta},i})$ for the next time step.

3.3. State tracking

We can take advantage of the filters in the second layer in order to provide state estimates as well. Let us write the expectation of $\boldsymbol{x}_t$ as

$$\mathbb{E}[\boldsymbol{x}_t|\boldsymbol{y}_{1:t}] = \int \left[\int \boldsymbol{x}_t p\left(\boldsymbol{x}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t}\right) d\boldsymbol{x}_t \right] p\left(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}\right) d\boldsymbol{\theta}, \tag{24}$$

[2] Although other metrics $d(\boldsymbol{\theta}_{t-1}^i, \boldsymbol{\theta}_{t-2}^i)$ could be used, we adopt p-norms of the difference $\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i$ for this work. This is a flexible setup that admits several variants, e.g.,

$$\|\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i\|_1 = \sum_{j=1}^{d_\theta} |\boldsymbol{\theta}_{t-1,j}^i - \boldsymbol{\theta}_{t-2,j}^i|, \tag{17}$$

$$\|\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i\|_2 = \sqrt{\sum_{j=1}^{d_\theta} (\boldsymbol{\theta}_{t-1,j}^i - \boldsymbol{\theta}_{t-2,j}^i)^2} \quad \text{and} \tag{18}$$

$$\|\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i\|_\infty = \max_{1 \leq j \leq d_\theta} |\boldsymbol{\theta}_{t-1,j}^i - \boldsymbol{\theta}_{t-2,j}^i|; \tag{19}$$

i.e., the taxicab norm or Manhattan norm ($p = 1$), the Euclidean norm ($p = 2$) and the maximum norm ($p = \infty$) respectively.

Algorithm 3 Recursive nested Gaussian filters.

```
Inputs:
 - Prior pdfs $p(\boldsymbol{x}_0)$ and $p(\boldsymbol{\theta})$.
 - A fixed threshold $\lambda > 0$.
Procedure:
 1 Initialization
   (a) Generate $M$ reference points, $\boldsymbol{\theta}_0^i$, for $p(\boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{\theta}_0, \mathbf{C}_0^{\boldsymbol{\theta}})$, $i = 0, \dots, M-1$, with weights $w_0^i$.
   (b) If $p(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0 | \hat{\boldsymbol{x}}_0, \mathbf{C}_0^{\boldsymbol{x}})$, then set $\hat{\boldsymbol{x}}_0^i = \hat{\boldsymbol{x}}_0$ and $\hat{\mathbf{C}}_0^{\boldsymbol{x},i} = \mathbf{C}_0^{\boldsymbol{x}}$ for $i = 0, \dots, M-1$.
 2 Sequential step, $t \geq 1$.
   (a) For $i = 0, \dots, M-1$:
       i If $\|\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i\|_p < \lambda \|\boldsymbol{\theta}_{t-2}^i\|_p$, then compute $p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$ from $p(\boldsymbol{x}_{t-1} | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i) \approx p(\boldsymbol{x}_{t-1} | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-2}^i)$, where $p(\boldsymbol{x}_{t-1} | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-2}^i) \approx \mathcal{N}(\boldsymbol{x}_{t-1} | \hat{\boldsymbol{x}}_{t-1|t-1,\boldsymbol{\theta}_{t-2}^i}, \hat{\mathbf{C}}^{\boldsymbol{x}}_{t-1|t-1,\boldsymbol{\theta}_{t-2}^i})$.
         Else, approximate $p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$ from the prior $p(\boldsymbol{x}_0)$.
       ii Use $p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$ to compute $p(\boldsymbol{y}_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$.
   (b) Compute $\hat{\boldsymbol{\theta}}_t, \hat{\mathbf{C}}_t^{\boldsymbol{\theta}}, \hat{\boldsymbol{x}}_t$ and $\hat{\mathbf{C}}_t^{\boldsymbol{x}}$ from (15), (16), (27) and (28), respectively.
   (c) Generate reference points $\boldsymbol{\theta}_t^i$ and weights $w_t^i$ from $\hat{\boldsymbol{\theta}}_t$ and $\hat{\mathbf{C}}_t^{\boldsymbol{\theta}}$ for $i = 0, \dots, M-1$.
Outputs: $\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{\theta}}_t, \hat{\mathbf{C}}_t^{\boldsymbol{x}}$ and $\hat{\mathbf{C}}_t^{\boldsymbol{\theta}}$.
```

where the integral in square brackets can be approximated by the M Gaussian filters of the second layer. In this case, we assume they are the EKFs of Algorithm 2 conditional on $\boldsymbol{\theta} = \boldsymbol{\theta}_{t-1}^i$. This yields a Gaussian approximation $p(\boldsymbol{x}_t | \boldsymbol{\theta}_{t-1}^i, \boldsymbol{y}_{1:t}) \simeq \mathcal{N}(\boldsymbol{x}_t | \hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i}, \hat{\mathbf{C}}_{t|t,\boldsymbol{\theta}_{t-1}^i})$, where

$$\hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i} \approx \mathbb{E}[\boldsymbol{x}_t | \boldsymbol{\theta}_{t-1}^i, \boldsymbol{y}_{1:t}] \quad \text{and} \tag{25}$$

$$\hat{\mathbf{C}}^{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i} \approx \mathbb{E}[(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i})(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i})^\top | \boldsymbol{y}_{1:t}, \boldsymbol{\theta}_{t-1}^i]. \tag{26}$$

Then, a Gaussian approximation $p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t}) \approx \mathcal{N}(\boldsymbol{x}_t | \hat{\boldsymbol{x}}_t, \hat{\mathbf{C}}_t^{\boldsymbol{x}})$ can be constructed, where

$$\hat{\boldsymbol{x}}_t \approx \sum_{i=0}^{M-1} \hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i} \frac{p(\boldsymbol{y}_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)}{p(\boldsymbol{y}_t | \boldsymbol{y}_{1:t-1})} w_{t-1}^i \quad \text{and} \tag{27}$$

$$\hat{\mathbf{C}}_t^{\boldsymbol{x}} \approx \sum_{i=0}^{M-1} (\hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i} - \hat{\boldsymbol{x}}_t)(\hat{\boldsymbol{x}}_{t|t,\boldsymbol{\theta}_{t-1}^i} - \hat{\boldsymbol{x}}_t)^\top \frac{p(\boldsymbol{y}_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)}{p(\boldsymbol{y}_t | \boldsymbol{y}_{1:t-1})} w_{t-1}^i. \tag{28}$$

3.4. Continuity of the conditional filter pdf

The key to keep Algorithm 3 recursive is the test in step 2(a), which sets

$$p(\boldsymbol{x}_{t-1} | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i) \approx \mathcal{N}(\boldsymbol{x}_{t-1} | \hat{\boldsymbol{x}}_{t-1|t-1,\boldsymbol{\theta}_{t-2}^i}, \hat{\mathbf{C}}^{\boldsymbol{x}}_{t-1|t-1,\boldsymbol{\theta}_{t-2}^i}) \tag{29}$$

when $\frac{\|\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i\|_p}{\|\boldsymbol{\theta}_{t-2}^i\|_p} < \lambda$ for some prescribed threshold $\lambda > 0$. This step relies on the assumption that $p(\boldsymbol{x}_{t-1} | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}) \approx p(\boldsymbol{x}_{t-1} | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}')$ when $\boldsymbol{\theta} \approx \boldsymbol{\theta}'$, i.e., we are assuming that the conditional filtering pdf $p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t}, \boldsymbol{\theta})$ is a continuous function of the parameter $\boldsymbol{\theta}$. In this section we state sufficient conditions for the conditional filter $p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t}, \boldsymbol{\theta})$ to be Lipschitz-continuous.

For conciseness, let us denote

$$\pi_t(\boldsymbol{x}_t | \boldsymbol{\theta}) := p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t}, \boldsymbol{\theta}), \tag{30}$$

$$\xi_t(\boldsymbol{x}_t | \boldsymbol{\theta}) := p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}), \quad \text{and} \tag{31}$$

$$\eta_t(\boldsymbol{y}_t | \boldsymbol{\theta}) := p(\boldsymbol{y}_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}). \tag{32}$$

Hereafter we assume that the observation sequence $\{\boldsymbol{y}_t, t \geq 1\}$ is arbitrary but fixed (i.e., deterministic). Additionally, we impose the following regularity assumptions:

Assumption 1. The conditional pdfs $\pi_t(\boldsymbol{x}_t | \boldsymbol{\theta})$, $\xi_t(\boldsymbol{x}_t | \boldsymbol{\theta})$ and $\eta_t(\boldsymbol{y}_t | \boldsymbol{\theta})$ exist for every $t \geq 1$, every $\boldsymbol{x}_t \in \mathbb{R}^{d_x}$ and every parameter vector $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^{d_\theta}$, where Θ denotes the parameter space.

Assumption 2. The transition pdf $p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\theta})$ is Lipschitz w.r.t. $\boldsymbol{\theta}$, i.e., there exists a constant $0 < L < \infty$ such that

$$\sup_{\boldsymbol{x}_{t-1} \in \mathbb{R}^{d_x}} \int |p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\theta}) - p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\theta}')| d\boldsymbol{x}_t < L \|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \tag{33}$$

for every $t \geq 1$ and every pair $(\boldsymbol{\theta}, \boldsymbol{\theta}') \in \Theta \times \Theta$.

Remark 1. In Assumption 2, we denote $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| = \sqrt{\sum_{i=1}^{d_\theta} (\theta_i - \theta_i')^2}$, the Euclidean distance between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$.

Assumption 3. The conditional pdfs $p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta})$ are strictly positive and uniformly Lipschitz w.r.t. $\boldsymbol{\theta}$. In particular, $p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta}) > 0$ and

$$\sup_{\boldsymbol{x}_t \in \mathbb{R}^{d_x}} \frac{|p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta}) - p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta}')|}{\eta_t(\boldsymbol{y}_t | \boldsymbol{\theta})} < G_t \|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \tag{34}$$

for some positive $G_t < \infty$.

Assumption 4. The ratio $\frac{p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta})}{\eta_t(\boldsymbol{y}_t | \boldsymbol{\theta})}$ is bounded. Specifically, there exist finite constants $0 < M_t < \infty$ such that

$$\sup_{\substack{\boldsymbol{\theta} \in \Theta \\ \boldsymbol{x}_{t-1} \in \mathbb{R}^{d_x}}} \frac{p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta})}{\eta_t(\boldsymbol{y}_t | \boldsymbol{\theta})} < M_t. \tag{35}$$

Assumptions 1 and 2 are rather mild and easy to check for a given state-space model. Assumptions 3 and 4, on the other hand, may be restrictive in some problems. We note, however, that for fixed $\boldsymbol{y}_t, t \geq 1$, and a compact parameter support $\Theta \subset \mathbb{R}^{d_\theta}$, the factor $\eta_t(\boldsymbol{y}_t | \boldsymbol{\theta})$ can often be bounded away from 0, while $p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta})$ is typically upper bounded. In any case, Assumptions 1–4 lead to the result below regarding the continuity of the filter, $\pi_t(\boldsymbol{x}_t | \boldsymbol{\theta})$, and predictive, $\xi_t(\boldsymbol{x}_t | \boldsymbol{\theta})$, pdfs w.r.t. the parameter vector $\boldsymbol{\theta}$.

Proposition 1. *: If Assumptions 1 to 4 hold, there exist sequences of finite constants $\tilde{L}_t$ and L_t such that, for $t \geq 1$,*

$$\int |\xi_t(\boldsymbol{x}_t | \boldsymbol{\theta}) - \xi_t(\boldsymbol{x}_t | \boldsymbol{\theta}')| d\boldsymbol{x}_t \leq \tilde{L}_t \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|, \quad \text{and} \tag{36}$$

$$\int |\pi_t(\boldsymbol{x}_t | \boldsymbol{\theta}) - \pi_t(\boldsymbol{x}_t | \boldsymbol{\theta}')| d\boldsymbol{x}_t \leq L_t \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|. \tag{37}$$

See Appendix A for a proof.

4. Example: the stochastic Lorenz 63 model

4.1. Stochastic Lorenz 63 model

Consider the 3-dimensional continuous-time stochastic process $\boldsymbol{x}(\tau) = [x_1(\tau), x_2(\tau), x_3(\tau)]^\top$, where $\tau \in (0, \infty)$ denotes continuous time, taking values on $\mathbb{R}^3$, whose dynamics are described by the system of stochastic differential equations (SDEs)

$$dx_1 = -S(x_1 - x_2) + \sigma dv_1, \tag{38}$$

$$dx_2 = Rx_1 - x_2 - x_1 x_3 + \sigma dv_2, \tag{39}$$

$$dx_3 = x_1 x_2 - B x_3 + \sigma dv_3, \tag{40}$$

where the v_i's are independent 1-dimensional Wiener processes, $\sigma > 0$ is a known scale parameter and $S, R, B \in \mathbb{R}$ are unknown static model parameters. Using the Euler-Maruyama scheme in order to integrate the SDEs (38)–(40), it is straightforward to cast them into the discrete-time state equation

$$\boldsymbol{x}_{t+1} = f_\Delta(\boldsymbol{x}_t, \boldsymbol{\theta}) + \sqrt{\Delta}\boldsymbol{v}_t, \quad t = 1, 2, \ldots \tag{41}$$

where $f_\Delta : \mathbb{R}^{d_x} \times \mathbb{R}^{d_\theta} \to \mathbb{R}^{d_x}$ $(d_x = d_\theta = 3)$ is the function defined by

$$f_{1,\Delta}(\boldsymbol{x}_t, \boldsymbol{\theta}) = x_{1,t} - \Delta S(x_{1,t} - x_{2,t}),$$
$$f_{2,\Delta}(\boldsymbol{x}_t, \boldsymbol{\theta}) = x_{2,t} + \Delta[(R - x_{3,t})x_{1,t} - x_{2,t}],$$
$$f_{3,\Delta}(\boldsymbol{x}_t, \boldsymbol{\theta}) = x_{3,t} + \Delta(x_{1,t}x_{2,t} - Bx_{3,t}),$$

Δ is the integration time-step (given in continuous-time units), $\boldsymbol{\theta} = (S, R, B)^\top$ is the 3×1 vector of unknown parameters and $\boldsymbol{v}_t$ is a sequence of 3-dimensional Gaussian independent random vectors with zero mean and covariance matrix $\sigma_x^2 \boldsymbol{I}_3$ (with $\boldsymbol{I}_d$ denoting the $d \times d$ identity matrix). Hence, the state transition density $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})$ is Gaussian and can be written down as $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_t|f_\Delta(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}), \sigma^2\Delta\boldsymbol{I}_{d_x})$. This function is Lipschitz on $\boldsymbol{\theta}$.

In order to complete the specification of a state space model, we need to characterize the observations. For our simulation setup we assume linear observations of the form

$$\boldsymbol{y}_t = k_o \begin{bmatrix} x_{1,t} \\ x_{3,t} \end{bmatrix} + \boldsymbol{r}_t, \tag{42}$$

where k_o is a fixed parameter and $\boldsymbol{r}_t \sim \mathcal{N}(\boldsymbol{r}_t|\boldsymbol{0}, \sigma_y^2 \boldsymbol{I}_2)$ is a 2-dimensional additive noise with zero mean and covariance function $\sigma_y^2 \boldsymbol{I}_2$. Therefore, the conditional part of the observations (and hence the likelihood function) is also Gaussian and can be written as $p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t|\boldsymbol{G}\boldsymbol{x}_t, \sigma_y^2\boldsymbol{I}_2)$, where $\boldsymbol{G} = \begin{bmatrix} k_o & 0 & 0 \\ 0 & 0 & k_o \end{bmatrix}$ is the observation matrix. This function has a finite upper bound independent of $\boldsymbol{\theta}$.

Observations are not collected at every time t. Instead we assume that an observation vector is received every M_o steps of the state Eq. (41). This amounts to receiving an observation every $\Delta \times M_o$ continuous-time units.

4.2. Simulation setup

For our computer experiments we have used the stochastic Lorenz 63 model outlined in (41) and (42) in order to generate signals $\boldsymbol{x}_t$ and $\boldsymbol{y}_t, t = \{0, 1, \ldots\}$, used as the ground truth and the data, respectively, for the assessment of the algorithm. We integrate the model with the step-size $\Delta = 2 \times 10^{-4}$ continuous-time units. The true parameters for the generation of the signal and data are $S = 10$, $R = 28$ and $B = \frac{8}{3}$ (which yield underlying chaotic dynamics); while the initial state is Gaussian with mean[3] $\hat{\boldsymbol{x}}_0 = [-6, -5.5, -24.5]^\top$ and covariance matrix $\boldsymbol{I}_3$, i.e., $p(\boldsymbol{x}_0|\hat{\boldsymbol{x}}_0, \boldsymbol{I}_3)$. The noise scale factors, $\sigma^2 = 0.1$ and $\sigma_y^2 = 1$, are assumed known.

For the estimation task we use Algorithm 3. We assume a Gaussian prior distribution for the unknown parameters, namely $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_\theta, \sigma_\theta^2\boldsymbol{I}_3)$, where the a priori mean $\boldsymbol{\mu}_\theta$ is drawn at random from a uniform distribution $\mathcal{U}(\boldsymbol{\theta}_\star - \boldsymbol{\epsilon}, \boldsymbol{\theta}_\star + \boldsymbol{\epsilon})$ for each independent simulation and $\sigma_\theta^2 = 1$. $\boldsymbol{\theta}_\star = [10, 28, \frac{8}{3}]^\top$ are the true parameter vector and the offset vector is $\boldsymbol{\epsilon} = [3, 1, 0.5]^\top$. The algorithm does not collect an observation at every time step, but every $M_o = 5$ discrete-time steps (i.e., every $M_o \times \Delta = 10^{-3}$ continuous-time units). Hence, the prediction step of the state variables at the second layer of nested filter corresponds to $M_o = 5$ discrete-time steps of the difference Eq. (41). When an observation $\boldsymbol{y}_t$ (at discrete time $t = kM_o$, for some $k \in \mathbb{N}$) arrives, both the state and parameter distributions are updated. The length of each simulation runs is $T = 40$ continuous-time units, which amounts to $\frac{T}{\Delta} = 2 \times 10^5$ discrete-time steps of the state Eq. (41).

Table 1
Computational complexity of the algorithms compared in the numerical study. For the EnKF with state-augmentation, M is the number of ensemble elements. For the NPF, N is the number of particles in the first layer and M is the number of particles in the second layer. For the SMC-EKF scheme, N is the number of particles in the first layer. For the UKF-EKF algorithm, q is the average "roll-back" probability in step 2(a)i of Algorithm 3.

Algorithm	Computational complexity
UKF with state augmentation	$\mathcal{O}((d_\theta + d_x)^3 d_y^3 t)$
EnKF with state augmentation	$\mathcal{O}((d_\theta + d_x) d_y^3 M t)$
NPF	$\mathcal{O}(MNt)$
NHF (SMC-EKF)	$\mathcal{O}(d_x^2 d_y^3 N t)$
Algorithm 3 (UKF-EKF)	$\mathcal{O}\left(d_\theta^3 d_x^2 d_y^3 \left[q t^2 + (1-q)t\right]\right)$

We have assessed the ability of several Bayesian computation algorithms to jointly track the state $\boldsymbol{x}_t$ and estimate the parameters $\boldsymbol{\theta} = (S, R, B)^\top$ of this model. To be specific, we have coded and run the following schemes:

- The proposed Algorithm 3 using a UKF in the first layer and a bank of EKFs in the second layer.
- A UKF [5] algorithm with state augmentation [8,9] where the parameters are added to the state vector.
- An EnKF [34] algorithm with state augmentation as well.
- A NHF [1] with a sequential Monte Carlo (SMC) algorithm in the first layer and a bank of EKFs in the second layer.

Table 1 summarizes the computational complexity of the different algorithms that we have compared in our study. For the comparison, we have assumed a roll-back probability q that corresponds to the probability of the event

$$\|\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i\|_p \geq \lambda\|\boldsymbol{\theta}_{t-2}^i\|_p$$

in step 2(a)i of Algorithm 3 for large t. In other words, q represents the average probability of Algorithm 3 to have to approximate the density $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i)$ starting from the prior $p(\boldsymbol{x}_0)$ at time t (an action which has a cost of order t^2). In practice, our simulations show that $q << 1$ for sufficiently large t.

The accuracy of the various algorithms is compared in terms of the normalized mean square error (NMSE) of the predictor of the state and the predictor of the parameters. We assess the empirical NMSE resulting directly from the simulations, namely,

$$\text{NMSE}_{x,t} = \frac{\|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|^2}{\|\boldsymbol{x}_t\|^2}, \quad \text{NMSE}_{\boldsymbol{\theta},t} = \frac{\|\boldsymbol{\theta}_t - \hat{\boldsymbol{\theta}}_t\|^2}{\|\boldsymbol{\theta}_t\|^2}, \tag{43}$$

as well as the averages

$$\text{NMSE}_x = \frac{\Delta}{T}\sum_{t=0}^{\frac{T}{\Delta}-1}\text{NMSE}_{x,t} \quad \text{and} \quad \text{NMSE}_{\boldsymbol{\theta}} = \frac{\Delta}{T}\sum_{t=0}^{\frac{T}{\Delta}-1}\text{NMSE}_{\boldsymbol{\theta},t}.$$

Finally, Table 2 presents a summary of the model and algorithm parameters, and their values, as needed to reproduce the simulation results in this section.

4.3. Numerical results

In the first computer experiments we study the choice of norm $\|\boldsymbol{\theta}_{t-1}^i - \boldsymbol{\theta}_{t-2}^i\|_p$ in step 2(a)i of Algorithm 3. Specifically, we have

[3] The initial vector $\hat{\boldsymbol{x}}_0$ is taken from a deterministic realization of the Lorenz 63 model.

Table 2
Model and algorithm parameters for the simulation setup of Section 4.3.

Parameter	Value
S	10
R	28
B	$\frac{8}{3}$
Δ	2×10^{-4}
T	40
M_o	5
k_o	5
$\boldsymbol{\theta}_\star$	$[10, 28, \frac{8}{3}]^\top$
$\hat{\boldsymbol{x}}_0$	$[-6, -5.5, -24.5]^\top$
σ^2	0.1
σ_y^2	1
σ_θ^2	1
$\boldsymbol{\epsilon}$	$[3, 1, 0.5]^\top$
λ	10^{-3}
M	10
N	120

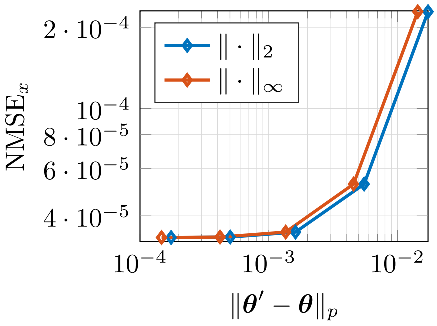


Fig. 1. EKF performance with known parameters, $\boldsymbol{\theta}'$, that are obtained by modifying the true parameters, $\boldsymbol{\theta}$. In the abscissa axis, we represent the average distance of the simulation runs to the ground truh.

considered a setup where the model parameters $\boldsymbol{\theta} = (S, R, B)^\top$ are assumed known and the goal is to track the state $\boldsymbol{x}_t$ using an EKF. We first generate a sequence of observations $\boldsymbol{y}_{1:T}$ from the model with parameters $\boldsymbol{\theta} = (10, 28, \frac{8}{3})^\top$. Then, for this sequence, the EKF runs with a perturbed set of parameters of the form $\boldsymbol{\theta}' = \boldsymbol{\theta} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_e^2)$ is a zero-mean Gaussian perturbation. We carry out 100 independent simulations for each value of σ_e^2 for $\sigma_e^2 = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$.

Fig. 1 summarizes the outcome of this experiment. In particular, it displays the NMSE in the tracking of $\boldsymbol{x}_t$, averaged over all 100 simulation runs, versus the average norms $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2$ and $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\infty$. The plot illustrates that:

1. The NMSE_x is a continuous magnitude w.r.t. the perturbation $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$, both with Euclidean or maximum norms. The NMSE_x remains below 10^{-4} when $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$ is approximately below 10^{-2}.
2. The NMSE_x is slightly higher when the parameter perturbation is given in terms of the norm $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\infty$.

In a second experiment, Fig. 2 shows the results of using Algorithm 3 with both $\|\cdot\|_2$ and $\|\cdot\|_\infty$ norms for several values of λ. Again, each point of the graphs represents the average of 80 independent simulation runs. We display $\mathrm{NMSE}_{\boldsymbol{\theta}}$, $\mathrm{NMSE}_{\boldsymbol{x}}$ and run-times in minutes[4] in Fig. 2(a), 2(b) and 2(c), respectively. In Fig. 2(a), we see that $\mathrm{NMSE}_{\boldsymbol{\theta}}$ increases with λ. This is as expected be cause the larger λ, the worse the approximation $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^i) \sim p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-2}^i)$. We also see that the Euclidean norm $\|\cdot\|_2$ yields a smaller error. However, in the results obtained for NMSE_x in Fig. 2(b), we observe that below $\lambda = 10^{-3}$ there is almost no improvement in the error, and the curve is similar to the one in Fig. 1. Finally, Fig. 2(c) shows that the runtime of the nested filtering Algorithm 3 increases significantly when $\lambda < 10^{-3}$ (because the algorithm takes longer to become strictly recursive). The comparison of the NMSE values in Fig. 2(a) and 2(b) with the run-time in Fig. 2(c) enables us to select the value of the threshold λ in order to attain a certain trade-off between computational cost and accuracy of the estimates provided by Algorithm 3. For the rest of the experiments in this section we set $\lambda = 10^{-3}$. When applied to different models, the effect of λ on the performance of Algorithm 3 can be different and adequate trade-offs may be attained for different values of the threshold. In a practical application, one may run a computer experiment with synthetic data in order to reproduce Fig. 2(a)–(c) for the model of interest and then choose the suitable value of λ to be used with real data.

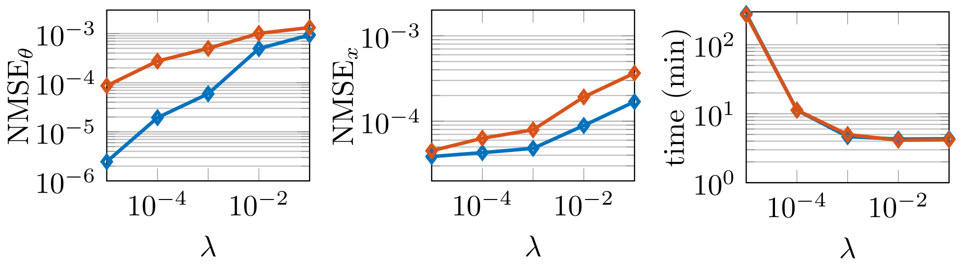
In the next experiment we compare the proposed nested Gaussian filters (Algorithm 3) with two classical methods: the unscented Kalman filter (UKF) [35] and the ensemble Kalman filter (EnKF) [34], both relying on the state-augmentation technique [10,11] to incorporate the unknown parameters. To be specific, this approach implies that the system state $\boldsymbol{x}_t$ is extended with the parameter vector to obtain the augmented state $\tilde{\boldsymbol{x}}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{\theta} \end{bmatrix}$. The UKF and EnKF algorithms are used to track $\tilde{\boldsymbol{x}}_t$ instead of $\boldsymbol{x}_t$.

We have carried out two sets of computer simulations. In the first one we assume that the observation vectors are of the form $\boldsymbol{y}_t = k_o \boldsymbol{x}_t + \boldsymbol{r}_t$, i.e., all the state variables are observed in Gaussian noise. The results are displayed in Fig. 3(a) and (c), which show the NMSE for the parameters $\boldsymbol{\theta}$ and the state $\boldsymbol{x}_t$ over time, respectively, for the three competing algorithms. The nested scheme outperforms the augmented-state methods clearly in terms of parameter estimation (Fig. 3(a)) and by a smaller margin in terms of state tracking (Fig. 3(c)). When the observations are reduced to two state variables $\boldsymbol{y}_t = k_o \begin{bmatrix} x_{1,t} \\ x_{3,t} \end{bmatrix} + \boldsymbol{r}_t$, in Gaussian noise, the advantage of the nested scheme becomes larger, as shown in Fig. 3(b) and (d).

Next, for the same simulation set-up of Fig. 3(b) and (d), we compare the performance of the UKF-EKF nested filter (Algorithm 3) with one of the nested hybrid filters in [1]. The latter method consists of a SMC filter with $N = 120$ particles for the first layer and a bank of EKFs for the second layer. Fig. 4(a) and (b) show the $\mathrm{NMSE}_{\theta,t}$ and the $\mathrm{NMSE}_{x,t}$ respectively, for both the SMC-EKF (violet line) and the UKF-EKF (yellow line) methods. Although the time of convergence of the SMC-EKF scheme can be reduced, the UKF-EKF algorithm converges clearly faster. Also, once it converges, the estimation error for both parameters and states is slightly lower for the UKF-EKF method. However, the greatest improvement is related to the computational cost. For this experiment the UKF-EKF algorithm is three times faster (4.5 min runtime versus 14.8) than the SMC-EKF scheme. Therefore, it considerably reduces the computational cost while obtaining similar or slightly better results in estimation error.

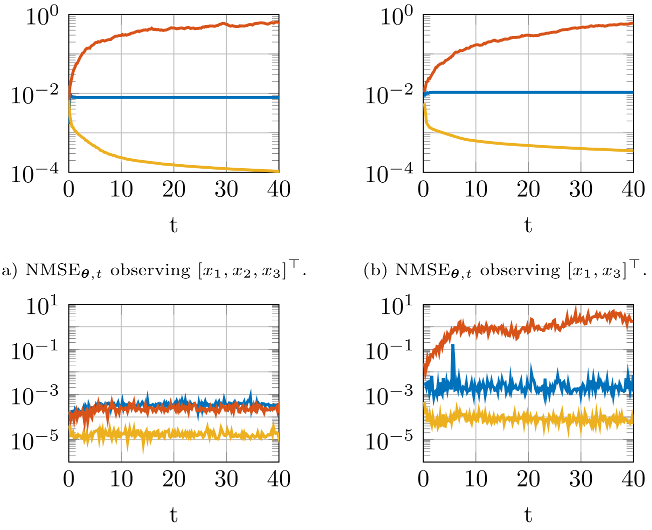
For the next computer experiment, Fig. 5 shows the parameter estimates obtained by running 50 independent simulations of the proposed UKF-EKF nested filter. The three dimensions of $\hat{\boldsymbol{\theta}}_t$ are displayed over time (Fig. 5(a)– (c)) in order to illustrate how they converge as observations are collected. Although the length of the simulations is $T = 40$ continuous-time units, we have plotted just the intervals of time where the estimates converge. The interval varies from one plot to another because the time of convergence is not the same for all parameters (having shorter times for B and longer times for S). In spite of that, this figure shows how all pa-

[4] The algorithms have been coded in MATLAB R2017a and run on a computer with 128 GB of DRAM and equipped with two Intel Xeon Gold 5115 10-Core CPU processors (running at 2.40 GHz).



(a) NMSE$_{\boldsymbol{\theta}}$ for different values of λ. (b) NMSE$_x$ for different values of λ. (c) Average simulation run-time.

Fig. 2. The average NMSE and average time of simulation in minutes over 80 simulation runs for different values of λ, using the Euclidean norm $\|\cdot\|_2$ (in blue) and the maximum norm $\|\cdot\|_\infty$ (in red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

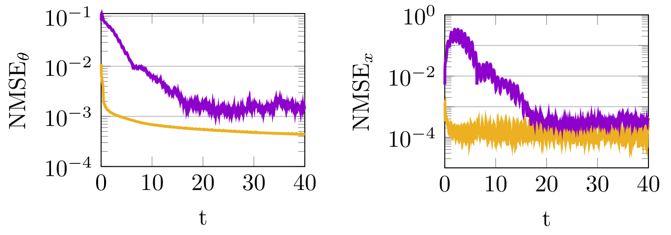


(a) NMSE$_{\boldsymbol{\theta},t}$ observing $[x_1, x_2, x_3]^\top$. (b) NMSE$_{\boldsymbol{\theta},t}$ observing $[x_1, x_3]^\top$.

(c) NMSE$_{x,t}$ observing $[x_1, x_2, x_3]^\top$. (d) NMSE$_{x,t}$ observing $[x_1, x_3]^\top$.

Fig. 3. Performance of UKF (red), EnKF (blue) and UKF-EKFs (yellow) for two different setups, averaged over 50 independent simulation runs. Figures 3(a) and 3(c) show NMSE$_{\theta,t}$ and NMSE$_{x,t}$ respectively, where the whole state vector is observed. In figures 3(b) and 3(d), the error is plotted for a setup where only the first and third components of the state (x_1 and x_3) are observed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

rameters converge to the true values for different initializations. In the same vein, Fig. 6 illustrates the evolution of the three entries ($\sigma^2_{S,t}$,$\sigma^2_{R,t}$ and $\sigma^2_{B,t}$) on the diagonal of the covariance matrix $\hat{\boldsymbol{C}}^{\boldsymbol{\theta}}_t$ over time (the plots are an average of 50 independent simulation runs). The same as the mean values shown in Fig. 5, the estimated variances of the parameters S, R and B stabilize over time at similar rates, albeit with different steady-state values.



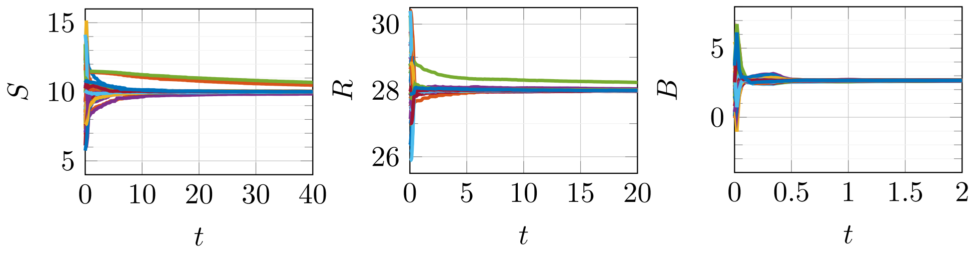
(a) Averaged NMSE$_{\boldsymbol{\theta},t}$. (b) Averaged NMSE$_{x,t}$.

Fig. 4. Performance of a SMC-EKF (violet) and a UKF-EKF (yellow) averaged over 100 simulation runs. The SMC scheme propagates $N = 120$ samples over time. Figure 4(a) shows NMSE$_{\theta,t}$ and figure 4(b) shows NMSE$_{x,t}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 7 (a), on the other hand, illustrates the accuracy of state estimates, $\hat{\boldsymbol{x}}_t$, by averaging the NMSE$_{x,t}$ obtained for the same set of 50 simulation runs as in Fig. 5. The error NMSE$_{x,t}$ decreases with time as the parameter estimates get closer to their true values, being its value stabilized around $t = 5$. By that time, all parameter estimates in Fig. 5 have already converged (or at least got closer to their steady values) and, consequently, the state estimates become reliable.

In Fig. 7(b)–(d), the estimated marginal pdfs of each element in $\hat{\boldsymbol{\theta}}_t$ at time $t = 40$ are plotted for a typical simulation run. These plot illustrates the uncertainty associated to each parameter. The means of these Gaussian pdfs are close to the true parameters, in agreement with results seen in Fig. 5. In addition, the variances are small, being all the probability distributions tightly packed around the ground truth.

Fig. 8 displays the average performance of the UKF-EKF nested filter for different observation noise variances, σ^2_y. Although all the previous experiments are done with $\sigma^2_y = 1$, in Fig. 8(a) we obtain similar results of NMSE$_\theta$ for $\sigma^2_y = 2$ and slightly worse errors for $\sigma^2_y = 4$ and $\sigma^2_y = 10$. Although the errors increase for values of σ^2_y



(a) Estimates $\hat{\theta}_{1,t} = \hat{S}_t$. (b) Estimates $\hat{\theta}_{2,t} = \hat{R}_t$. (c) Estimates $\hat{\theta}_{3,t} = \hat{B}_t$.

Fig. 5. Sequences of posterior-mean estimates, $\hat{\boldsymbol{\theta}}_t$, over time obtained from 50 independent simulation runs.

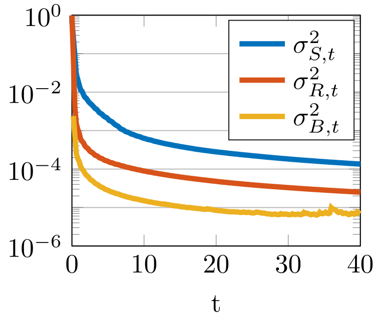
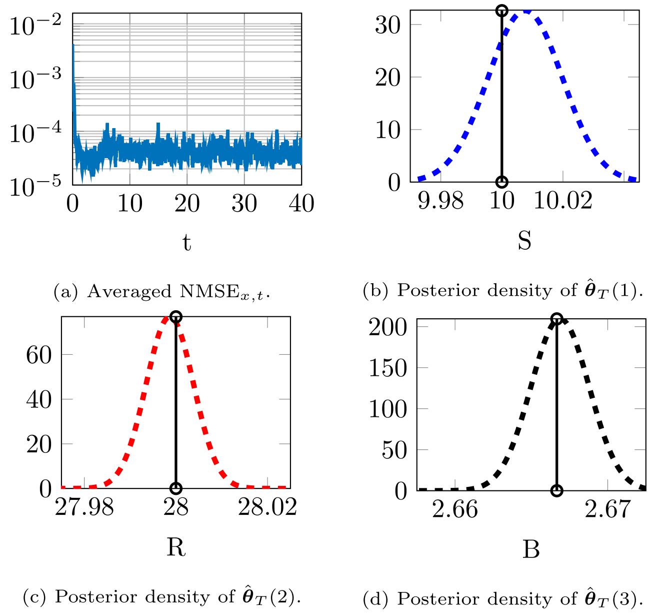


Fig. 6. Values on the diagonal of the posterior covariance matrix, $\hat{C}^{\theta}_t$, over time. The plot is averaged over 50 independent simulation runs.



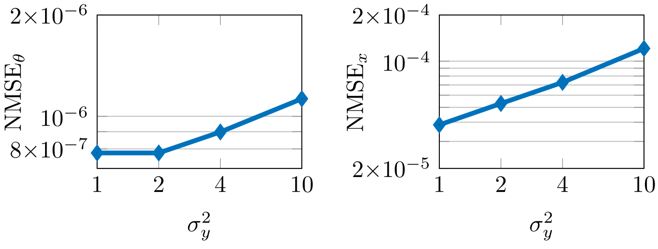
(a) Averaged NMSE$_{x,t}$.

(b) Posterior density of $\hat{\boldsymbol{\theta}}_T(1)$.

(c) Posterior density of $\hat{\boldsymbol{\theta}}_T(2)$.

(d) Posterior density of $\hat{\boldsymbol{\theta}}_T(3)$.

Fig. 7. The mean NMSE$_{x,t}$ of 50 simulation runs over time is plotted in Fig. 7(a). Fig. 7(b)–(d) show the posterior density of parameters (dashed lines) at time $t = T$ and their true values (black vertical lines).



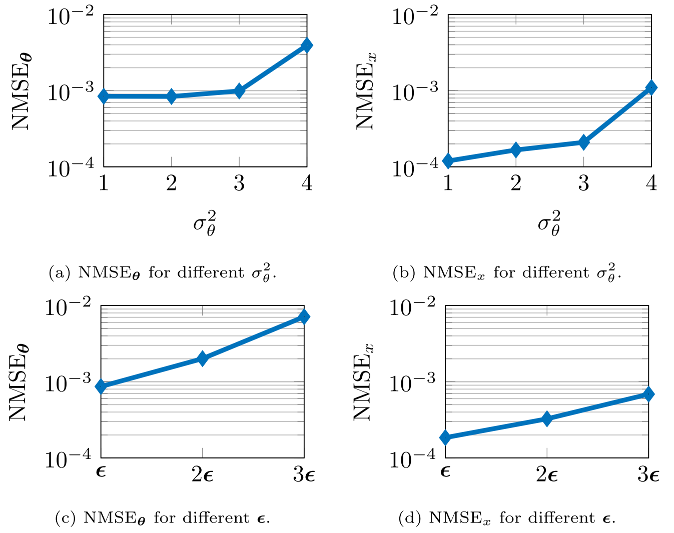
(a) NMSE$_\theta$ for different σ^2_y.

(b) NMSE$_x$ for different σ^2_y.

Fig. 8. NMSE$_\theta$ (Fig. 8(a)) and NMSE$_x$ (Fig. 8(b)) of UKF-EKF, averaged over 50 simulation runs, for different values of the noise variance σ^2_y.

greater than one, the general performance of the algorithm is still accurate for larger values of the variance in the observation noise.

Fig. 9 illustrates the average performance of the proposed nested filter (UKF-EKF) in terms of NMSE$_{\boldsymbol{\theta}}$ and NMSE$_x$ given different prior distributions of the form $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_\theta, \sigma^2_\theta \boldsymbol{I}_3)$. In Fig. 9(a) and (b) we depict the NMSE$_{\boldsymbol{\theta}}$ and NMSE$_x$, respectively, as the prior variance σ^2_θ is increased and the a priori mean $\boldsymbol{\mu}_\theta$ is drawn at random from a uniform distribution $\mathcal{U}(\boldsymbol{\theta}_\star - \boldsymbol{\epsilon}, \boldsymbol{\theta}_\star + \boldsymbol{\epsilon})$ for each independent simulation, where $\boldsymbol{\epsilon} = [3, 1, 0.5]^\top$. Changes in the variance of the prior distribution $p(\boldsymbol{\theta})$ slightly degrade the performance of the nested filter in terms of NMSE$_{\boldsymbol{\theta}}$ and NMSE$_x$, obtaining comparable results for values of σ^2_θ from 1 to 3. Similarly, in Fig. 9(c) and (d) the variance is fixed ($\sigma^2_\theta = 1$) but the priori mean $\boldsymbol{\mu}_\theta$ is drawn from a uniform distribution of the form $\mathcal{U}(\boldsymbol{\theta}_\star - k\boldsymbol{\epsilon}, \boldsymbol{\theta}_\star + k\boldsymbol{\epsilon})$ with increasing values of k, while keeping $\boldsymbol{\epsilon} = [3, 1, 0.5]^\top$. The NMSEs obtained in this way increase approximately linearly with k, to yield an NMSE$_{\boldsymbol{\theta}}$ of almost one order of magnitude higher for $k = 3$ (but still below 10^{-2}).



(a) NMSE$_{\boldsymbol{\theta}}$ for different σ^2_θ.

(b) NMSE$_x$ for different σ^2_θ.

(c) NMSE$_{\boldsymbol{\theta}}$ for different $\boldsymbol{\epsilon}$.

(d) NMSE$_x$ for different $\boldsymbol{\epsilon}$.

Fig. 9. NMSE$_{\boldsymbol{\theta}}$ (Fig. 9(a)) and NMSE$_x$ (Fig. 9(b)) of UKF-EKF, averaged over 60 simulation runs, for different values of the σ^2_θ. Also, NMSE$_{\boldsymbol{\theta}}$ (Fig. 9(c)) and NMSE$_x$ (Fig. 9(d)) for greater values of the $\boldsymbol{\epsilon}$ and fixed variance (σ^2_θ).

5. Example: a stochastic volatility model with real data

In this section, we assess the performance of the proposed algorithm (UKF-EKF) for the task of estimating the parameters of a stochastic-volatility model using real-world time-series data (namely, euro-USD exchange rates between December 2014 and December 2016).

5.1. Stochastic volatility model

We assume the stochastic volatility state-space model [36]

$$x_0 = \mu + \sqrt{\frac{\sigma_v^2}{1-\phi^2}}v_0, \tag{44}$$

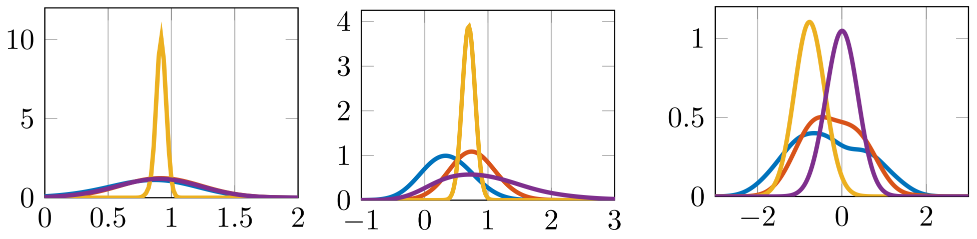
$$x_t = \mu + \phi(x_{t-1} - \mu) + \sigma_v v_t, \tag{45}$$

$$y_t = x_t + \sqrt{\omega} r_t, \tag{46}$$

where $\boldsymbol{\theta} = [\mu, \sigma_v^2, \phi]^\top \in \mathbb{R} \times \mathbb{R}_+ \times [-1, 1]$ is the vector of static unknown parameters and the state x_t represents the log-volatility of the time-series of observations $y_t, t = 0, \dots, T$. The variables v_t's and r_t's are iid Gaussian random variables and the variance of the observation noise $\omega = \frac{\pi^2}{2}$ is assumed known.

Given the historical EUR-USD exchange rate data sequence $s_0, \dots, s_T$ from 2014-12-31 to 2016-12-31 (obtained from www.quandl.com), we generate the time series of log-returns y_t following the same procedure as in [37–39]. To be specific, at time t we compute

$$\bar{y}_t = 100 \log\left(\frac{s_t}{s_{t-1}}\right), \tag{47}$$



(a) Averaged pdfs of ϕ. (b) Averaged pdfs of σ_v^2. (c) Averaged pdfs of μ.

Fig. 10. Pdfs of the unknown parameters $\boldsymbol{\theta} = (\phi, \sigma_v^2, \mu)^\top$ at time $T = 513$, averaged over 40 simulation runs. We have assessed four different algorithms: the NPF (blue), the SMC-EKF (red), the EnKF (violet) and the UKF-EKF (yellow). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

for $1 \leq t \leq T$. Then, we further transform the log-returns into

$$y_t = \log(\bar{y}_t^2) + 1.27, \tag{48}$$

in order to obtain the observations as described in (46).

5.2. Simulation setup

Similar to Section 4, we have assessed several algorithms for the estimation of the parameters $\boldsymbol{\theta} = [\mu, \sigma_v^2, \phi]^\top$ and the state x_t of the stochastic volatility model. In particular, we have implemented the following methods:

- The proposed Algorithm 3, using a UKF and a bank of EKFs in the first and second layers of the filter, respectively.
- An EnKF algorithm [34] with state augmentation (running $N = 100$ ensembles).
- An NHF [1] using a SMC scheme in the first layer (with $N = 100$ particles) and a bank of EKFs in the second layer.
- An NPF [25] with $N = 100$ particles in the SMC scheme of each layer (i.e., 10,000 particles overall).

We recall that the computational complexity of these algorithms is outlined in Table 1.

For the UKF-EKF scheme we assume a Gaussian prior distribution for the unknown parameters, namely $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\boldsymbol{\theta}}, 10^{-2}\boldsymbol{I}_3)$, where the a priori mean is drawn at random from a normal distribution $\boldsymbol{\mu}_{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}|[0.85, 0.05, 0]^\top, 10^{-2}\boldsymbol{I}_3)$. For the rest of the methods, the prior distribution for the unknown parameters is $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|[0.9, 0.2, 0]^\top, 10^{-1}\boldsymbol{I}_3)$. The prior distribution for the state variable x_t is given conditional on the parameters, namely $p(x_0) = \mathcal{N}\left(x_0|\mu, \frac{\sigma_v^2}{1-\phi^2}\right)$ for all the different methods used in this section.

We have assessed the performance of the algorithms in terms of marginal log-likelihood (MLL) estimated for the model, since there is no available ground truth for either the state or the parameters. We also illustrate the performance of the algorithms with the marginal pdfs of the three parameters at the final discrete time step T. For all methods, the marginal densities are averaged over 40 simulation runs.

5.3. Numerical results

Fig. 10 shows the averaged marginal pdfs of the parameters at the end of the simulation runs (at discrete time $T = 513$) for the different algorithms compared here (NPF in blue, SMC-EKF in red, the proposed UKF-EKF in yellow and EnKF in violet). In general, all the algorithms obtain pdfs that overlap significantly, showing that the estimation task yields coherent similar results independently of the method applied. However, it is seen from Fig. 10(c) that the NPF and NHF methods yield pdfs which are non-Gaussian. Moreover, the estimates of the marginal pdfs output by the UKF-EKF method are narrower than the densities estimated by the other methods. This suggests that the UKF-EKF scheme may yield rather accurate point estimates for the parameters but possibly underestimate their variance.

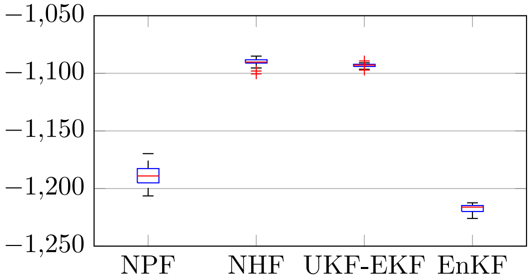


Fig. 11. Marginal log-likelihood at time T, obtained from 40 simulation runs, for the NPF, the SMC-EKF, the UKF-EKF and the EnKF with state augmentation.

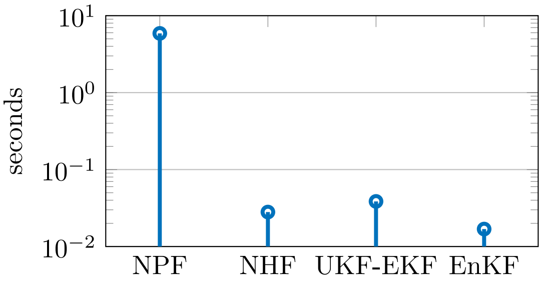


Fig. 12. Running time in seconds, averaged over 40 simulation runs, for the NPF, the SMC-EKF, the UKF-EKF and the EnKF with state augmentation.

Fig. 11 shows the marginal log-likelihood (MLL) as estimated using the same four methods. The smaller estimated variance for the parameters of the UKF-EKF algorithms reflects on a higher confidence on the model (i.e., a larger MLL). As for the dispersion on the MLL estimates, we should note that the UKF-EKF scheme is a deterministic method, hence the only source of variance is the randomness in the prior that we have introduced for the simulations. All other algorithms are stochastic and so display a larger variability of the estimates.

Finally, Fig. 12 illustrates the run-times averaged over 40 simulation runs. The use of Gaussian filters reduces considerably the computational cost, since the NPF is 2 orders of magnitude slower than any of the other algorithms. These run-times have been obtained running MATLAB R2018 on a MacBook Pro laptop computer with a 2,3 GHz Dual-Core Intel Core i5processor and 16 GB 2133 MHz LPDDR3 of RAM.

6. Conclusions

We have introduced a generalization of the NHF methodology of [1] that, using long sequences of observations collected over time, estimates the static parameters and tracks the stochastic dynamical variables of a state space model. This scheme combines two layers of filters, one inside the other, in order to compute the joint posterior probability distribution of the parameters and the states. In this generalization of the methodology, we introduce the use of deterministic sampling techniques in the first layer of the algorithm (the cubature Kalman filter (CKF) or the unscented Kalman filter (UKF)), instead of Monte Carlo methods, describing in detail how the algorithms can work sequentially and recursively. We have presented numerical results for a stochastic Lorenz 63 model with synthetic data and for a stochastic volatility model with real-world data, using a scheme with an UKF for the parameters in the first layer, and EKFs for the time-varying state variables in the second layer. We have introduced and assessed the values of a relative threshold that enables the algorithm to work recursively, and we have evaluated the performance of the algorithm in terms of the normalized mean square errors for the parameters and the dynamic state variables. We have also compared these results with other algorithms, such as the ensemble Kalman filter (EnKF) or the unscented Kalman filter (UKF), that implement state augmentation (i.e., an extended state that includes both parameters and state), and also with an NHF (with a SMC in the first layer and EKFs in the second layer) and an NPF. The use of Gaussian filters in the two layers of the algorithm not only leads to a significant reduction in computational complexity compared to Monte Carlo-based implementations but also increases the accuracy compared to state-augmented Gaussian filters, as shown by the computer simulations of Section 4.3.

Additional research is still needed for an analytical characterization of the performance, possibly under suitable regularity conditions that enable a theoretical study. We also expect the proposed multi-layer structure to be advantageous for nonlinear tracking and forecasting in state-space systems that display multi-scale dynamics, i.e., where different subsets of state variables evolve at different rates (over significantly different time scales). The computation of marginal and conditional posterior distributions at different layers of computation that has been implemented in the current paper to estimate static parameters and dynamic variables appears as a natural approach to tackle the estimation/prediction of state variables evolving on different time scales.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Sara Pérez-Vieites: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Joaquín Míguez:** Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition.

Acknowledgment

This research was partially supported by the Office of Naval Research (award no. N0014-19-1-2226), Agencia Estatal de Investigación of Spain (RTI2018-099655-BI00 CLARA) and the regional government of Madrid (project no. Y2018/TCS-4705 PRACTICO).

Appendix A. Proof of Proposition 1

Proof. We proceed by induction in the time index t. For $t=0$ we have $\pi_0(\boldsymbol{x}_0|\boldsymbol{\theta}) = p(\boldsymbol{x}_0)$ independently of $\boldsymbol{\theta}$, hence for any pair $(\boldsymbol{\theta}, \boldsymbol{\theta}') \in \Theta \times \Theta$ we obtain

$$\int |\pi_0\big(\boldsymbol{x}_0|\boldsymbol{\theta}\big) - \pi_0\big(\boldsymbol{x}_0|\boldsymbol{\theta}^{'}\big)|d\boldsymbol{x}_0 = \int |p(\boldsymbol{x}_0) - p(\boldsymbol{x}_0)| = L_0\|\boldsymbol{\theta} - \boldsymbol{\theta}^{'}\|$$

for $L_0 = 0$.

For the induction step, assume that

$$\int \left|\pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}) - \pi_{t-1}(\boldsymbol{x}_{t-1}\middle|\boldsymbol{\theta}')\right|d\boldsymbol{x}_{t-1} < L_{t-1}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \tag{A.1}$$

for some $L_{t-1} < \infty$. Straightforward calculations yield

$$\begin{aligned}
&\int \left|\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - \xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\right|d\boldsymbol{x}_t = \\
&\quad = \int \bigg| \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})\pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta})d\boldsymbol{x}_{t-1} \\
&\quad - \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}')\pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}')d\boldsymbol{x}_{t-1} \\
&\quad \pm \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})\pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}')d\boldsymbol{x}_{t-1}\bigg|d\boldsymbol{x}_t \\
&\quad \le \int\int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})\left|\pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}) - \pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}')\right|d\boldsymbol{x}_{t-1}d\boldsymbol{x}_t \\
&\quad + \int\int \left|p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) - p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}')\right|\pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}')d\boldsymbol{x}_{t-1}d\boldsymbol{x}_t
\end{aligned}$$

and reordering the integrals we obtain

$$\begin{aligned}
&\int \left|\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - \xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\right|d\boldsymbol{x}_t \le \\
&\quad \le \int \left[\int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})d\boldsymbol{x}_t\right]\left|\pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}) - \pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}')\right|d\boldsymbol{x}_{t-1} \\
&+ \int \left[\int \left|p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) - p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}')\right|d\boldsymbol{x}_t\right]\pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}')d\boldsymbol{x}_{t-1}
\end{aligned} \tag{A.2}$$

However, $\int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta})d\boldsymbol{x}_t = 1$ for any $\boldsymbol{x}_{t-1}$ and any $\boldsymbol{\theta}$, while Assumption 2 yields $\int |p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) - p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}')|d\boldsymbol{x}_t \le L\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$. Therefore, (A.2) becomes

$$\begin{aligned}
\int \left|\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - \xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\right|d\boldsymbol{x}_t &\le \int \left|\pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}) - \pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}')\right|d\boldsymbol{x}_{t-1} \\
&\quad + L\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \int \pi_{t-1}(\boldsymbol{x}_{t-1}|\boldsymbol{\theta}')d\boldsymbol{x}_{t-1} \\
&\quad \le (L_{t-1} + L)\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|
\end{aligned} \tag{A.3}$$

where the second inequality follows from the induction hypothesis (A.1).

As for the difference between $\pi_t(\cdot|\boldsymbol{\theta})$ and $\pi_t(\cdot|\boldsymbol{\theta}')$, the Bayes theorem readily yields

$$\begin{aligned}
&\int \left|\pi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - \pi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\right|d\boldsymbol{x}_t \\
&\quad = \int \left|\frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta})}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})} - \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}')\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta}')}\right|d\boldsymbol{x}_t
\end{aligned} \tag{A.4}$$

and the absolute difference in the integrand of (A.4) can be rewritten as

$$\begin{aligned}
&\left|\frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta})}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})} - \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}')\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta}')}\right| \\
&\quad = \bigg|\frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta})}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})} \pm \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}')\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})} \\
&\quad - \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}')\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta}')}\bigg|
\end{aligned}$$

$$= \left| \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}')\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})} + \pi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\frac{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta}') - \eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})} \right| \tag{A.5}$$

where we have used the relationship $\pi_t(\boldsymbol{x}_t|\theta') = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\theta}')\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta})}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta}')}$ to obtain the second identity. Now, if we substitute (A.5) into (A.4) and then realize that

$$\left|\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta}) - \eta_t(\boldsymbol{y}_t|\boldsymbol{\theta}')\right| \leq \int \left|p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}')\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\right| d\boldsymbol{x}_t$$

and $\int \pi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')d\boldsymbol{x}_t = 1$, we obtain the upper bounds

$$\int \left|\pi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - \pi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\right| d\boldsymbol{x}_t \leq \frac{2}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})} \int \left|p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}')\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\right| d\boldsymbol{x}_t \tag{A.6}$$

$$\leq \frac{2}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})} \left[\int p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}) \left|\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - \xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\right| d\boldsymbol{x}_t + \int \left|p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}) - p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}')\right| \xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')d\boldsymbol{x}_t \right] \tag{A.7}$$

where (A.7) is obtained by applying a triangular inequality in (A.6).

The first integral in (A.7) can be bounded using Assumption 4 and inequality (A.3), which together yield,

$$\int \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})} \left|\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - \xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\right| d\boldsymbol{x}_t \leq M_t \int \left|\xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - \xi_t(\boldsymbol{x}_t|\boldsymbol{\theta})\right| d\boldsymbol{x}_t \leq M_t(L_{t-1} + L)\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|, \tag{A.8}$$

while the second integral can be bounded using Assumption 3, which leads to

$$\frac{2}{\eta_t(\boldsymbol{y}_t|\boldsymbol{\theta})} \int \left|p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}) - p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}')\right| \xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')d\boldsymbol{x}_t \leq 2G_t\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \int \xi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')d\boldsymbol{x}_t = 2G_t\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|. \tag{A.9}$$

Plugging (A.8) and (A.9) into (A.7) yields

$$\int \left|\pi_t(\boldsymbol{x}_t|\boldsymbol{\theta}) - \pi_t(\boldsymbol{x}_t|\boldsymbol{\theta}')\right| \leq M_t(L_{t-1} + L)\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| + 2G_t\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \leq L_t\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$$

with $L_t = M_t(L_{t-1} + L) + 2G_t < \infty$. □

Appendix B. The unscented transform (UT)

The unscented Kalman filter (UKF) is a Gaussian filter that uses a deterministic sampling technique known as the unscented transform (UT) in order to approximate the predicted and corrected posterior pdfs. The UT is a mathematical device that can be used to estimate the result of applying a nonlinear transformation to a random vector that is characterised by a set of weighted reference points (called sigma-points). The UT can be used to approximate integrals of a function $f(\boldsymbol{\theta})$ with respect to a Gaussian distribution, that takes the form

$$\int f(\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$$

where $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}, \mathbf{C}), \boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$ and the first and second order moments of $\boldsymbol{\theta}$ can be characterised by a collection of (deterministic) sigma-points with associated weights, $\{\boldsymbol{\theta}^i, w^i\}_{0 \leq i \leq M-1}$, for $M = 2d_\theta + 1$. This yields to the approximation

$$\int f(\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} \approx \sum_{i=0}^{M-1} w^i \boldsymbol{\theta}^i f(\boldsymbol{\theta}) \tag{B.1}$$

for any integrable test function $f(\cdot)$.

There are different ways in which the weighted sigma-points can be selected (see [40] for a survey). In this paper, the set $\{\boldsymbol{\theta}^i, w^i\}_{0 \leq i \leq M-1}$ is computed as

$$\boldsymbol{\theta}^0 = \hat{\boldsymbol{\theta}}, \qquad w^0 = \frac{\kappa}{\kappa + d_\theta}, \tag{B.2}$$

$$\boldsymbol{\theta}^j = \hat{\boldsymbol{\theta}} + (\sqrt{(d_\theta + \kappa)\mathbf{C}})_j, \qquad w^j = \frac{1}{2(d_\theta + \kappa)}, \tag{B.3}$$

$$\boldsymbol{\theta}^{j+d_\theta} = \hat{\boldsymbol{\theta}} + (\sqrt{(d_\theta + \kappa)\mathbf{C}})_j, \qquad w^{j+d_\theta} = \frac{1}{2(d_\theta + \kappa)}, \tag{B.4}$$

where $\kappa = 1, j = 1, \ldots, d_\theta$, and $(\sqrt{(d_\theta + \kappa)\mathbf{C}})_j$ is the jth column of the matrix square-root of $(d_\theta + \kappa)\mathbf{C}$.

References

[1] J.M.S. Pérez-Vieites, I.P. Mariño, Probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems, Phys. Rev. E 98 (6) (2017) 063305.
[2] B.D.O. Anderson, J.B. Moore, Optimal Filtering, Englewood Cliffs, 1979.
[3] N. Gordon, D. Salmond, A.F.M. Smith, Novel approach to nonlinear and non–Gaussian Bayesian state estimation, IEE Proc.-F 140 (2) (1993) 107–113.
[4] A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo Sampling methods for Bayesian filtering, Stat. Comput. 10 (3) (2000) 197–208.
[5] J.U.S.J. Julier, H.F. Durrant-Whyte, A new method for the non linear transformation of means and covariances in filters and estimators, IEEE Trans. Autom. Control 3 (2000) 477–482.
[6] P.M. Djurić, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, J. Míguez, Particle filtering, IEEE Signal Process Mag. 20 (5) (2003) 19–38.
[7] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House, Boston, 2004.
[8] E. Lourens, E. Reynders, G. De Roeck, G. Degrande, G. Lombaert, An augmented Kalman filter for force identification in structural dynamics, Mech. Syst. Signal Process. 27 (2012) 446–460.
[9] I. Hassanzadeh, A. Mehdi, Design of augmented extended and unscented Kalman filters, J. Appl. Sci. 8 (16) (2008) 2901–2906.
[10] J.S. Liu, R. Chen, T. Logvinenko, A theoretical framework for sequential importance sampling with resampling, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer, 2001, pp. 225–246. Ch. 11
[11] C. Andrieu, A. Doucet, S.S. Singh, V.B. Tadić, Particle methods for change detection, system identification and control, Proc. IEEE 92 (3) (2004) 423–438.
[12] G. Kitagawa, A self-organizing state-space model, J. Am. Stat. Assoc. (1998) 1203–1215.
[13] H. Zhang, H.-J. Franssen, X. Han, J.A. Vrugt, H. Vereecken, State and parameter estimation of two land surface models using the ensemble Kalman filter and the particle filter, Hydrol. Earth Syst. Sci. 21 (9) (2017) 4927–4958.
[14] J. Liu, M. West, Combined parameter and state estimation in simulation-based filtering, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer, 2001, pp. 197–223. Ch. 10
[15] C.M. Carvalho, M.S. Johannes, H.F. Lopes, N.G. Polson, Particle learning and smoothing, Stat. Sci. 25 (1) (2010) 88–106.
[16] G. Storvik, Particle filters for state-space models with the presence of unknown static parameters, IEEE Trans. Signal Process. 50 (2) (2002) 281–289.
[17] P.M. Djurić, J. Míguez, Sequential particle filtering in the presence of additive Gaussian noise with unknown parameters, in: Proceedings of ICASSP, 2002.
[18] C. Nemeth, P. Fearnhead, L. Mihaylova, Sequential Monte Carlo methods for state and parameter estimation in abruptly changing environments, IEEE Trans. Signal Process. 62 (5) (2013) 1245–1255.
[19] N. Kantas, A. Doucet, S.S. Singh, J.M. Maciejowski, N. Chopin, On particle methods for parameter estimation in state-space models, Stat. Sci. 30 (2015) 328–351.
[20] C. Andrieu, A. Doucet, V.B. Tadić, One-line parameter estimation in general state-space models using a pseudo-likelihood approach, IFAC Proc. Vol. 45 (16) (2012) 500–505.
[21] V.B. Tadic, Analyticity, convergence, and convergence rate of recursive maximum-likelihood estimation in hidden Markov models, IEEE Trans. Inf. Theory 56 (12) (2010) 6406–6432.
[22] F. Ding, State filtering and parameter estimation for state space systems with scarce measurements, Signal Process. 104 (2014) 369–380.
[23] N. Chopin, P.E. Jacob, O. Papaspiliopoulos, SMC^2: an efficient algorithm for sequential analysis of state space models, J. R. Stat. Soc. Ser. B 75 (3) (2013) 397–426.

[24] C. Andrieu, A. Doucet, R. Holenstein, Particle Markov chain Monte Carlo methods, J. R. Stat. Soc. B 72 (2010) 269–342.

[25] D. Crisan, J. Miguez, et al., Nested particle filters for online parameter estimation in discrete-time state-space Markov models, Bernoulli 24 (4A) (2018) 3039–3086.

[26] J. Kokkala, S. Särkkä, Combining particle MCMC with Rao-Blackwellized Monte Carlo data association for parameter estimation in multiple target tracking, Digit. Signal Process. 47 (2015) 84–95.

[27] F. Lu, N. Weitzel, A.H. Monahan, Joint state-parameter estimation of a nonlinear stochastic energy balance model from sparse noisy data, Nonlinear Process. Geophys. 26 (3) (2019) 227–250.

[28] S. Schlupkothen, T. Heidenblut, G. Ascheid, Random field-aided tracking of autonomous kinetically passive wireless agents, EURASIP J. Adv. Signal Process. 2020 (1) (2020) 1–27.

[29] R.M. Vieira, Bayesian online state and parameter estimation for streaming data, Newcastle University, 2018 Ph.d. thesis.

[30] X. Yang, R. Chatpatanasiri, P. Sattayatham, Value at risk estimation under stochastic volatility models using adaptive PMCMC methods, Commun. Stat.-Simul.Comput. 46 (9) (2017) 7221–7237.

[31] D. Crisan, J. Miguez, Uniform convergence over time of a nested particle filtering scheme for recursive parameter estimation in state–space Markov models, Adv. Appl. Probab. 49 (4) (2017) 1170–1200.

[32] M. Gerber, N. Chopin, Sequential quasi Monte Carlo, J. R. Stat. Soc. Ser. B 77 (3) (2015) 509–579.

[33] I. Arasaratnam, S. Haykin, Cubature Kalman filters, IEEE Trans. Automat. Contr. 54 (6) (2009) 1254–1269.

[34] G. Evensen, The ensemble Kalman filter: theoretical formulation and practical implementation, Ocean Dyn. 53 (4) (2003) 343–367.

[35] S.J. Julier, J. Uhlmann, Unscented filtering and nonlinear estimation, Proc. IEEE 92 (2) (2004) 401–422.

[36] S. Kim, N. Shephard, S. Chib, Stochastic volatility: likelihood inference and comparison with ARCH models, Rev Econ Stud 65 (3) (1998) 361–393.

[37] J. Dahlin, T.B. Schön, Getting started with particle Metropolis-Hastings for inference in nonlinear dynamical models, 2015, arXiv preprint arXiv:1511.01707.

[38] O.D. Akyildiz, J. Míguez, Nudging the particle filter, Stat. Comput. 30 (2) (2020) 305–330.

[39] R.S. Tsay, Analysis of Financial Time Series, vol. 543, John Wiley & Sons, 2005.

[40] H.M. Menegaz, J.Y. Ishihara, G.A. Borges, A.N. Vargas, A systematization of the unscented Kalman filter theory, IEEE Trans. Automat. Contr. 60 (10) (2015) 2583–2598.