ADAPTIVE GAUSSIAN NESTED FILTER FOR PARAMETER ESTIMATION AND STATE TRACKING IN DYNAMICAL SYSTEMS

Sara Pérez-Vieites[⋆] *Víctor Elvira*[†]

[⋆] IMT Nord Europe, Institut Mines-Télécom, Centre for Digital Systems, France
[†] School of Mathematics, University of Edinburgh, United Kingdom

ABSTRACT

We introduce the adaptive Gaussian nested filter (AGNesF), the first nested method that adapts the number of samples to estimate both the static parameters and the dynamical variables of a state-space model. The proposed method is based on the nested Gaussian filter (NGF), that combines two layers of inference, one inside the other, to compute the joint posterior probability distribution of the static parameters and the state variables. We propose two novel rules to reduce computational complexity without compromising the performance. One enables the bottom layer techniques to run recursively, while the other reduces automatically the number of samples in the parameter space when they are redundant. We describe a specific implementation of the new scheme that uses a quadrature Kalman filter (QKF) in the parameter layer, and we study its performance in a stochastic Lorenz 63 model.

Index Terms— Bayesian inference, stochastic filtering, Gaussian filter, parameter estimation, adaptive filter

1. INTRODUCTION

The estimation of the evolution of dynamical systems over time is a crucial problem in many fields of science. Many examples can be found in meteorology [1], ecology [2], epidemiology [3], quantitative finance [4] and engineering [5]. These systems are often characterized by state-space models (SSMs) that consist of a sequence of state vectors, $\boldsymbol{x}_t$, a sequence of noisy observation vectors, $\boldsymbol{y}_t$, and a set of static model parameters, $\boldsymbol{\theta}$, that describe the behavior of the state. Classical filtering methods [6, 7, 8], including both Gaussian-approximation techniques and Monte Carlo schemes, address the estimation of the states $\boldsymbol{x}_t$ while assuming the parameters $\boldsymbol{\theta}$ to be known. However, this is hardly ever the case in practice, and the parameters need to be estimated as well.

In the last few years, there have been advances in the development of well-principled probabilistic methods that compute the posterior probability distribution of both the parameters and the state variables. Among these methods, the sequential Monte Carlo square (SMC^2) [9], the particle Markov chain Monte Carlo (PMCMC) [10], and the nested particle filter (NPF) [11] stand out, all of them based on particle filters [12, 13, 14]. The NPF is the only method that is recursive. This scheme is characterized by two intertwined layers of approximation methods, the top layer for the parameter estimation task and the bottom layer for the state tracking. However, the use of Monte Carlo in both layers is still computationally costly, due to the high number of samples or particles needed. In order to alleviate the computational cost, the nested hybrid filter (NHF) [15] and the NGF [16] use lower-cost filtering techniques in each layer. In particular, the NHF combines deterministic methods with Monte Carlo-based techniques, while the NGF implements deterministic techniques in both layers of inference.

A key feature of all the aforementioned methods is the selection of the number of samples or points. In general, a large number of samples entails a good performance but also a higher computational cost. Several rules to adapt/select the number of samples in Monte Carlo methods have been proposed. Some of them rely on a Kullback–Leibler divergence-based approximation error [17], a heuristic approach based on the effective sample size [18, 19], the variance of the particle estimators [20, 21], or the performance of several candidate models [22]. Also, the uniformity of a predictive statistic is used in [23, 24] to change automatically the number of samples and achieve a prescribed performance.

In this paper, we propose the adaptive Gaussian nested filter (AGNesF), a new nested method that incorporates two mechanisms to reduce the computational complexity. First, we introduce a new rule to determine when the filters of the bottom layer can run recursively. Therefore, the whole sequence of data is not reprocessed every time there is a new observation. Second, we propose a novel rule to reduce automatically the number of points when t grows. In this way, the parameter space is explored when processing the beginning of the time series, and the computational complexity is reduced when the performance is no longer compromised. Within this framework, we describe the implementation of the QKF in the top layer of the nested methodology and show numerical results for a stochastic Lorenz 63 model.

We introduce the problem and the technical background in Section 2. In Section 3, we describe the new methodology. The numerical results are shown in Section 4 and we draw the conclusions in Section 5.

S. P.-V. and V. E. acknowledge support from the *Agence Nationale de la Recherche* of France under PISCES (ANR-17-CE40-0031-01) project. The work of V. E. is also supported by the Leverhulme Research Fellowship (RF-2021-593) and by ARL/ARO under grants W911NF-20-1-0126 and W911NF-22-1-0235.

2. BACKGROUND

2.1. State-space models (SSMs)

We are interested in Markov state-space dynamical systems 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 a d_x-dimensional state vector, $\boldsymbol{y}_t \in \mathbb{R}^{d_y}$ is the d_y-dimensional noisy observation vector, and $\boldsymbol{v}_t$ and $\boldsymbol{r}_t$ are zero-mean random vectors that represent the state and observation independent noises. The evolution of the system depends on the functions $f\colon \mathbb{R}^{d_x} \times \mathbb{R}^{d_\theta} \longrightarrow \mathbb{R}^{d_x}$ and $g\colon \mathbb{R}^{d_x} \times \mathbb{R}^{d_\theta} \longrightarrow \mathbb{R}^{d_y}$, and an unknown parameter vector, $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$.

2.2. Bayesian inference in SSMs

The goal of nested methodologies is to approximate sequentially the joint posterior probability density function (pdf) of the unknown parameters, $\boldsymbol{\theta}$, and the state at time t, $\boldsymbol{x}_t$,

$$p(\boldsymbol{\theta}, \boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}). \tag{3}$$

Figure 1 shows a generic nested scheme. In the top layer (the parameter layer), the posterior pdf of the parameters $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$ is computed, while in the bottom layer (the state layer), we calculate $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})$. We assume identifiable SSMs, thus, there is a unique, true parameter $\boldsymbol{\theta}$ that maximizes the posterior $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$. Different methods (e.g., deterministic or Monte Carlo filters) can be used in each layer to approximate these pdfs, leading to a broad class of nested methods (e.g., NPF [11], NHF [15], NGF [16]).

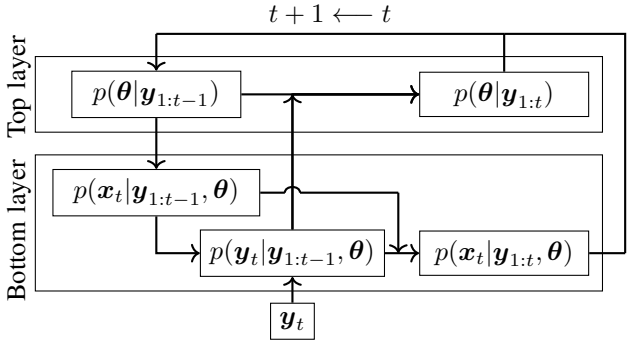

Fig. 1: Recursive scheme of the nested filters, showing the relevant pdfs and their relations.

The key difficulty of all nested methodologies is to sequentially approximate

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1}), \tag{4}$$

since the likelihood $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ cannot be evaluated exactly. However, it can be rewritten as the integral

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}) p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}) d\boldsymbol{x}_t, \tag{5}$$

where $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$ is given by the SSM, and $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ is approximated by the bottom layer. Therefore, the likelihood $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ can be evaluated in the bottom layer of inference (see Fig. 1).

2.3. Nested Gaussian filters

NGFs [16] estimate the pdf $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$ for the approximation of expectations of the form

$$\begin{aligned}\int f(\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) d\boldsymbol{\theta} &= \int f(\boldsymbol{\theta}) \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1}) d\boldsymbol{\theta} \\ &\approx \sum_{n=1}^{N_\theta} f(\boldsymbol{\theta}_t^n) \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n)}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} w_t^n,\end{aligned} \tag{6}$$

for any function of the parameters, $f(\boldsymbol{\theta})$. Eq. (6) is obtained by approximating $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$ with a set of weighted points, $\{\boldsymbol{\theta}_t^n, w_t^n\}_{n=1}^{N_\theta}$, e.g., using cubature rules [25] or the unscented transform (UT) [7]. The constants $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n)$, $n = 1, \ldots, N_\theta$, are approximated using a bank of N_θ Gaussian filters placed in the bottom layer of the nested scheme. Then, $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})$ is approximated as

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

$$\approx \sum_{n=1}^{N_\theta} p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n) w_t^n. \tag{8}$$

3. ADAPTIVE GAUSSIAN NESTED FILTER

We now introduce the adaptive Gaussian nested filter (AGNesF), a new algorithm of the family of nested methods that adapts online the number of points of the parameter layer. We implement a QKF [26] in the upper layer and, since the number of points will be adaptive, we will denote $N_{\theta,t}$ instead of N_θ. The method is summarized in Algorithm 1. First, we initialize $N_{\theta,1}$ as well as the initial state for the bottom layer filters. We also set the first mean and covariance matrix of the parameters, $\widehat{\boldsymbol{\theta}}_0$ and $\widehat{\boldsymbol{C}}_0^{\boldsymbol{\theta}}$. Then, at every time step, we generate a new set of weighted points $\{\boldsymbol{\theta}_t^n, w_t^n\}_{n=1}^{N_{\theta,t}}$, and we run the $N_{\theta,t}$ bottom filters to compute the pdfs $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n)$. For this task, we assume that the function $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ is continuous on $\boldsymbol{\theta}$ (see [16] for more details), and thus, $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n) \approx p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^n)$ when $\boldsymbol{\theta}_t^n \approx \boldsymbol{\theta}_{t-1}^n$. In line 9 and based on this assumption, we introduce a new recursive rule (described in Section 3.1) to decide whether the bottom layer filters compute $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n)$ recursively (line 9) or not (line 11). After computing all the likelihoods in the bottom layer, in line 14 we estimate both the means and covariance matrices for the parameter and the state by applying Eq. (6). Finally, we compute ρ_t in Eq. (10) to update $N_{\theta,t+1}$, as described in Section 3.2.

Algorithm 1 Adaptive Gaussian nested filter (AGNesF)

Input: $p(\boldsymbol{x}_0)$, $p(\boldsymbol{\theta})$, $\alpha_1 > 1$, $\alpha_{\text{min}} = 2$, and $\lambda, \epsilon > 0$.

1: Set $N_{\theta,1} = \alpha_1^{d_\theta}$.
2: Set $\widehat{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_0$ and $\widehat{\boldsymbol{C}}_0^{\boldsymbol{\theta}} = \boldsymbol{C}_0^{\boldsymbol{\theta}}$ from $p(\boldsymbol{\theta})$.
3: Initialise state from $p(\boldsymbol{x}_0)$ in each bottom layer filter.
4: **for** $t \geq 1$ **do**
5: Generate $\{\boldsymbol{\theta}_t^n, w_t^n\}_{n=1}^{N_{\theta,t}}$ as

$$\boldsymbol{\theta}_t^n = \sqrt{\widehat{\boldsymbol{C}}_{t-1}^{\boldsymbol{\theta}}}\xi^n + \widehat{\boldsymbol{\theta}}_{t-1} \quad \text{and} \quad w_t^n = \upsilon^n,$$

where $\{\xi^n, \upsilon^n\}_{n=1}^{N_{\theta,t}}$ is a set of quadrature points and weights for the standard normal, and $\sqrt{\boldsymbol{C}}$ represents the Cholesky factor of $\boldsymbol{C}$.
6: **for** $n = 1$ to $N_{\theta,t}$ **do**
7: Calculate the index m^n in (9).
8: **if** $\|\boldsymbol{\theta}_t^n - \boldsymbol{\theta}_{t-1}^{m^n}\|_p < \lambda \|\boldsymbol{\theta}_{t-1}^{m^n}\|_p$ **then**
9: Run filter to compute $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n)$ from $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^n) \approx p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^{m^n})$.
10: **else**
11: Run filter to compute $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n)$ from the prior $p(\boldsymbol{x}_0)$, processing the sequence $\boldsymbol{y}_{1:t}$.
12: **end if**
13: **end for**
14: Compute $\widehat{\boldsymbol{x}}_t$, $\widehat{\boldsymbol{C}}_t^{\boldsymbol{x}}$, $\widehat{\boldsymbol{\theta}}_t$ and $\widehat{\boldsymbol{C}}_t^{\boldsymbol{\theta}}$ from (6).
15: Calculate ρ_t from (10).
16: **if** $\frac{\rho_t}{N_{\theta,t}} > 1 - \epsilon$ **then**
17: $N_{\theta,t+1} = (\alpha_{t+1})^{d_\theta}$ with $\alpha_{t+1} = \max(\alpha_t - 1, \alpha_{\text{min}})$.
18: **else**
19: $N_{\theta,t+1} = N_{\theta,t}$ with $\alpha_{t+1} = \alpha_t$.
20: **end if**
21: **end for**
22: **return** $\widehat{\boldsymbol{x}}_t$, $\widehat{\boldsymbol{C}}_t^{\boldsymbol{x}}$, $\widehat{\boldsymbol{\theta}}_t$ and $\widehat{\boldsymbol{C}}_t^{\boldsymbol{\theta}}$

In the proposed method, we introduce two mechanisms to reduce the computational complexity in the nested methodology. First, we describe a new rule that enables the bottom layer filters to run recursively. Otherwise, the whole sequence $\boldsymbol{y}_{1:t}$ has to be reprocessed every time there is a new observation. Second, we propose a novel rule that, based on a statistic, reduces the number of points in the parameter estimation layer (top layer in Fig. 1) when t grows. The parameter space is properly explored at the beginning of the experiments and the computational complexity is reduced when the performance is no longer compromised. In the next two sections, we provide more details about these two efficient mechanisms.

3.1. Recursive rule

The recursive rule that is introduced in the NGF [16] is based on the distance between the n-th parameter points of consecutive time steps. However, as the number of points changes with time in the AGNesF, the n-th point at time t might not always exist and the rule cannot be used. Here, we propose to check which point in $\{\boldsymbol{\theta}_{t-1}^j, w_{t-1}^j\}_{j=1}^{N_{\theta,t-1}}$ is the closest one to $\boldsymbol{\theta}_t^n$:

$$m^n = \underset{j \in 1,\ldots,N_{\theta,t-1}}{\arg\min} \; (\|\boldsymbol{\theta}_t^n - \boldsymbol{\theta}_{t-1}^j\|_p), \tag{9}$$

for $n = 1, \ldots, N_{\theta,t}$. Next, the maximum component of the vector $\|\boldsymbol{\theta}_t^n - \boldsymbol{\theta}_{t-1}^{m^n}\|_p$ is evaluated and compared against the relative threshold $\lambda\|\boldsymbol{\theta}_{t-1}^{m^n}\|$, for $\lambda > 0$. A smaller λ leads to a more restrictive rule and the algorithm takes longer to become strictly recursive, leading to lower estimation errors but larger runtimes. Contrarily, a larger λ allows to approximate $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n) \approx p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_{t-1}^{m^n})$ more often in the algorithm. Thus, the algorithm runs recursively, reducing the runtime but also the accuracy. However, the effect of an specific value of λ will depend on the problem or model of interest.

3.2. Adaptive rule

In order to decide when to reduce $N_{\theta,t}$ without compromising the performance of the filter, we use the following statistic

$$\rho_t = \frac{1}{\sum_{n=1}^{N_{\theta,t}} (\bar{s}_t^n)^2}, \tag{10}$$

$$\text{with} \quad \bar{s}_t^n = \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n)}{\sum_{n=1}^{N_{\theta,t}} p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n)}. \tag{11}$$

The statistic takes

- its minimum value in $\rho_t = 1$, which occurs when only one likelihood evaluation $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n)$, for $n = 1, \ldots, N_{\theta,t}$, is different from zero; and
- its maximum value in $\rho_t = N_{\theta,t}$, when for all $n = 1, \ldots, N_{\theta,t}$, the evaluations $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_t^n)$ are equal.

The latter case ($\rho_t = N_{\theta,t}$) happens when the posterior pdf $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$ is *narrow*, since the QKF locates the points closely in the parameter space. In this case, most of the points are redundant when evaluated in $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$ and barely provide any information. Therefore, we introduce an adaptive rule based on the statistic ρ_t to reduce $N_{\theta,t}$ in line 16 of Alg. 1, for $\epsilon > 0$. Otherwise, the number of points remains the same, i.e., $\alpha_t = \alpha_{t-1}$ and $N_{\theta,t} = N_{\theta,t-1}$. Thus, smaller ϵ translates into a more conservative adaptation.

The statistic ρ_t share some properties with the effective sample size (ESS) [27, 28], a widely used measure in sample-based simulation methods to assess the sample quality, with existing variations also for quadrature-based methods [29]. Although both statistic measures range from 1 to $N_{\theta,t}$, their interpretation differs. While a high ESS is related to a high-quality sample set, higher values of ρ_t indicate redundancy in the point set.

3.3. Discussion

The introduction of the recursive and the adaptive rules simplifies the configuration of the adaptive method, since we

avoid choosing $N_{\theta,t}$ at each time step by setting three parameters (λ, ϵ and α_{min}) at the begining of the simulation. First, we need to set ϵ in the range $0 < \epsilon \ll 1$. Smaller values of ϵ turn the algorithm more conservative, i.e., it is less likely to reduce $N_{\theta,t}$, thus, keeping longer a large number of samples. This entails a better performance but at a higher computational cost. Second, we set $\alpha_{\text{min}} = 2$ to ensure $N_{\theta,t} > 1$. Note that AGNesF runs a bank of $N_{\theta,t}$ filters in the bottom layer of the scheme. We implement extended Kalman filters (EKFs) in this layer for simplicity, although any filtering technique (i.e., deterministic or Monte Carlo-based) can be used.

4. NUMERICAL EXAMPLE

4.1. Stochastic Lorenz 63 model

Consider a stochastic version of the Lorenz 63 model [30, 31], with state $\boldsymbol{x}_t = [x_{1,t}, x_{2,t}, x_{3,t}]^\top \in \mathbb{R}^3$ described by

$$\begin{aligned} x_{1,t} &= x_{1,t-1} - \Delta S(x_{1,t-1} - x_{2,t-1}) + \sqrt{\Delta}\sigma v_{1,t}, \\ x_{2,t} &= x_{2,t-1} + \Delta[(R - x_{3,t-1})x_{1,t-1} - x_{2,t-1}] + \sqrt{\Delta}\sigma v_{2,t}, \\ x_{3,t} &= x_{3,t-1} + \Delta(x_{1,t-1}x_{2,t-1} - Bx_{3,t-1}) + \sqrt{\Delta}\sigma v_{3,t}, \end{aligned} \tag{12}$$

where $\boldsymbol{\theta} = [S, R, B]^\top \in \mathbb{R}^3$ is the unknown static parameter vector, Δ is the integration time-step (given in continuous-time units), $\sigma > 0$ is a known scale parameter, and $\{v_{i,t}\}_{i=1}^3$ are independent Wiener processes.

We assume linear observations of the form

$$\boldsymbol{y}_t = k_o \begin{bmatrix} x_{1,t} \\ x_{3,t} \end{bmatrix} + \boldsymbol{r}_t, \tag{13}$$

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[1]. Moreover, a new observation is only available every M_o steps of the state Eq. (12).

4.2. Numerical results

We generate signals $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, with $t = \{0, 1, \ldots\}$, using Eqs. (12) and (13). The true parameters are $S = 10$, $R = 28$, and $B = \frac{8}{3}$. We assume $k_o = 5$, $\sigma^2 = 0.1$, and $\sigma_y^2 = 1$ are known. We use an integration step of $\Delta = 2 \times 10^{-4}$ continuous-time units and we have $M_o = 5$ discrete-time steps between consecutive observations. The length of each simulation run is $T = 2 \times 10^5$ discrete-time steps of the state Eq. (12). For the algorithms, we set $\lambda = 10^{-3}$. We assume a Gaussian prior distribution $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_\theta, \boldsymbol{I}_3)$, where $\boldsymbol{\mu}_\theta$ is drawn at random from a uniform distribution $\mathcal{U}(\boldsymbol{\theta}^-, \boldsymbol{\theta}^+)$, for $\boldsymbol{\theta}^- = [5, 10, 0]^\top$ and $\boldsymbol{\theta}^+ = [15, 40, 10]^\top$. We also assume $p(\boldsymbol{x}_0) = \mathcal{N}(\widehat{\boldsymbol{x}}_0, \boldsymbol{I}_3)$, where $\widehat{\boldsymbol{x}}_0 = [-6, -5.5, -24.5]^\top$.

In Fig. 2, we compare the performance of the AGNesF with a NGF that uses a QKF in the parameter layer and a bank of EKFs in the state layer (i.e., QKF-EKF as in [16]). The QKF-EKF has a fixed number of points, $\alpha = \{2, 3, 4\}$, while AGNesF initialize with $\alpha_1 = 4$ which is reduced during the simulation run. We evaluate the proposed algorithm for $\epsilon = 10^{-i}$, with $i = \{1, \ldots, 5\}$. The performance is assessed by the averaged normalized mean square error (NMSE) of $\boldsymbol{\theta}$ over time, $\text{NMSE}_{\boldsymbol{\theta}}$, and the averaged running time. Each point of the graph represents the average of 80 independent simulation runs.

The error of the QKF-EKF is considerably lower for higher values of α (i.e., more points in the parameter space). However, there is a noticeable increase in the running time, going from 3.5 minutes for $\alpha = 2$ to 24 minutes for $\alpha = 4$. Similarly, the AGNesF obtains better performance in terms of error when the adaptive rule is more restrictive (i.e., for small values of ϵ). Then, the conditions to reduce $N_{\theta,t}$ are more difficult to meet and a large $N_{\theta,t}$ is kept for longer. As expected, the lowest computational cost is obtained for higher values of ϵ. In this case, α_t (and consequently $N_{\theta,t}$) is reduced quickly and the accuracy decreases. In general, the AGNesF significantly reduces the computational complexity in comparison to the NGF (QKF-EKF) for a given performance.

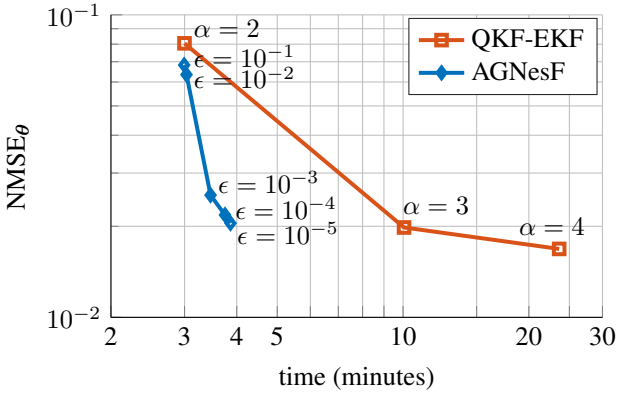

Fig. 2: Averaged $\text{NMSE}_{\boldsymbol{\theta}}$ against averaged running time in minutes for a QKF-EKF, and for the AGNesF with $\alpha_1 = 4$.

5. CONCLUSION

We have proposed the AGNesF algorithm, the first nested method that adapts the number of samples to compute the joint posterior probability distribution of the static parameters and the dynamical variables. The method, that is based on the two-layer NGF framework, implements two novel mechanisms to reduce the computational complexity. First, a new recursive rule enables the method to run recursively, avoiding to reprocess the whole sequence of data every time there is a new observation. Second, we introduce an adaptive rule to reduce automatically the number of samples in the parameter space when t grows. For that purpose, we describe a specific implementation that uses a QKF in the parameter layer. We have presented numerical results for a stochastic Lorenz 63 model, showing that the adaptive rule attains a significant reduction in the computational cost without accuracy loss.

[1] $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C})$ denotes the Gaussian pdf with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{C}$, and $\boldsymbol{I}_d$ denotes the $d \times d$ identity matrix.

6. REFERENCES

[1] A. M. Clayton, A. Lorenc, and D. M. Barker, "Operational implementation of a hybrid ensemble/4D-Var global data assimilation system at the Met Office," *Quarterly Journal of the Royal Meteorological Society*, vol. 139, no. 675, pp. 1445–1461, 2013.

[2] R. van de Schoot, S. Depaoli, R. King, B. Kramer, K. Märtens, M. G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, et al., "Bayesian statistics and modelling," *Nature Reviews Methods Primers*, vol. 1, no. 1, pp. 1–26, 2021.

[3] J. M. Read, J. R. Bridgen, D. A. Cummings, A. Ho, and C. P. Jewell, "Novel coronavirus 2019-nCoV: early estimation of epidemiological parameters and epidemic predictions," *medrxiv*, 2020.

[4] A. Virbickaitė, H. F. Lopes, M. C. Ausín, and P. Galeano, "Particle learning for Bayesian semi-parametric stochastic volatility model," *Econometric Reviews*, 2019.

[5] X. Wang, T. Li, S. Sun, and J. M. Corchado, "A survey of recent advances in particle filters and remaining challenges for multitarget tracking," *Sensors*, vol. 17, no. 12, pp. 2707, 2017.

[6] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo Sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[7] S. J. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the non linear transformation of means and covariances in filters and estimators," *IEEE Transactions Automatic Control*, vol. 3, pp. 477–482, March 2000.

[8] S. Särkkä, *Bayesian filtering and smoothing*, Number 3. Cambridge university press, 2013.

[9] N. Chopin, P. E. Jacob, and O. Papaspiliopoulos, "SMC^2: an efficient algorithm for sequential analysis of state space models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 75, no. 3, pp. 397–426, 2013.

[10] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society B*, vol. 72, pp. 269–342, 2010.

[11] D. Crisan and J. Míguez, "Nested particle filters for online parameter estimation in discrete-time state-space Markov models," *Bernoulli*, vol. 24, no. 4A, pp. 3039–3086, 2018.

[12] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE signal processing magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[13] A. Doucet, A. M. Johansen, et al., "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, pp. 3, 2009.

[14] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric, "Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 145–152, 2019.

[15] S. Pérez-Vieites, I. P. Mariño, and J. Míguez, "Probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems," *Physical Review E*, vol. 98(6), 063305, 2017.

[16] S. Pérez-Vieites and J. Míguez, "Nested Gaussian filters for recursive Bayesian inference and nonlinear tracking in state space models," *Signal Processing*, vol. 189, pp. 108295, 2021.

[17] A. Soto, "Self adaptive particle filter," in *IJCAI*. Citeseer, 2005, pp. 1398–1406.

[18] O. Straka and M. Šimandl, "Particle filter adaptation based on efficient sample size," *IFAC Proceedings Volumes*, vol. 39, no. 1, pp. 991–996, 2006.

[19] J. Cornebise, E. Moulines, and J. Olsson, "Adaptive methods for sequential importance sampling with application to state space models," *Statistics and Computing*, vol. 18, no. 4, pp. 461–480, 2008.

[20] A. Lee and N. Whiteley, "Variance estimation in the particle filter," *Biometrika*, vol. 105, no. 3, pp. 609–625, 2018.

[21] A. Bhadra and E. L. Ionides, "Adaptive particle allocation in iterated sequential Monte Carlo via approximating metamodels," *Statistics and Computing*, vol. 26, no. 1, pp. 393–407, 2016.

[22] L. Martino, J. Read, V. Elvira, and F. Louzada, "Cooperative parallel particle filters for online model selection and applications to urban mobility," *Digital Signal Processing*, vol. 60, pp. 172–185, 2017.

[23] V. Elvira, J. Míguez, and P. M. Djurić, "Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1781–1794, April 2017.

[24] V. Elvira, J. Míguez, and P. M. Djurić, "On the performance of particle filters with adaptive number of particles," *Statistics and Computing*, vol. 31, no. 6, pp. 1–18, 2021.

[25] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[26] I. Arasaratnam, S. Haykin, and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.

[27] L. Martino, V. Elvira, and F. Louzada, "Effective sample size for importance sampling based on discrepancy measures," *Signal Processing*, vol. 131, pp. 386–401, 2017.

[28] V. Elvira, L. Martino, and C. P. Robert, "Rethinking the effective sample size," *International Statistical Review*, vol. 90, no. 3, pp. 525–550, 2022.

[29] V. Elvira, L. Martino, and P. Closas, "Importance gaussian quadrature," *IEEE Transactions on Signal Processing*, vol. 69, pp. 474–488, 2020.

[30] I. P. Mariño and J. Míguez, "On a recursive method for the estimation of unknown parameters of partially observed chaotic systems," *Physica D*, vol. 220, pp. 175–182, September 2006.

[31] J. Míguez, "Analysis of parallelizable resampling algorithms for particle filtering," *Signal Processing*, vol. 87, no. 12, pp. 3155–3174, 2007.