A variational Bayesian multiple particle filtering scheme for large-dimensional systems

Boujemaa Ait-El-Fquih , *Member, IEEE,* and Ibrahim Hoteit

Abstract

This paper considers the Bayesian filtering problem in high-dimensional nonlinear state-space systems. In such systems, classical particle filters (PFs) are impractical due to the prohibitive number of required particles to obtain reasonable performances. One approach that has been introduced to overcome this problem is the concept of multiple PFs (MPFs), where the state-space is split into low-dimensional subspaces and then a separate PF is applied to each subspace. Remarkable performances of MPF-like filters motivated our investigation here into a new strategy that combines the variational Bayesian approach to split the state-space with random sampling techniques, to derive a new computationally efficient MPF. The propagation of each particle in the prediction step of the resulting filter requires generating only a single particle in contrast with standard MPFs, for which a set of (children) particles is required. We present simulation results to evaluate the behavior of the proposed filter and compare its performances against standard PF and a MPF.

Index Terms

Bayesian filtering; Particle filtering; Sequential Monte Carlo; High dimension; Variational Bayes; Hidden Markov Chain.

I. Introduction

Many problems in statistical signal processing are interested in estimating an unknown dynamical process, $\mathbf{x} = \{\mathbf{x}_n\}_{n\in\mathbb{N}} \in \mathbb{R}^{n_\mathbf{x}}$, from an observed process, $\mathbf{y} = \{\mathbf{y}_n\}_{n\in\mathbb{N}} \in \mathbb{R}^{n_\mathbf{y}}$. Let note $\mathbf{x}_{0:n} =$

B. Ait-El-Fquih and I. Hoteit are with the Division of Applied Mathematics & Computational Science, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia (e-mail:boujemaa.aitelfquih@kaust.edu.sa, ibrahim.hoteit@kaust.edu.sa).

$\{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_n\}$ and $\mathbf{y}_{0:n} = \{\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_n\}$. Let also $p(\mathbf{x}_n)$ and $p(\mathbf{x}_n|\mathbf{y}_{0:m})$ denote the probability density function (pdf) (with respect to - w.r.t. - Lebesgue measure) of $\mathbf{x}_n$ and the pdf of $\mathbf{x}_n$ conditional on $\mathbf{y}_{0:m}$, respectively. Such problems are usually addressed in the framework of a hidden Markov chain (HMC) model for which [1] [2] [3]:

$$p(\mathbf{x}_{0:n}) = p(\mathbf{x}_0) \prod_{k=1}^{n} p(\mathbf{x}_k|\mathbf{x}_{k-1}), \tag{1}$$

$$p(\mathbf{y}_{0:n}|\mathbf{x}_{0:n}) = \prod_{k=0}^{n} p(\mathbf{y}_k|\mathbf{x}_k). \tag{2}$$

A classical HMC is given by the well-known state-space system [1] [2] [4] [5],

$$\begin{cases} \mathbf{x}_{n+1} = \mathcal{F}_n(\mathbf{x}_n, \mathbf{u}_n), \\ \mathbf{y}_n = \mathcal{H}_n(\mathbf{x}_n, \mathbf{v}_n), \end{cases} \tag{3}$$

for which the state transition function, $\mathcal{F}_n$, and the measurement function, $\mathcal{H}_n$, are not necessarily linear; the input noise, $\mathbf{u} = \{\mathbf{u}_n\}_{n\in\mathbb{N}}$, and the observation noise, $\mathbf{v} = \{\mathbf{v}_n\}_{n\in\mathbb{N}}$, are assumed to be independent, jointly independent, and independent of the initial state, $\mathbf{x}_0$. The filtering problem consists in computing the so-called *filtering* pdf, $p(\mathbf{x}_n|\mathbf{y}_{0:n})$, at any instant in time $n = 0, 1, \cdots$. Any relevant information of $\mathbf{x}_n$ given $\mathbf{y}_{0:n}$ can then be computed, at least in theory, as for instance the *a posteriori* mean (AM),

$$\hat{\mathbf{x}}_{n|n} = \mathbb{E}_{p(\mathbf{x}_n|\mathbf{y}_{0:n})}[\mathbf{x}_n] = \int \mathbf{x}_n p(\mathbf{x}_n|\mathbf{y}_{0:n}) d\mathbf{x}_n, \tag{4}$$

which minimizes the *a posteriori* mean square error (MSE). Hereafter, as in (4), we denote by $\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})]$ the expected value of a function $f(\mathbf{x})$ w.r.t. the distribution $p(\mathbf{x})$. Factorizations (1)-(2) are key tools enabling *recursive* computation of the filtering pdf following two steps [4]:

- *Markovian* (or *prediction*) step, which uses the transition pdf of the HMC, $p(\mathbf{x}_n|\mathbf{x}_{n-1})$, to compute the prediction pdf:
$$p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1}) p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1}. \tag{5}$$
- *Bayesian* (or *filtering*) step, in which the likelihood, $p(\mathbf{y}_n|\mathbf{x}_n)$, is used to update the prediction pdf via the Bayes' rule:
$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n-1})}{\int p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) d\mathbf{x}_n}. \tag{6}$$

In practice, unless the HMC model is linear and Gaussian, say, $\mathcal{F}_n$ and $\mathcal{H}_n$ in (3) are linear, and $\mathbf{x}_0$, $\mathbf{u}_n$ and $\mathbf{v}_n$ are Gaussian, an analytical computation of the integrals in (4)-(6) is rarely possible. Many numerical methods approximating these integrals have therefore been developed. Among these, the most popular are the particle filters (PFs), which aim to propagate a Monte Carlo (MC) approximation of

$p(d\mathbf{x}_n|\mathbf{y}_{0:n})$, and subsequently a MC approximation of the AM estimate [2] [5] [6] [7] [8] [9] [10] [11] [12]. However, it is well-known that PFs, in their generic form, suffer from weights degeneracy in the sense that after a few filtering cycles, the weights of all particles, except one, become equal to zero, and this may significantly degrade the filter behavior and even cause divergence [5]. A number of judicious choices of importance densities, along with various resampling procedures have been proposed in an attempt to tackle this issue [5] [9] [13]. Although many of these have provided satisfactory results in a number of low-dimensional state-space applications [10] [5], they nevertheless remain inefficient for systems with very large state dimension, $n_{\mathbf{x}}$ [14] [15] [16] [17]. This is mainly because the number of particles required to sufficiently sample the state-space needs to be very large; in some special situations it has been even shown that the required number of particles scales exponentially with $n_{\mathbf{x}}$ (see e.g. [15]).

To overcome this drawback, a class of multiple PFs (MPFs) has recently been proposed [18] [19] [20] [14] [21] [22], among other approaches (see e.g. [23] [24] and references therein). Loosely speaking, the concept of MPFs involves splitting the state-space into K subspaces of smaller dimensions, assuming (*a posteriori*) the associated state partitions to be independent, and then applying one PF to each subspace. Motivated by the remarkable performances of MPF-like filters, here, we apply the variational Bayesian (VB) approach to introduce a new MPF, called the variational Bayesian MPF (VBMPF), which we derive by combining VB-like density approximations with MC-like approximations. Instead of directly assuming that the posterior pdf of the system state is separable, this pdf is first (theoretically) approximated by a product of independent marginal pdfs based on the Kullback-Leibler (KL) divergence minimization criteria (VB approximation stage) [25] [26] [27]. Each VB-like marginal distribution is then sampled as in the standard PF [11]. Each PF component of the VBMPF uses particles from the other components, similar to the MPF approach but following a different mechanism. More specifically, our goal is to derive a new MPF-like scheme that is theoretically sound and numerically efficient as compared to the standard MPF. The new filter will be derived based on the VB-like optimization criteria to split (*a posteriori*) the system state, in contrast with the standard MPF that assumes posterior independence of state partitions without using any mathematical criteria. Furthermore, for each state partition, the sampling step of our scheme generates one single particle for each trajectory which makes it computationally more attractive than the MPF, which requires generating a set of particles. Numerical experiments will be presented to demonstrate the efficiency of the proposed filter compared to the standard MPF and PF.

While the VB approach has already been adopted in several Bayesian inference problems (see e.g. [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] and references therein), it has not yet been applied for splitting the (large-dimensional) state in a PF framework. This paper is organized as follows. Section II

describes the generic algorithm that propagates VB approximations of the filtering pdf. The VBMPF is then derived in Section III by applying two classical random sampling techniques to the generic algorithm from Section II, and a guideline for using the VBMPF is provided among other findings. Comparison with standard MPFs, along with extensions to more general cases are discussed in Section IV. Numerical simulations are performed in Section V comparing the performances of the proposed VBMPF w.r.t. standard PF and a MPF. Concluding remarks are given in Section VI.

II. The generic VB filtering algorithm

Let the state vector, $\mathbf{x}_n$, be split into K sub-vectors (or partitions), $\mathbf{x}_n^1, \mathbf{x}_n^2, \cdots, \mathbf{x}_n^K$, of the same dimension, $n_{\mathbf{x}^k} = \frac{n_\mathbf{x}}{K} \in \mathbb{N}$. The extension of the proposed scheme to the more general case, for which the state is split into partitions with different dimensions, is trivial. A separable VB approximation, $\prod_{k=1}^K q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$, of the filtering pdf, $p(\mathbf{x}_n|\mathbf{y}_{0:n})$, is obtained under the criteria

$$\prod_{k=1}^K q(\mathbf{x}_n^k|\mathbf{y}_{0:n}) = \underset{\prod_k \tilde{p}(\mathbf{x}_n^k|\mathbf{y}_{0:n})}{\operatorname{argmin}} \mathrm{KL}\left(\prod_{k=1}^K \tilde{p}(\mathbf{x}_n^k|\mathbf{y}_{0:n}) || p(\mathbf{x}_n|\mathbf{y}_{0:n})\right), \tag{7}$$

where each marginal $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$ "part" of the solution of (7) is given by [27] [30]:

$$q(\mathbf{x}_n^k|\mathbf{y}_{0:n}) \propto \exp\left(\mathbb{E}_{q(\mathbf{x}_n^{k^-}|\mathbf{y}_{0:n})}\left[\ln(p(\mathbf{x}_n, \mathbf{y}_{0:n}))\right]\right), \tag{8}$$

where $\mathbf{x}_n^{k^-}$ denotes the complement of $\mathbf{x}_n^k$ in the vector $\mathbf{x}_n$. Eq. (8) points to the fact that although the approximated marginal pdf of $\mathbf{x}_n^k$, $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$, is independent of those of the other components $\mathbf{x}_n^{k^-}$, $q(\mathbf{x}_n^{k^-}|\mathbf{y}_{0:n})$, it nevertheless remains dependent on the expected value of $\ln[p(\mathbf{x}_n, \mathbf{y}_{0:n})]$ w.r.t. $q(\mathbf{x}_n^{k-}|\mathbf{y}_{0:n})$. In other words, the "enforced" independence on the marginal pdfs is partially compensated by a functional dependence involving expectations w.r.t. these pdfs. For instance, one can easily verify that in the Gaussian case $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$ remains dependent of the first two moments of $q(\mathbf{x}_n^{k-}|\mathbf{y}_{0:n})$.

Once $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$ is computed, an estimate of $\mathbf{x}_n^k$ can then be obtained by the AM, $\mathbb{E}_{q(\mathbf{x}_n^k|\mathbf{y}_{0:n})}[\mathbf{x}_n^k]$. Below we propose a *recursive* algorithm to compute the solutions (8) based on (5)-(6) in two steps.

A. Prediction step

Starting at time $n-1$ from a VB-like approximation,

$$p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) \approx \prod_{k=1}^K q(\mathbf{x}_{n-1}^k|\mathbf{y}_{0:n-1}), \tag{9}$$

we aim to compute a separable approximation of the prediction pdf, $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$, using (5). Let's consider for now the following factorization of the HMC transition pdf:

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}) = \prod_{k=1}^K p(\mathbf{x}_n^k|\mathbf{x}_{n-1}). \tag{10}$$

By inserting (9) and (10) into (5), one obtains

$$p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) \approx \int \prod_{k=1}^{K} p(\mathbf{x}_n^k|\mathbf{x}_{n-1}) q(\mathbf{x}_{n-1}^k|\mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1}. \tag{11}$$

However, the approximation (11) is not separable, that is not under a form of products of K marginal pdfs of $\mathbf{x}_n^k$ given $\mathbf{y}_{0:n-1}$ unless the marginal processes, $\mathbf{x}^k = \{\mathbf{x}_n^k\}_n$, are independent Markov chains ($p(\mathbf{x}_n^k|\mathbf{x}_{0:n-1}) = p(\mathbf{x}_n^k|\mathbf{x}_{0:n-1}^k) = p(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k)$; see [30] [38] for the linear Gaussian case). Moreover, using this approximation in the filtering step (6) does not guarantee a separable approximation of $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ even when the approximation of $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$ (9) is separable. One way to go around this is to approximate the marginal transition pdf, $p(\mathbf{x}_n^k|\mathbf{x}_{n-1})$ (involved in (11)), which is equal to $p(\mathbf{x}_n^k|\mathbf{x}_{n-1}, \mathbf{y}_{0:n-1})$, by a pdf of the form, $q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1})$. In other words, the idea is to drop the dependence between $\mathbf{x}_n^k$ and $\mathbf{x}_{n-1}^{k-}$ conditionally on $(\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1})$, for $k = 1, 2, \cdots, K$. This can be done using the VB approach, by approximating $p(\mathbf{x}_n^k, \mathbf{x}_{n-1}^{k-}|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1})$ as

$$p(\mathbf{x}_n^k, \mathbf{x}_{n-1}^{k-}|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1}) \approx q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1}) \times \underbrace{q(\mathbf{x}_{n-1}^{k-}|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1})}_{q(\mathbf{x}_{n-1}^{k-}|\mathbf{y}_{0:n-1}) \text{ (see (9))}}. \tag{12}$$

Similarly to (8), one obtains,

$$q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1}) \propto \exp\Big(\mathbb{E}_{q(\mathbf{x}_{n-1}^{k-}|\mathbf{y}_{0:n-1})}\Big[\ln(p(\mathbf{x}_n^k|\mathbf{x}_{n-1}))\Big]\Big). \tag{13}$$

Replacing in (11) $p(\mathbf{x}_n^k|\mathbf{x}_{n-1})$ by its approximation (13), one eventually obtains a separable approximation of the joint prediction pdf:

$$p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) \approx \prod_{k=1}^{K} q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1}), \tag{14}$$

with

$$q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1}) = \int q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1}) q(\mathbf{x}_{n-1}^k|\mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1}^k. \tag{15}$$

B. Filtering step

The filtering step consists in establishing, for each sub-vector $\mathbf{x}_n^k$, a relationship between the VB-like marginal filtering pdf, $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$, given in (8), and the marginal prediction pdf, $q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$. Inserting in (8) the factorization,

$$p(\mathbf{x}_n, \mathbf{y}_{0:n}) \propto p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n-1}),$$

which, in turn, arises from (1)-(2), then using (14) one obtains[1]

$$q(\mathbf{x}_n^k|\mathbf{y}_{0:n}) \propto q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1}) \times \underbrace{\exp\Big(\mathbb{E}_{q(\mathbf{x}_n^{k-}|\mathbf{y}_{0:n})}[\ln(p(\mathbf{y}_n|\mathbf{x}_n)]\Big)}_{\mathcal{L}(\mathbf{x}_n^k)}. \tag{16}$$

The computation of the VB marginal filtering pdf of the k^{th} sub-vector, $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$, using (16) requires the knowledge of those of the other sub-vectors, $q(\mathbf{x}_n^{k-}|\mathbf{y}_{0:n})$. However, these latter are unknown, which makes it impossible to derive an analytical solution of (16). Similarly to [27] [30], one may circumvent this issue by proceeding with iterations of (16) for all k, starting from the initialization $p(\mathbf{x}_n^{1-}|\mathbf{y}_{0:n}) = p(\mathbf{x}_n^{1-}|\mathbf{y}_{0:n-1})$, to approximate the VB marginal filtering pdfs of interest.

Remark 1: An important remark can be made from Eqs. (15)-(16). Eq. (15) appears to be a fully Markovian step computing the VB marginal prediction pdf, $q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$, from the VB marginal filtering pdf, $q(\mathbf{x}_{n-1}^k|\mathbf{y}_{0:n-1})$, using the Markovian transition pdf, $q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1})$, which approximates $p(\mathbf{x}_n^k|\mathbf{x}_{n-1})$. On the other hand, Eq. (16) looks like a fully Bayesian step linking the prior, $q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$, with the posterior, $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$, using Bayes' rule, where the multiplicative term, $\mathcal{L}(\mathbf{x}_n^k)$, is seen as a likelihood approximating $p(\mathbf{y}_n|\mathbf{x}_n^k, \mathbf{y}_{0:n-1})$. This emphasises the relevance of the proposed approach to approximate the Markovian and Bayesian steps (5)-(6) involving the full joint state vector, $\mathbf{x}_n$, by the Markovian and Bayesian steps (15)-(16) associated with each marginal $\mathbf{x}_n^k$.

III. The VBMPF algorithm

The practical derivation of the proposed approach depends on the nature of the system (3). Four classical cases can be distinguished depending on whether the noises, $\mathbf{u}_n$ and $\mathbf{v}_n$, are additive or multiplicative. For the sake of simplicity we focus here on the case of additive noises, that is

$$\begin{cases} \mathbf{x}_{n+1} = \mathbf{f}_n(\mathbf{x}_n) + \mathbf{u}_n, \\ \mathbf{y}_n = \mathbf{h}_n(\mathbf{x}_n) + \mathbf{v}_n, \end{cases} \tag{17}$$

where $\mathbf{f}_n$ and $\mathbf{h}_n$ are nonlinear functions. Lets assume $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$, $\mathbf{u}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_n)$, and $\mathbf{v}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_n)$, with $\mathbf{P}_0$, $\mathbf{Q}_n$ and $\mathbf{R}_n$ are positive definite matrices; $\mathbf{0}$ stands for a zero vector with appropriate dimension. The assumption (10) amounts to assuming $\mathbf{Q}_n$ block diagonal. The nonlinearity and the assumed multidimensionality of $\mathbf{f}_n$ and $\mathbf{h}_n$ make it very difficult, or even impossible, to analytically compute the integrals in (13), (15) and (16); closed forms of $q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$ and $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$ are therefore inaccessible. We thus resort to a PF-like algorithm, the VBMPF, propagating MC representations of

[1]Without abuse of notation, the right hand side (r.h.s.) of (16) is rather proportional to an *approximation* of (8).

these pdfs. The derivation of the proposed algorithm mainly relies on two random sampling techniques that we recall in Properties 1 and 2 below. The first technique, which is used in the prediction step, requires using a transition pdf, $p(\mathbf{x}_2|\mathbf{x}_1)$, to (exactly) draw samples from $p(\mathbf{x}_2) = \int p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1)d\mathbf{x}_1$ starting from those of $p(\mathbf{x}_1)$. The second technique, which is used in the filtering step, is the Rubin's Sampling Importance Resampling (SIR) mechanism, which consists in using the likelihood, $p(\mathbf{y}|\mathbf{x})$, to (approximately) draw samples from the posterior pdf, $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, starting from those of the prior pdf, $p(\mathbf{x})$.

Property 1 (Hierarchical sampling [39]): Assuming that one can sample from $p(\mathbf{x}_1)$ and $p(\mathbf{x}_2|\mathbf{x}_1)$. Then, a sample, $\mathbf{x}_2^*$, from $p(\mathbf{x}_2)$ can be drawn as follows:

1) $\mathbf{x}_1^* \sim p(\mathbf{x}_1)$;
2) $\mathbf{x}_2^* \sim p(\mathbf{x}_2|\mathbf{x}_1^*)$.

Property 2 (Rubin's SIR [40]): Assuming that one has a set of samples, $\{\mathbf{x}^{(s)}\}_{s=1}^S$, independently and identically drawn (i.i.d.) from a prior, $p(\mathbf{x})$. Then, one can asymptotically draw an i.i.d. set, $\{\mathbf{x}^{*(s)}\}_{s=1}^S$, from the posterior, $p(\mathbf{x}|\mathbf{y})$, in two steps:

1) *Weighting.* A weight, $\omega^{(s)} \propto p(\mathbf{y}|\mathbf{x}^{(s)})$ with $\sum_{s=1}^S \omega^{(s)} = 1$, is first assigned to each sample $\mathbf{x}^{(s)}$; $s = 1, 2, \cdots, S$.
2) *Resampling.* Each sample, $\mathbf{x}^{*(s)}$, is then computed by sampling from the probability mass function (pmf), $\sum_{\ell=1}^S \omega^{(\ell)}\delta(\mathbf{x} - \mathbf{x}^{(\ell)})$, where $\delta(.)$ is the symbol of Kronecker.

A. Sampling step (Prediction)

This section proposes MC approximations of the generic formulas of Section II-A. Assume that for $n \geq 1$ one has an i.i.d. random set of samples (or particles), $\{\mathbf{x}_{n-1}^{k,(s)}\}_{s=1}^{S\gg 1}$, from $q(\mathbf{x}_{n-1}^k|\mathbf{y}_{0:n-1})$, and wants to sample an i.i.d. random set, $\{\tilde{\mathbf{x}}_n^{k,(s)}\}_{s=1}^S$, from $q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$, $k = 1, \cdots, K$. This can be achieved by first deriving (13) to provide an explicit form (but not exact) of $q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1})$, then by using the hierarchical sampling technique (Prop. 1) in (15).

1) Approximation of $q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1})$: From (17) we have,

$$p(\mathbf{x}_n^k|\mathbf{x}_{n-1}) = \mathcal{N}_{\mathbf{x}_n^k}(\mathbf{f}_{n-1}^k(\mathbf{x}_{n-1}), \mathbf{Q}_{n-1}^k), \tag{18}$$

where $\mathcal{N}_{\mathbf{u}}(\mathbf{m}, \mathbf{C})$ denotes a Gaussian pdf with argument $\mathbf{u}$ and parameters $(\mathbf{m}, \mathbf{C})$, $\mathbf{f}_{n-1}^k(\mathbf{x}_{n-1})$ the k^{th} $n_{\mathbf{x}^k} \times 1$ sub-vector of $\mathbf{f}_{n-1}(\mathbf{x}_{n-1})$, and $\mathbf{Q}_{n-1}^k$ the k^{th} $n_{\mathbf{x}_n^k} \times n_{\mathbf{x}_n^k}$ diagonal block of $\mathbf{Q}_{n-1}$. Inserting (18) in (13) shows that $q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1})$ is Gaussian of the mean,

$$\boldsymbol{\mu}_n^k(\mathbf{x}_{n-1}^k) = \mathbb{E}_{q(\mathbf{x}_{n-1}^{k-}|\mathbf{y}_{0:n-1})}\left[\mathbf{f}_{n-1}^k(\mathbf{x}_{n-1})\right], \tag{19}$$

and covariance, $\mathbf{Q}_{n-1}^k$. However, since the multidimensional function, $\mathbf{f}_{n-1}^k(\mathbf{x}_{n-1})$, is possibly nonlinear and only a MC approximation of the measure, $q(d\mathbf{x}_{n-1}^{k-}|\mathbf{y}_{0:n-1})$, is available, the analytical computation of the mean (integral) (19) is not possible. We thus use the MC approximation of $q(d\mathbf{x}_{n-1}^{k-}|\mathbf{y}_{0:n-1})$, $\sum_{s=1}^S \frac{1}{S}\delta_{\mathbf{x}_{n-1}^{k-,(s)}}(d\mathbf{x}_{n-1}^{k-})$, where $\delta_{\mathbf{u}}(.)$ stands for the Dirac mass at point $\mathbf{u}$, to approximate (19). One obtains,

$$\hat{\boldsymbol{\mu}}_n^k(\mathbf{x}_{n-1}^k) = \frac{1}{J}\sum_{\ell=1}^J \mathbf{f}_{n-1}^k(\mathcal{X}_{k,n-1}^{(s_\ell)}), \tag{20}$$

where J could be equal to S, $\mathcal{X}_{k,n-1}^{(s_\ell)} \stackrel{\text{def}}{=} \left[(\mathbf{x}_{n-1}^{1,(s_\ell)})^T,\cdots,(\mathbf{x}_{n-1}^{k-1,(s_\ell)})^T,(\mathbf{x}_{n-1}^{k})^T,(\mathbf{x}_{n-1}^{k+1,(s_\ell)})^T,\cdots,(\mathbf{x}_{n-1}^{K,(s_\ell)})^T\right]^T$, and each sub-vector $\mathbf{x}_{n-1}^{j,(s_\ell)}$ with $j \neq k$ (all except the k^{th}) is drawn from the pmf, $\sum_{s=1}^S \frac{1}{S}\delta(\mathbf{x}_{n-1}^{j}-\mathbf{x}_{n-1}^{j,(s)})$. In other words, for each subspace k, the mean (19) is approximated by resampling J particles $\mathbf{x}_{n-1}^{j,(s_\ell)}$ from the other subspaces $j \neq k$; evaluating the functions $\mathbf{f}_{n-1}^k(\mathcal{X}_{k,n-1}^{(s_\ell)})$ using the resampled particles; then taking their average as in (20).

2) Computation of $\{\tilde{\mathbf{x}}_n^{k,(s)}\}_{s=1}^S \sim q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$*:* Based on the previous result and (15), the hierarchical sampling technique (Prop. 1) can then be used to independently sample $\{\tilde{\mathbf{x}}_n^{k,(s)}\}_{s=1}^S$ from $q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$ by drawing each particle, $\tilde{\mathbf{x}}_n^{k,(s)}$, from the Gaussian, $\mathcal{N}(\hat{\boldsymbol{\mu}}_n^k(\mathbf{x}_{n-1}^{k,(s)}),\, \mathbf{Q}_{n-1}^k)$. The prediction step (or sampling step) of the VBMPF that computes for $n \geq 1$ a random set $\{\tilde{\mathbf{x}}_n^{(s)}\}_{s=1}^S$ drawn from the VB-like approximate prediction pdf, $\prod_{k=1}^K q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$, starting from a random set $\{\mathbf{x}_{n-1}^{(s)}\}_{s=1}^S$ drawn from the previous VB-like approximate filtering pdf, $\prod_{k=1}^K q(\mathbf{x}_{n-1}^k|\mathbf{y}_{0:n-1})$, is summarized in Algorithm 1 below. The initial (predicted) particles, $\{\tilde{\mathbf{x}}_0^{(s)}\}_{s=1}^S$, are independently generated from $p(\mathbf{x}_0) = \mathcal{N}_{\mathbf{x}_0}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$ by assuming $\mathbf{P}_0$ to be block diagonal with $(n_{\mathbf{x}_n^k} \times n_{\mathbf{x}^k})$ blocks.

Algorithm 1 : $\{\tilde{\mathbf{x}}_n^{k,(s)}\}_{s=1}^S = \text{SAMPLING}\left(\{\mathbf{x}_{n-1}^{(s)}\}_{s=1}^S\right),\ k = 1, 2, \cdots, K$

- For $j = 1, 2, \cdots, K$, draw independently $\{\mathbf{x}_{n-1}^{j,(s_\ell)}\}_{\ell=1}^J$ from $\sum_{s=1}^S \frac{1}{S}\delta(\mathbf{x}_{n-1}^j - \mathbf{x}_{n-1}^{j,(s)})$
- For $k = 1, 2, \cdots, K$, for $s = 1, 2, \cdots, S$,
  - For $\ell = 1, 2, \cdots, J$, evaluate the functions, $\mathbf{f}_{n-1}^k(\mathcal{X}_{k,n-1}^{(s_\ell)})$, where $\mathcal{X}_{k,n-1}^{(s_\ell)}$ is given in (20) with $\mathbf{x}_{n-1}^k = \mathbf{x}_{n-1}^{k,(s)}$
  - Take their average as in (20) to obtain the approximate mean, $\hat{\mu}_n^k(\mathbf{x}_{n-1}^{k,(s)})$
  - Draw $\tilde{\mathbf{x}}_n^{k,(s)}$ from $\mathcal{N}\left(\hat{\boldsymbol{\mu}}_n^k(\mathbf{x}_{n-1}^{k,(s)}), \mathbf{Q}_{n-1}^k\right)$

Let $\mathcal{C}_{\mathbf{f}}$ and $\mathcal{C}_{\mathsf{R}}(J)$ respectively denote the cost of evaluating the function, $\mathbf{f}_n(\mathbf{x}_n)$, and the cost of drawing J samples from a (possibly weighted) discret set of $M \geq J$ particles. The number of floating

operations (flops) required by Algorithm 1 at each time step is approximately

$$\mathcal{C}^{\mathsf{S}}_{\mathsf{VBMPF}} \quad = \quad SJ\mathcal{C}_{\mathbf{f}} + \left[SJ + 3S + \frac{\mathcal{C}_{\mathsf{R}}(J)}{n_{\mathbf{x}^k}} + 1\right] n_{\mathbf{x}}. \tag{21}$$

For simplicity, the cost (21) has been computed under the assumption of $\mathbf{Q}_n$ diagonal. All other costs computed later in the manuscript will be based on this same assumption.

B. Weighting and resampling steps (Filtering)

This section considers a MC implementation of the Bayes' like formula (16). Starting for $n \geq 0$ from the i.i.d. prediction sets of particles, $\{\tilde{\mathbf{x}}_n^{k,(s)}\}_{s=1}^S$, $k = 1, 2, \cdots, K$, computed with Algorithm 1, one wishes to sample i.i.d. sets of particles, $\{\mathbf{x}_n^{k,(s)}\}_{s=1}^S$, from $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$, $k = 1, 2, \cdots, K$. Using the Rubin's SIR strategy (Prop. 2), this can be done by performing a weighting step assigning, for each k, a normalized weight, $\lambda_n^{k,(s)}$, followed by a resampling step from the discrete pmf, $\sum_{s=1}^S \lambda_n^{k,(s)} \delta(\mathbf{x}_n^k - \tilde{\mathbf{x}}_n^{k,(s)})$.

Regarding the weighting step, one can see from (16) that for each k,

$$\lambda_n^{k,(s)} = \frac{\mathcal{L}(\tilde{\mathbf{x}}_n^{k,(s)})}{\sum_{s'=1}^S \mathcal{L}(\tilde{\mathbf{x}}_n^{k,(s')})}. \tag{22}$$

Due to the Gaussian likelihood (see (17)),

$$p(\mathbf{y}_n|\mathbf{x}_n) = \mathcal{N}_{\mathbf{y}_n}(\mathbf{h}_n(\mathbf{x}_n), \mathbf{R}_n), \tag{23}$$

$\mathcal{L}(\mathbf{x}_n^k)$ is expressed as (considering any sub-state $\mathbf{x}_n^k$ for now),

$$\mathcal{L}(\mathbf{x}_n^k) \propto \exp\left(-\frac{1}{2}\mathbb{E}_{q(\mathbf{x}_n^{k-}|\mathbf{y}_{0:n})}\left[||\mathbf{y}_n - \mathbf{h}_n(\mathbf{x}_n)||^2_{\mathbf{R}_n^{-1}}\right]\right), \tag{24}$$

where $||.||^2_{\mathbf{W}}$ denotes the euclidean norm weighted by $\mathbf{W}$. The expectation in (24) involves the separable VB approximation of the marginal filtering pdf $p(\mathbf{x}_n^{k-}|\mathbf{y}_{0:n})$, $q(\mathbf{x}_n^{k-}|\mathbf{y}_{0:n}) = \prod_{\substack{j=1 \\ j\neq k}}^K q(\mathbf{x}_n^j|\mathbf{y}_{0:n})$, which is not known. Instead of considering an iterative strategy to overcome this problem, as suggested in Section II-B, we base our derivation on the available prediction samples, $\{\tilde{\mathbf{x}}_n^{k,(s)}\}_{s=1}^S$, $k = 1, \cdots, K$, and resort to a non iterative computationally efficient scheme to compute a MC approximation of the expectation in (24). Indeed, multiplying both sides of the Bayesian update (6) by $d\mathbf{x}_n$, and replacing $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$ by its separable approximation, $q(\mathbf{x}_n|\mathbf{y}_{0:n-1}) = \prod_{k=1}^K q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$, one obtains,

$$p(d\mathbf{x}_n|\mathbf{y}_{0:n}) \quad \approx \quad \frac{p(\mathbf{y}_n|\mathbf{x}_n) q(d\mathbf{x}_n|\mathbf{y}_{0:n-1})}{\int p(\mathbf{y}_n|\mathbf{x}_n) q(\mathbf{x}_n|\mathbf{y}_{0:n-1}) d\mathbf{x}_n}, \tag{25}$$

$$\approx \quad \sum_{s=1}^S w_n^{(s)} \delta_{\tilde{\mathbf{x}}_n^{(s)}}(d\mathbf{x}_n), \tag{26}$$

where

$$\begin{aligned} w_n^{(s)} &\propto p(\mathbf{y}_n|\tilde{\mathbf{x}}_n^{(s)}) \\ &\propto \exp\left(-\frac{1}{2}||\mathbf{y}_n - \mathbf{h}_n(\tilde{\mathbf{x}}_n^{(s)})||^2_{\mathbf{R}_n^{-1}}\right); \end{aligned} \tag{27}$$

$\sum_{s=1}^S w_n^{(s)} = 1$. A MC approximation of $p(d\mathbf{x}_n^{k-}|\mathbf{y}_{0:n})$ is then obtained by marginalization and inserted in r.h.s. of (24) to obtain (by setting $\mathbf{x}_n^k = \tilde{\mathbf{x}}_n^{k,(s)}$):

$$\lambda_n^{k,(s)} \propto \exp\left(-\frac{1}{2}\sum_{s'=1}^S w_n^{(s')}||\mathbf{y}_n - \mathbf{h}_n(\mathcal{Z}_{k,n}^{(s,s')})||^2_{\mathbf{R}_n^{-1}}\right), \tag{28}$$

with $\mathcal{Z}_{k,n}^{(s,s')} \stackrel{\text{def}}{=} \left[(\tilde{\mathbf{x}}_n^{1,(s')})^T, \cdots, (\tilde{\mathbf{x}}_n^{k-1,(s')})^T, (\tilde{\mathbf{x}}_n^{k,(s)})^T, (\tilde{\mathbf{x}}_n^{k+1,(s')})^T, \cdots, (\tilde{\mathbf{x}}_n^{K,(s')})^T\right]^T$. In other words, the weight of each particle, $\tilde{\mathbf{x}}_n^{k,(s)}$, is computed by evaluating the observation model function, $\mathbf{h}_n(\mathcal{Z}_{k,n}^{(s,s')})$, using the particles from the other subspaces, $\{\tilde{\mathbf{x}}_n^{j,(s')}\}_{s'=1}^S$, $j \neq k$; evaluating the weighted squared errors, $-\frac{1}{2}w_n^{(s')}||\mathbf{y}_n - \mathbf{h}_n(\mathcal{Z}_{k,n}^{(s,s')})||^2_{\mathbf{R}_n^{-1}}$, for $s' = 1, \cdots, S$; and taking the exponential of their sum as in (28). This should be followed by a normalization step as in the standard PF to set $\sum_{s=1}^S \lambda_n^{k,(s)} = 1$.

Based on the weighted set of particles, $\{\tilde{\mathbf{x}}_n^{k,(s)}, \lambda_n^{k,(s)}\}_{s=1}^S$, any finite integral of the form $\int g(\mathbf{x}_n^k) q(\mathbf{x}_n^k|\mathbf{y}_{0:n}) d\mathbf{x}_n^k$ can be approximated by $\sum_{s=1}^S \lambda_n^{k,(s)} g(\tilde{\mathbf{x}}_n^{k,(s)})$; the AM estimate of the k^{th} partition, $\hat{\mathbf{x}}_{n|n}^k$, corresponds to $g(\mathbf{x}_n^k) = \mathbf{x}_n^k$. The weighting step of the VBMPF that computes for $n \geq 0$ a normalized weight, $\lambda_n^{k,(s)}$, for each particle, $\tilde{\mathbf{x}}_n^{k,(s)}$, is summarized in Algorithm 2 below. Once the weighted support of particles, $\{\tilde{\mathbf{x}}_n^{k,(s)}, \lambda_n^{k,(s)}\}_{s=1}^S$, is formed for each k, the resampling step is then performed providing an i.i.d. set of particles, $\{\mathbf{x}_n^{k,(s)}, \frac{1}{S}\}_{s=1}^S$, by independently drawing each particle, $\mathbf{x}_n^{k,(s)}$, from the discrete pmf, $\sum_{s'=1}^S \lambda_n^{k,(s')} \delta(\mathbf{x}_n^k - \tilde{\mathbf{x}}_n^{k,(s')})$.

Algorithm 2 : $\{\lambda_n^{k,(s)}\}_{s=1}^S = \text{WEIGHTING}\left(\{\tilde{\mathbf{x}}_n^{(s)}, \frac{1}{S}\}_{s=1}^S\right)$, $k = 1, 2, \cdots, K$

- For $s = 1, 2, \cdots, S$, use (27) to compute the normalized weights, $w_n^{(s)}$, associated with the full joint particle, $\tilde{\mathbf{x}}_n^{(s)}$
- For $k = 1, 2, \cdots, K$, for $s = 1, 2, \cdots, S$,
  - For $s' = 1, 2, \cdots, S$, compute $\epsilon_{k,n}^{(s,s')} = w_n^{(s')}||\mathbf{y}_n - \mathbf{h}_n(\mathcal{Z}_{k,n}^{(s,s')})||^2_{\mathbf{R}_n^{-1}}$, where $\mathcal{Z}_{k,n}^{(s,s')}$ is given in (28)
  - Compute $\tilde{\lambda}_n^{k,(s)} = \exp\left(-\frac{1}{2}\sum_{s'=1}^S \epsilon_{k,n}^{(s,s')}\right)$
- For $k = 1, 2, \cdots, K$, for $s = 1, 2, \cdots, S$, normalize the weights as $\lambda_n^{k,(s)} = \frac{\tilde{\lambda}_n^{k,(s)}}{\sum_{s'=1}^S \tilde{\lambda}_n^{k,(s')}}$

Let $\mathcal{C}_{\mathbf{h}}$ denotes the cost of evaluating the function $\mathbf{h}_n(\mathbf{x}_n)$. The number of flops of each weighting step is approximately

$$\begin{aligned}\mathcal{C}^{\mathsf{W}}_{\mathsf{VBMPF}} &= \left[(S-1)\frac{n_{\mathbf{x}}}{n^k_{\mathbf{x}}}+1\right]S\mathcal{C}_{\mathbf{h}}+\left[(4n_{\mathbf{y}}+1)S^2-(4n_{\mathbf{y}}-4)S-1\right]\frac{n_{\mathbf{x}}}{n^k_{\mathbf{x}}}\\ &+ (4n_{\mathbf{y}}+3)S-1,\end{aligned} \tag{29}$$

if the measurement noise covariance $\mathbf{R}_n$ is diagonal, and

$$\begin{aligned}\mathcal{C}^{\mathsf{W}}_{\mathsf{VBMPF}} &= \left[(S-1)\frac{n_{\mathbf{x}}}{n^k_{\mathbf{x}}}+1\right]S\mathcal{C}_{\mathbf{h}}+\left[\left(n^2_{\mathbf{y}}+\frac{5}{2}n_{\mathbf{y}}+1\right)S^2-\left(n^2_{\mathbf{y}}+\frac{5}{2}n_{\mathbf{y}}-3\right)S-1\right]\frac{n_{\mathbf{x}}}{n^k_{\mathbf{x}}}\\ &+ \left(n^2_{\mathbf{y}}+\frac{5}{2}n_{\mathbf{y}}+3\right)S+\frac{2}{3}n^3_{\mathbf{y}}+\frac{3}{2}n^2_{\mathbf{y}}+\frac{5}{6}n_{\mathbf{y}}-1,\end{aligned} \tag{30}$$

if not.

Remark 2: Simplified versions of Algorithm 2 can be obtained in two particular cases:

- *Case of linear observation models.* In the case $\mathbf{h}_n(\mathbf{x}_n)=\mathbf{H}_n\mathbf{x}_n$, one can show that Eq. (28) reduces to [41]:

$$\lambda^{k,(s)}_n \propto \exp\left(-\frac{1}{2}\left[(\mathbf{H}^k_n\tilde{\mathbf{x}}^{k,(s)}_n)^T\mathbf{R}^{-1}_n\mathbf{H}^k_n\tilde{\mathbf{x}}^{k,(s)}_n-2(\mathbf{H}^k_n\tilde{\mathbf{x}}^{k,(s)}_n)^T\mathbf{R}^{-1}_n(\mathbf{y}_n-\mathbf{H}^{k-}_n\overline{\mathbf{x}}^{k-}_{n|n})\right]\right), \tag{31}$$

  where $\mathbf{H}^k_n$ and $\mathbf{H}^{k-}_n$ verify $\mathbf{H}_n\mathbf{x}_n=\mathbf{H}^k_n\mathbf{x}^k_n+\mathbf{H}^{k-}_n\mathbf{x}^{k-}_n$, and $\overline{\mathbf{x}}^{k-}_{n|n}=\sum_{s=1}^S w^{(s)}_n\tilde{\mathbf{x}}^{k-,(s)}_n$. In such a case, $\mathcal{C}_{\mathbf{h}}=(2n_{\mathbf{x}}-1)n_{\mathbf{y}}$ and $\mathcal{C}^{\mathsf{W}}_{\mathsf{VBMPF}}$ in (29) and (30) respectively become,

$$\mathcal{C}^{\mathsf{W}}_{\mathsf{VBMPF}}=2Sn_{\mathbf{y}}\frac{n^2_{\mathbf{x}}}{n_{\mathbf{x}^k}}+\left[2Sn_{\mathbf{y}}+\left(S^2+(4n_{\mathbf{y}}+4)S-1\right)\frac{1}{n_{\mathbf{x}^k}}\right]n_{\mathbf{x}}+(3n_{\mathbf{y}}+3)S-1, \tag{32}$$

  and

$$\begin{aligned}\mathcal{C}^{\mathsf{W}}_{\mathsf{VBMPF}} &= 2Sn_{\mathbf{y}}\frac{n^2_{\mathbf{x}}}{n_{\mathbf{x}^k}}+\left[2n_{\mathbf{y}}S+\left(S^2+(n^2_{\mathbf{y}}+n_{\mathbf{y}}+3)S-1\right)\frac{1}{n_{\mathbf{x}^k}}\right]n_{\mathbf{x}}+\left(n^2_{\mathbf{y}}+\frac{3}{2}n_{\mathbf{y}}+3\right)S\\ &+ \frac{2}{3}n^3_{\mathbf{y}}+\frac{3}{2}n^2_{\mathbf{y}}+\frac{5}{6}n_{\mathbf{y}}-1.\end{aligned} \tag{33}$$

- *Case of independent marginal likelihoods.* As stated above, the weights are computed following (28) by exchanging particles between the partitions. However, this is not needed when the likelihood is factorized as:

$$p(\mathbf{y}_n|\mathbf{x}_n)=\prod_{k=1}^{n_{\mathbf{y}}}p(\mathbf{y}^k_n|\mathbf{x}^k_n); \tag{34}$$

  $\mathbf{y}^k_n$ denotes the k^{th} partition of $\mathbf{y}_n$ with size $n_{\mathbf{y}^k}$ ($\mathbf{x}^k_n$ and $\mathbf{y}^k_n$ are not necessarily of the same size). Indeed, based on (34), starting from a separable approximation of the prediction pdf, $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$, the Bayes' rule leads to an approximation of the posterior, $p(\mathbf{x}_n|\mathbf{y}_{0:n})$, that is also separable. In

practice, this negates the need for Algorithm 2 since the weights, $\lambda_n^{k,(s)}$, can be separately computed using the marginal likelihoods only as[2]

$$\lambda_n^{k,(s)} \propto p(\mathbf{y}_n^k|\tilde{\mathbf{x}}_n^{k,(s)}), \quad k=1,\cdots,K; s=1,\cdots,S. \tag{35}$$

Based on factorization (34), system (17) reduces to a new system for which $\mathbf{h}_n^k(\mathbf{x}_n) = \mathbf{h}_n^k(\mathbf{x}_n^k)$, for any k, and $\mathbf{R}_n$ block diagonal with $(n_{\mathbf{y}^k} \times n_{\mathbf{y}^k})$ blocks. In this new system, the approximate number of flops for computing $\lambda_n^{k,(s)}$, for all k and s, becomes

$$\mathcal{C}_{\mathsf{VBMPF}}^{\mathsf{W}} = S\mathcal{C}_{\mathbf{h}} + (3S-1)\frac{n_{\mathbf{x}}}{n_{\mathbf{x}}^k}, \tag{36}$$

if the blocks of $\mathbf{R}_n$ are diagonal, and

$$\mathcal{C}_{\mathsf{VBMPF}}^{\mathsf{W}} = S\mathcal{C}_{\mathbf{h}} + (3S-1)\frac{n_{\mathbf{x}}}{n_{\mathbf{x}}^k} + \frac{2}{3K^2}n_{\mathbf{y}}^3 + \left(S+\frac{3}{2}\right)\frac{n_{\mathbf{y}}^2}{K} + \left(\frac{5}{2}S+\frac{5}{6}\right)n_{\mathbf{y}}, \tag{37}$$

otherwise.

C. Summary

The sampling (S), weighting (W), and resampling (R) steps of the VBMPF in a general nonlinear system (17) are summarized in Algorithm 3 below.

Algorithm 3 : $\{\mathbf{x}_n^{k,(s)}\}_{s=1}^S = \text{VBMPF}\left(\{\mathbf{x}_{n-1}^{(s)}\}_{s=1}^S\right)$, $k=1,2,\cdots,K$

For $n=0$:

- S step: Draw independently $\{\tilde{\mathbf{x}}_0^{(s)}\}_{s=1}^S$ from $\mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$
- W step: Use Algorithm 2 to compute the weights, $\lambda_0^{k,(s)}$, for $k=1,\cdots,K$ and $s=1,\cdots,S$
- R step: Draw independently $\{\mathbf{x}_0^{k,(s)}\}_{s=1}^S$ from $\sum_{s=1}^S \lambda_0^{k,(s)}\delta(\mathbf{x}_0^k - \tilde{\mathbf{x}}_0^{k,(s)})$, for $k=1,\cdots,K$

For $n \geq 1$:

- S step: Use Algorithm 1 to draw independently $\{\tilde{\mathbf{x}}_n^{k,(s)}\}_{s=1}^S$ starting from $\{\mathbf{x}_{n-1}^{(s)}\}_{s=1}^S$, for $k=1,\cdots,K$
- W step: Use Algorithm 2 to compute the weights, $\lambda_n^{k,(s)}$, for $k=1,\cdots,K$ and $s=1,\cdots,S$
- R step: Draw independently $\{\mathbf{x}_n^{k,(s)}\}_{s=1}^S$ from $\sum_{s=1}^S \lambda_n^{k,(s)}\delta(\mathbf{x}_n^k - \tilde{\mathbf{x}}_n^{k,(s)})$, for $k=1,\cdots,K$

[2]This reduces to K parallel weighting steps of the standard PF [11].

D. Discussions

The proposed VBMPF algorithm combines VB-like density approximations with MC sampling techniques. The (joint) filtering pdf of the full system state, $p(\mathbf{x}_n|\mathbf{y}_{0:n})$, is first approximated by a separable product of VB-based marginal filtering pdfs, $\prod_{k=1}^{K} q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$, by splitting the system, $(\mathbf{x}, \mathbf{y})$, into low-dimensional sub-systems, $(\mathbf{x}^k, \mathbf{y})$, $k = 1, 2, \cdots, K$. MC simulations are then used for each sub-system $(\mathbf{x}^k, \mathbf{y})$, leading to a PF algorithm approximating the associated (marginal) VB-like filtering distribution, $q(d\mathbf{x}_n^k|\mathbf{y}_{0:n})$, and the AM estimate of $\mathbf{x}_n^k$. The loss of posterior dependence between the state partitions, $\{\mathbf{x}_n^k\}_{k=1}^{K}$, is partially compensated by their functional dependence through the posterior expectations as in (16). The functional dependence (16) is represented in practice through the exchange of information between the (marginal) PFs. Indeed, in the weighting step, the computation of the weight (28) of a particle $\tilde{\mathbf{x}}_n^{k,(s)}$ uses the particles from the other subspaces, $\{\tilde{\mathbf{x}}_n^{k^-,(s')}\}_{s'=1}^{S}$. A similar remark holds in the sampling step in the calculation of the means of VB marginal transition pdfs, $\hat{\mu}_n^k(\mathbf{x}_{n-1}^{k,(s)})$, using (20).

As it is well known in the VB literature (see e.g. [42] [43] [25] [26]), it is difficult to clearly characterize the accuracy of VB-like approximations, since this depends on several factors, such as the number of partitions, K, and the strength of the dependencies between these partitions [42] [25]. However, one can intuitively examine the question of when the accuracy of VB approximation is likely to be reasonable and when one may expect it to fail. Clearly, when the state partitions are weakly dependent (*a posteriori*) of each other, the VB approximation should be viable and even nearly perfect if the partitions are almost independent. In our system (17), this can occur if each k^{th} sub-vector $\mathbf{f}_n^k(\mathbf{x}_n)$ of the state transition function and $\mathbf{h}_n^k(\mathbf{x}_n)$ of the measurement function are weakly dependent on the other state sub-vectors $\mathbf{x}_n^{k^-}$, and the covariances $\mathbf{Q}_n$ and $\mathbf{R}_n$ are block diagonal matrices whose diagonal blocks are $n_{\mathbf{x}^k} \times n_{\mathbf{x}^k}$ matrices. In contrast, when these partitions become strongly dependent, the "assumption" of conditional independence becomes too strong and this may degrade the accuracy of the approximation. Consequently, the number of partitions K should be chosen to achieve as good a trade-off as possible between accuracy and computational efficiency. The larger K is, the more important computational efficiency is, but also the more dependence between the state partitions will be dropped, and this can degrade the VB approximation.

Although VB approximations should (theoretically) improve with decreasing K (or with increasing state partitions dimension, $n_{\mathbf{x}^k}$), the MC approximations, in turn, require a large enough number of particles, S. Thereby, for a given S the rule of thumb is to carefully choose $n_{\mathbf{x}^k}$ large enough to achieve as accurate a VB density estimate as possible, but not too large to mitigate the weights degeneracy phenomena.

Finally, it is important to note that the prediction and filtering VB marginals, $q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$ and $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$, are not necessarily unimodal pdfs (as is often the case for VB-like approximations). Indeed, inserting the approximation $q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k,\mathbf{y}_{0:n-1}) \approx \mathcal{N}_{\mathbf{x}_n^k}(\hat{\mu}_n(\mathbf{x}_{n-1}^k),\mathbf{Q}_{n-1}^k)$ (see Section III-A1) in (15) reveals that $q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$ is approximated by a Gaussian mixture, $\frac{1}{S}\sum_{s=1}^{S}\mathcal{N}_{\mathbf{x}_n^k}(\hat{\mu}_n(\mathbf{x}_{n-1}^{k,(s)}),\mathbf{Q}_{n-1}^k)$, which is multimodal. On the other hand, following (16), multiplying the Gaussian mixture pdf $q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$ with the "likelihood" $\mathcal{L}_n(\mathbf{x}_n^k)$ (24), leads to a multimodal VB marginal filtering pdf, $q(\mathbf{x}_n^k|\mathbf{y}_{0:n})$. The case of unimodal (Gaussian) VB-like approximations has recently been addressed in [38] [30].

IV. Comparisons and extensions

In this section, we discuss similarities and differences between the VBMPF and existing MPF schemes. We further investigate the more general case for which the state transition pdf is not separable as in (10), and the case of state-space systems with multiplicative noises.

A. Comparison with other MPF schemes

Although the VBMPF shares the same (S,W,R) structure as the existing MPFs (see e.g. [19] [14]), these steps were however derived based on different mechanisms. Let's consider the MPF introduced in [14], whose performances are numerically compared in Section V against those of the proposed VBMPF. Starting from a filtering particle, $\mathbf{x}_{n-1}^{k,(s)}$, the S step of this filter generates J predicted particles (children), $\tilde{\mathbf{x}}_n^{k,(s,j)}$, while only one particle, $\tilde{\mathbf{x}}_n^{k,(s)}$, is generated in the VBMPF[3]. Consequently, the W and R steps consider SJ particles in the MPF and only S particles in the VBMPF. Furthermore, in the W step of the MPF, the weight of each $\tilde{\mathbf{x}}_n^{k,(s,j)}$ is computed by drawing L particles from the other subspaces $k' \neq k$ (i.e., $\{\overline{\mathbf{x}}_n^{k',(\ell)}\}_{\ell=1}^{L} \sim \frac{1}{SJ}\sum_{s=1}^{S}\sum_{j=1}^{J}\delta(\mathbf{x}_n^{k'} - \tilde{\mathbf{x}}_n^{k',(s,j)})$), and taking the average likelihood as (see [14, eq. (7)]):

$$\lambda_n^{k,(s,j)} \propto \frac{1}{L}\sum_{\ell=1}^{L} p(\mathbf{y}_n|\overline{\mathbf{x}}_n^{1,(\ell)},\cdots,\overline{\mathbf{x}}_n^{k'-1,(\ell)},\tilde{\mathbf{x}}_n^{k,(s,j)},\overline{\mathbf{x}}_n^{k'+1,(\ell)},\cdots,\overline{\mathbf{x}}_n^{K,(\ell)}). \tag{38}$$

Once the weighted set of particles, $\left\{\{\tilde{\mathbf{x}}_n^{k,(s,j)},\lambda_n^{k,(s,j)}\}_{j=1}^{J}\right\}_{s=1}^{S}$, is computed, a resampling step is then performed to bring back the number of SJ particles to S i.i.d. filtering particles, $\{\mathbf{x}_n^{k,(s)},\frac{1}{S}\}_{s=1}^{S}$. Each $\mathbf{x}_n^{k,(s)}$ can be drawn from the discrete pmf, $\sum_{s=1}^{S}\sum_{j=1}^{J}\lambda_n^{k,(s,j)}\delta(\mathbf{x}_n^k - \tilde{\mathbf{x}}_n^{k,(s,j)})$.

[3]To avoid any confusion, note that for each partition k and trajectory s, the VBMPF generates one predicted particle by inherently drawing J "particles" $\mathbf{z}_n^{k,(s,s_\ell)} = \mathbf{f}_{n-1}^k(\mathcal{X}_{k,n}^{(s_\ell)})$, then taking their average as in (20) (see also Algorithm 1).

The number of flops required by the MPF at each step in system (17) is approximately

$$\begin{aligned}\mathcal{C}_{\mathsf{MPF}} &= SJ\mathcal{C}_{\mathbf{f}} + SJL\frac{n_{\mathbf{x}}}{n_{\mathbf{x}}^k}\mathcal{C}_{\mathbf{h}} + \Big[3SJ + 1 + \frac{1}{n_{\mathbf{x}}^k}\left(SJL[4n_{\mathbf{y}}+2] + 2SJ - S - J\right.\\ &+ \left.\mathcal{C}_{\mathsf{R}}(S) + \mathcal{C}_{\mathsf{R}}(J) + 1\right)\Big] n_{\mathbf{x}}, \end{aligned} \tag{39}$$

if $\mathbf{R}_n$ is diagonal, and

$$\begin{aligned}\mathcal{C}_{\mathsf{MPF}} &= SJ\mathcal{C}_{\mathbf{f}} + SJL\frac{n_{\mathbf{x}}}{n_{\mathbf{x}}^k}\mathcal{C}_{\mathbf{h}} + \left[3SJ + 1 + \frac{1}{n_{\mathbf{x}}^k}\left(SJL\left[n_{\mathbf{y}}^2 + \frac{5}{2}n_{\mathbf{y}} + 2\right] + 2SJ - S - J\right.\right.\\ &+ \left.\left.\mathcal{C}_{\mathsf{R}}(S) + \mathcal{C}_{\mathsf{R}}(J) + 1\right)\right] n_{\mathbf{x}} + \frac{2}{3}n_{\mathbf{y}}^3 + \frac{3}{2}n_{\mathbf{y}}^2 + \frac{5}{6}n_{\mathbf{y}}, \end{aligned} \tag{40}$$

if not.

For a given state-space system, the comparison with the cost of the VBMPF,

$$\mathcal{C}_{\mathsf{VBMPF}} = \underbrace{\mathcal{C}_{\mathsf{VBMPF}}^{\mathsf{S}}}_{(21)} + \underbrace{\mathcal{C}_{\mathsf{VBMPF}}^{\mathsf{W}}}_{(29)-(30)} + \underbrace{K\mathcal{C}_{\mathsf{R}}(S)}_{\text{Resampling step}}, \tag{41}$$

depends on the parameters S, J, L and $n_{\mathbf{x}^k} = \frac{n_{\mathbf{x}}}{K}$. Let's consider the example of a high-dimensional linear Gaussian state-space system for which $n_{\mathbf{x}} = 1000$, $n_{\mathbf{y}} = 100$, $\mathbf{Q}_n$ and $\mathbf{R}_n$ are diagonal, $\mathcal{C}_{\mathbf{f}} = 2n_{\mathbf{x}}^2 - n_{\mathbf{x}} = 199000$ and $\mathcal{C}_{\mathbf{h}} = (2n_{\mathbf{x}} - 1)n_{\mathbf{y}} = 19900$. We assume that the resampling cost, $\mathcal{C}_{\mathsf{R}}(S)$, which can be of the order of S (see e.g. [44]), is equal to S. Table I outlines the ratio $\frac{\mathcal{C}_{\mathsf{MPF}}}{\mathcal{C}_{\mathsf{VBMPF}}}$ for different number of particles, S, J and L in full-factorization case, that is, when the system state is partitioned into $K = n_{\mathbf{x}}$ (scalar) components[4]. The proposed VBMPF is computationally cheaper than the MPF, and this becomes more pronounced when increasing S, J and L. For instance, the VBMPF is about three times cheaper when $S = 10$ and $J = L = 5$, and 100 times when $S = J = L = 100$.

Furthermore, in standard MPFs, the separation of the filtering pdf is not based on any optimization criteria, in contrast with the VBMPF. However, these filters still have the advantage of being applicable to systems for which the noises, $\mathbf{u}_n$ and $\mathbf{v}_n$, follow any type of distributions, not necessarily Gaussian.

B. Extension to a non separable state transition pdf

The derivation of the VBMPF assumes, through (10), that the state transition pdf, $p(\mathbf{x}_n|\mathbf{x}_{n-1})$, is separable. In other words, the marginals of the system state at a given time n, $\{\mathbf{x}_n^k\}_{k=1}^K$, are conditionally

[4]Note that, on the one hand, results very close to those in Table I have been obtained in the case of a non-diagonal matrix, $\mathbf{R}$, and on the other hand, the overall behavior of $\mathcal{C}_{\mathsf{VBMPF}}$ w.r.t. $\mathcal{C}_{\mathsf{MPF}}$ remains unchanged for any state partition size, $n_{\mathbf{x}^k}$, and number of particles, S, verifying $S \geq n_{\mathbf{x}^k}$. Such a condition (number of samples bigger than their dimension) should be fulfilled in any Monte Carlo-based scheme (law of large numbers).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSP.2016.2580524, IEEE Transactions on Signal Processing

Table I: The ratio $\frac{\mathcal{C}_{MPF}}{\mathcal{C}_{VBMPF}}$ in full-factorization case for linear Gaussian systems with $n_{\mathbf{x}} = 1000$, $n_{\mathbf{y}} = 100$, and diagonal covariances, $\mathbf{Q}_n$ and $\mathbf{R}_n$.

S	$(J = \frac{S}{2}\ ,\ L = \frac{S}{2})$	$(J = S\ ,\ L = \frac{S}{2})$	$(J = \frac{S}{2}\ ,\ L = S)$	$(J = S\ ,\ L = S)$
10	2.77	5.5	5.53	11
20	5.24	10.43	10.48	20.84
50	12.7	25.26	25.39	50.52
100	25.13	50	50.26	100

independent based on the state at previous time, $\mathbf{x}_{n-1}$. This assumption does not remove dependence between marginal states, $\{\mathbf{x}_n^k\}_{k=1}^K$, since $p(\mathbf{x}_n) \neq \prod_{k=1}^K p(\mathbf{x}_n^k)$ still holds.

In the more general case of a non-separable state transition pdf; that is when (10) does not hold, the generic algorithm introduced in Section II (and its MC implementation of Section III) is no longer valid. One way to overcome this issue is to use the VB approach to approximate the state transition pdf, $p(\mathbf{x}_n|\mathbf{x}_{n-1})$, with a separable product of marginal pdfs, $\prod_{k=1}^K q(\mathbf{x}_n^k|\mathbf{x}_{n-1})$. Similar to (8) and using $p(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}_{\mathbf{x}_n}(\mathbf{f}_{n-1}(\mathbf{x}_{n-1}), \mathbf{Q}_{n-1})$ (see (17)), one obtains

$$\begin{aligned} q(\mathbf{x}_n^k|\mathbf{x}_{n-1}) &\propto \exp\Big(\mathbb{E}_{q(\mathbf{x}_n^{k^-}|\mathbf{x}_{n-1})}\left[\ln(p(\mathbf{x}_n|\mathbf{x}_{n-1}))\right]\Big), \qquad (42)\\ &\propto \exp\left(-\frac{1}{2}\left[||\mathbf{x}_n^k||^2_{\Lambda_{n-1}^k} - 2\left(\mathbf{x}_n^k\right)^T\Big[\Lambda_{n-1}(k,:)\mathbf{f}_{n-1}(\mathbf{x}_{n-1}) - \Lambda_{n-1}(k,k^-)\mathbb{E}_{q(\mathbf{x}_n^{k^-}|\mathbf{x}_{n-1})}[\mathbf{x}_n^{k^-}]\Big]\right]\right), \end{aligned}$$

where $\Lambda_{n-1} = \mathbf{Q}_{n-1}^{-1}$ ($\mathbf{Q}_{n-1}$ is no longer block diagonal), Λ_{n-1}^k denotes the k^{th} $n_{\mathbf{x}^k} \times n_{\mathbf{x}^k}$ block diagonal of Λ_{n-1}, $\Lambda_{n-1}(k,:)$ a $n_{\mathbf{x}^k} \times n_{\mathbf{x}}$ matrix formed by $n_{\mathbf{x}^k}$ rows of Λ_{n-1} starting from the k^{th} row, and $\Lambda_{n-1}(k,k^-)$ the complementary part of Λ_{n-1}^k in $\Lambda_{n-1}(k,:)$. Approximating $\mathbb{E}_{q(\mathbf{x}_n^{k^-}|\mathbf{x}_{n-1})}[\mathbf{x}_n^{k^-}]$ with $\mathbb{E}_{p(\mathbf{x}_n^{k^-}|\mathbf{x}_{n-1})}[\mathbf{x}_n^{k^-}] = \mathbf{f}_{n-1}^{k^-}(\mathbf{x}_{n-1})$, one eventually gets,

$$q(\mathbf{x}_n^k|\mathbf{x}_{n-1}) \approx \mathcal{N}_{\mathbf{x}_n^k}\left(\mathbf{f}_{n-1}^k(\mathbf{x}_{n-1}), \left(\Lambda_{n-1}^k\right)^{-1}\right),\ k = 1, 2, \cdots, K. \qquad (43)$$

In (43), the covariances $\left(\Lambda_{n-1}^k\right)^{-1}$, which are not equal to $\mathbf{Q}_{n-1}^k$, are computed from all entries of $\mathbf{Q}_{n-1}$. This is important since it emphasizes the fact that although the covariance of the approximated transition pdf, $\prod_{k=1}^K q(\mathbf{x}_n^k|\mathbf{x}_{n-1})$, is block diagonal, it nevertheless involves all off-diagonal entries of the original covariance, $\mathbf{Q}_{n-1}$.

Now, comparing (43) with (18), one can see that the VBMPF algorithm is still valid by replacing $\mathbf{Q}_{n-1}^k$ by $\left(\Lambda_{n-1}^k\right)^{-1}$. The cost of computing $\left(\Lambda_{n-1}^k\right)^{-1}$ is of the order $\mathcal{O}(n_{\mathbf{x}}^3)$ because this requires inverting the

$n_{\mathbf{x}} \times n_{\mathbf{x}}$ covariance matrix, $\mathbf{Q}_{n-1}$. In situations for which $\mathbf{Q}_{n-1}$ is invariant in time, it must be inverted only once (at the initial time only), and then the sub-matrices $\left(\Lambda_0^k\right)^{-1} = \left(\Lambda^k\right)^{-1}$ for $k = 1, 2, \cdots, K$, can be computed and stored for subsequent use in the filtering algorithm.

C. Systems with multiplicative noises

The proposed VBMPF algorithm considers a state-space system (17) with additive noises. Here, we address the case for which (at least) one of these noises is multiplicative.

- In the case of a multiplicative input process noise; that is when[5]

$$\mathbf{x}_n = \mathbf{f}_{n-1}(\mathbf{x}_{n-1}) \circ \mathbf{u}_{n-1}, \tag{44}$$

where $\circ$ denotes the Hadamard product, the sampling step (Algorithm 1) does not hold since (18), (19) and (20) are no longer valid. Indeed, the state transition pdf becomes

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}_{\mathbf{x}_n}\left(\mathbf{0}, \widetilde{\mathbf{Q}}_{n-1}(\mathbf{x}_{n-1})\right), \tag{45}$$

with $\widetilde{\mathbf{Q}}_{n-1}(\mathbf{x}_{n-1}) = \mathrm{Diag}\left(\mathbf{f}_{n-1}(\mathbf{x}_{n-1})\right).\mathbf{Q}_{n-1}.\mathrm{Diag}\left(\mathbf{f}_{n-1}(\mathbf{x}_{n-1})\right)$; $\mathrm{Diag}(\mathbf{v})$ denotes a diagonal matrix with diagonal $\mathbf{v}$. Unlike the additive noise case, the covariance $\widetilde{\mathbf{Q}}_{n-1}$ depends on the previous state, $\mathbf{x}_{n-1}$. Now, based on (45), one can verify that (18) becomes

$$p(\mathbf{x}_n^k|\mathbf{x}_{n-1}) = \mathcal{N}_{\mathbf{x}_n^k}\left(\mathbf{0}, \widetilde{\mathbf{Q}}_{n-1}^k(\mathbf{x}_{n-1})\right). \tag{46}$$

Inserting (46) in (13), one obtains

$$q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1}) = \mathcal{N}_{\mathbf{x}_n^k}\left(\mathbf{0}, \Gamma_{n-1}(\mathbf{x}_{n-1}^k)\right), \tag{47}$$

with $\Gamma_{n-1}(\mathbf{x}_{n-1}^k) = \mathbb{E}_{q(\mathbf{x}_{n-1}^{k^-}|\mathbf{y}_{0:n-1})}\left[\widetilde{\mathbf{Q}}_{n-1}^k(\mathbf{x}_{n-1})\right]$. Unlike the additive noise case, the expectation w.r.t. $q(\mathbf{x}_{n-1}^{k^-}|\mathbf{y}_{0:n-1})$ is involved in the covariance of $q(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k, \mathbf{y}_{0:n-1})$ instead of the mean (see (19) and (20)). Similarly to (20), the covariance in (47) can be approximated using the MC representation of $q(d\mathbf{x}_{n-1}^{k^-}|\mathbf{y}_{0:n-1})$, $\sum_{s=1}^{S} \frac{1}{S}\delta_{\mathbf{x}_{n-1}^{k^-,(s)}}(d\mathbf{x}_{n-1}^{k^-})$. One obtains for $k = 1, 2, \cdots, K$,

$$\widehat{\Gamma}_{n-1}(\mathbf{x}_{n-1}^k) = \frac{1}{J}\sum_{\ell=1}^{J} \widetilde{\mathbf{Q}}_{n-1}^k(\mathcal{X}_{k,n-1}^{(s_\ell)}), \tag{48}$$

[5]Without loss of generality, we consider a conditionally independent multiplicative noise model (44). The extension to the more general model, $\mathbf{x}_n = \mathbf{F}_{n-1}(\mathbf{x}_{n-1}).\mathbf{u}_{n-1}$, where $\mathbf{F}_{n-1}(.)$ is any $n_{\mathbf{x}} \times n_{\mathbf{x}}$ function, is trivial. The same remark holds for the measurement noise case below (see (49)).

where $\mathcal{X}_{k,n-1}^{(s_\ell)} \overset{\text{def}}{=} \left[(\mathbf{x}_{n-1}^{1,(s_\ell)})^T,\cdots,(\mathbf{x}_{n-1}^{k-1,(s_\ell)})^T,(\mathbf{x}_{n-1}^{k})^T,(\mathbf{x}_{n-1}^{k+1,(s_\ell)})^T,\cdots,(\mathbf{x}_{n-1}^{K,(s_\ell)})^T\right]^T$ and each sub-vector $\mathbf{x}_{n-1}^{j,(s_\ell)}$ with $j \neq k$ is drawn from the pmf, $\sum_{s=1}^{S} \frac{1}{S}\delta(\mathbf{x}_{n-1}^{j}-\mathbf{x}_{n-1}^{j,(s)})$. Finally, using the hierarchical sampling technique, one can independently sample the particles $\{\tilde{\mathbf{x}}_n^{k,(s)}\}_{s=1}^{S}$ from $q(\mathbf{x}_n^k|\mathbf{y}_{0:n-1})$ by drawing each $\tilde{\mathbf{x}}_n^{k,(s)}$ from $\mathcal{N}\left(\mathbf{0},\widehat{\Gamma}_{n-1}(\mathbf{x}_{n-1}^{k,(s)})\right)$.

- In the case of a multiplicative measurement noise, that is when

$$\mathbf{y}_n = \mathbf{h}_n(\mathbf{x}_n) \circ \mathbf{v}_n, \tag{49}$$

the weighting step (Algorithm 2) does not hold because (23)-(24) and (27)-(28) are no longer valid. Indeed, the likelihood (23) becomes

$$p(\mathbf{y}_n|\mathbf{x}_n) = \mathcal{N}_{\mathbf{y}_n}\left(\mathbf{0},\widetilde{\mathbf{R}}_n(\mathbf{x}_n)\right), \tag{50}$$

with $\widetilde{\mathbf{R}}_n(\mathbf{x}_n) = \text{Diag}\left(\mathbf{h}_n(\mathbf{x}_n)\right).\mathbf{R}_n.\text{Diag}\left(\mathbf{h}_n(\mathbf{x}_n)\right)$. Again, unlike the additive noise case, the covariance $\widetilde{\mathbf{R}}_n$ depends on the state, $\mathbf{x}_n$. Now, based on (50), one can verify that (24) becomes

$$\mathcal{L}(\mathbf{x}_n^k) \propto \exp\left(-\frac{1}{2}\mathbb{E}_{q(\mathbf{x}_n^{k-}|\mathbf{y}_{0:n})}\left[\ln\left(\det\left(\widetilde{\mathbf{R}}_n(\mathbf{x}_n)\right)\right) + ||\mathbf{y}_n||^2_{\widetilde{\mathbf{R}}_n^{-1}(\mathbf{x}_n)}\right]\right). \tag{51}$$

Similarly to Section III-B, the expectation in (51) can be approximated using the available prediction samples, $\{\tilde{\mathbf{x}}_n^{k,(s)}\}_{s=1}^{S}$, for $k = 1, 2, \cdots, K$. The weight of each particle $\tilde{\mathbf{x}}_n^{k,(s)}$ is then given by

$$\lambda_n^{k,(s)} \propto \exp\left(-\frac{1}{2}\sum_{s'=1}^{S} w_n^{(s')}\left[\ln\left(\det\left(\widetilde{\mathbf{R}}_n\left(\mathcal{Z}_{k,n}^{(s,s')}\right)\right)\right) + ||\mathbf{y}_n||^2_{\widetilde{\mathbf{R}}_n^{-1}\left(\mathcal{Z}_{k,n}^{(s,s')}\right)}\right]\right), \tag{52}$$

where $\sum_{s=1}^{S}\lambda_n^{k,(s)} = 1$; $w_n^{(s)} \propto \mathcal{N}_{\mathbf{y}_n}\left(\mathbf{0},\widetilde{\mathbf{R}}_n\left(\tilde{\mathbf{x}}_n^{(s)}\right)\right)$ with $\sum_{s=1}^{S} w_n^{(s)} = 1$; and $\mathcal{Z}_{k,n}^{(s,s')} \overset{\text{def}}{=} \left[(\tilde{\mathbf{x}}_n^{1,(s')})^T, \cdots,(\tilde{\mathbf{x}}_n^{k-1,(s')})^T,(\tilde{\mathbf{x}}_n^{k,(s)})^T,(\tilde{\mathbf{x}}_n^{k+1,(s')})^T,\cdots,(\tilde{\mathbf{x}}_n^{K,(s')})^T\right]^T$.

V. Simulations

In this section, we perform numerical experiments to assess the performance of the proposed VBMPF filter against the standard PF [11] and the MPF [14]. We consider a vectorial version of a nonlinear system extensively used in the PF literature (see e.g. [6] [11] [12]):

$$\begin{cases} x_{n+1}^k = 0.5x_n^k + \frac{25\sum_{j=1}^{n_\mathbf{x}} x_n^j}{1+\left(\sum_{j=1}^{n_\mathbf{x}} x_n^j\right)^2} + 8\cos(1.2(n+1)) + u_n^k, \\ y_n^k \quad = \frac{(x_n^k)^2}{20} + v_n^k, \end{cases} \tag{53}$$

where x_n^k denotes the k^{th} scalar component of $\mathbf{x}_n$ (the other components are defined similarly), $N = 49$ (50 observations), $n_\mathbf{x} = n_\mathbf{y} = 40$, $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, 10 \times \mathbb{I})$, $\mathbf{u}_n \sim \mathcal{N}(\mathbf{0},\mathbf{Q})$ with $\mathbf{Q} = 10 \times \mathbb{I}$, and $\mathbf{v}_n \sim \mathcal{N}(\mathbf{0},\mathbf{R})$. All our results are averaged over 30 independent realizations (with different initial states). Two cases will be addressed depending on whether the measurement noise covariance, $\mathbf{R}$, is diagonal or not.

A. The case of $\mathbf{R}$ diagonal

We consider $\mathbf{R} = \mathbb{I}$. We start by using the VBMPF and the MPF with full-factorization (i.e., $K = n_{\mathbf{x}}$ components, x_n^k). Based on the diagonal property of $\mathbf{R}_n$ one has:

$$p(\mathbf{y}_n|\mathbf{x}_n) = \prod_{k=1}^{n_{\mathbf{x}}} p(y_n^k|x_n^k). \tag{54}$$

Thus, following Remark 2, Algorithm 2 is not needed to compute the weights since these can be computed separately using the marginal likelihoods as:

$$\lambda_n^{k,(s)} \propto p(y_n^k|\tilde{x}_n^{k,(s)}), \quad k = 1, \cdots, n_{\mathbf{x}}; s = 1, \cdots, S.$$

Instead we apply the SIR scheme to the Bayesian step associated with each state component, $p(x_n^k|y_{0:n}^k) \propto p(y_n^k|x_n^k)p(x_n^k|y_{0:n-1}^k)$, $k = 1, \cdots, n_{\mathbf{x}}$. This independent weighting mechanism is also used in the MPF [14].

We use $S = 20$ particles in the proposed VBMPF and the MPF, and $S = 20$ and 800 (= $20 \times n_{\mathbf{x}}$) for the standard PF. In all our experiments, J, which, let recall, is used as the number of resampled predicted particles in the VBMPF, and the number of children in the MPF, is equal to S; the number L of resampled predicted particles in the MPF is also assumed to be equal to S. Figure 1 plots the time-evolution of the MSE of the approximate AM estimate of the system state, $\mathbf{x}_n$. As one can see, the standard PF suffers from the large dimension of the system, requiring more than 800 particles to reach the accuracy of the MPF and VBMPF with only 20 particles. The MPF performs slightly better than the VBMPF. Such an outperformance can be explained by the fact that, on the one hand, the state variables in system (53) are weakly dependent, supporting the (free) independence assumption of the filtering pdf in the MPF, and on the other hand, the VB density estimation in the VBMPF becomes no longer necessary, and even seems to introduce extra errors when combined with another approximate SIR scheme.

In the following, we investigate the sensitivity of the VBMPF and MPF to changes in the number of particles, S, and the dimension of state partitions, $n_{\mathbf{x}^k}$, which recall is inversely proportional to their number, K. For this purpose, we use the empirical standard deviation defined by

$$\mathcal{D} = \frac{1}{N}\sum_{n=0}^{N-1}\left[\frac{1}{30}\sum_{j=1}^{30}||\overline{\mathbf{x}}_{n|n}(j) - \mathbf{x}_n(j)||^2\right]^{\frac{1}{2}},$$

where $\mathbf{x}_n(j)$ is the state at the j^{th} realization of the system, and $\overline{\mathbf{x}}_{n|n}(j)$ its estimate computed by one of the filtering algorithms ($\mathcal{D}$ is averaged over 30 independent realizations, and $N = 50$ time indices).

Table II outlines the standard deviation of the state estimates using the proposed VBMPF and the MPF with full-factorization, and the PF, as a function of the number of particles. Overall, the performances of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSP.2016.2580524, IEEE Transactions on Signal Processing

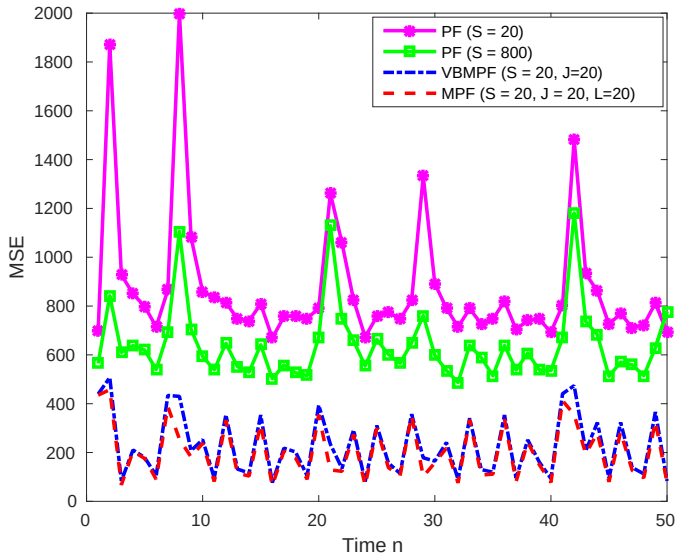

Figure 1: MSE of the state estimate as it results from the PF, and the MPF and VBMPF with full-factorization.

all filters improve with increased number of particles. One can further see that for any number of particles, the MPF is slightly more performant than the VBMPF, which, in turn, is widely more performant than the PF, which is consistent with Figure 1. On the other hand, the performances of the VBMPF and the MPF tend to be closer with increased number on particles; for instance, the difference between the errors $\mathcal{D}$ suggested by these filters is 2.49 when $S = 5$, and 0.26 only when $S = 100$.

Table II: Standard deviation $\mathcal{D}$ of the state estimates using the VBMPF and the MPF with full-factorization, and the PF.

S	5	10	15	20	25	30	50	100
VBMPF	17.82	15.16	14.9	14.5	14.3	14	13.81	13.63
MPF	15.33	13.89	13.81	13.6	13.66	13.37	13.46	13.37
PF	30.91	30.04	29.4	28.67	28.96	28.47	27.98	27.6

Table III presents the standard deviation of the state estimates using the proposed VBMPF and the MPF with $S = 20$ particles, as a function of the dimensions of partitions, $n_{\mathbf{x}^k}$, and the PF with $S = 20$ particles. For any state partition, the MPF behaves slightly better than the VBMPF, which, in turn, is more performant than the PF. Furthermore, overall, for the given number of particles, the two former

filters tend to be less performant when increasing the dimension of partitions. For system (53), this is due to the need of more particles to deal with the increase of $n_{\mathbf{x}^k}$, as it is well known in PF-based algorithms.

Table III: Standard deviation $\mathcal{D}$ of the state estimates using the VBMPF, the MPF and the PF with $S = 20$ particles.

$n_{\mathbf{x}^k} = \frac{n_{\mathbf{x}}}{K}$	1	2	5	8	10	20
VBMPF	14.5	15.45	19.62	22.03	23.33	26.71
MPF	13.6	13.58	14.76	17.12	18.33	22.68
PF	28.67	28.67	28.67	28.67	28.67	28.67

B. The case of $\mathbf{R}$ *non-diagonal*

Now, let's consider the more complex measurement equation,

$$\mathbf{y}_n = \frac{\mathbf{x}_n \circ \mathbf{x}_n}{20} + \mathbf{v}_n, \tag{55}$$

where the covariance of the measurement noise, $\mathbf{v}_n$, is no longer diagonal and whose entries are defined as

$$R(i,j) = \exp\left(-\frac{|i-j|}{100}\right); \quad i,j = 1,2,\cdots,n_{\mathbf{y}}. \tag{56}$$

Unlike the 2^{nd} eq. of system (53), the measurement components, $\{y_n^k\}_{k=1}^{n_{\mathbf{y}}}$, are not independent conditionally on the state, $\mathbf{x}_n$, and the factorization (54) of the likelihood, $p(\mathbf{y}_n|\mathbf{x}_n)$, is no longer valid (*i.e.*, $p(\mathbf{y}_n|\mathbf{x}_n) \neq \prod_{k=1}^{n_{\mathbf{x}}} p(y_n^k|x_n^k)$). The filtering pdf, $p(\mathbf{x}_n|\mathbf{y}_{0:n})$, is thus no longer separable even when starting from a separable prediction pdf, $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$, and thereby the separability assumption of $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ in the MPF becomes too severe. Table IV outlines the standard deviation of the state estimates as suggested by the VBMPF and the MPF with different numbers of particles and different dimensions of state partitions. The ratios, $\frac{\mathcal{C}_{\text{MPF}}}{\mathcal{C}_{\text{VBMPF}}}$, are also given to compare between the computational complexities of these algorithms; these are computed based on (40), (41), $\mathcal{C}_{\mathbf{f}} = 3n_{\mathbf{x}} + 8$ and $\mathcal{C}_{\mathbf{h}} = 2n_{\mathbf{x}}$. One can see that almost all errors in Table IV are much larger than those of Tables II and III. This can be explained by the fact that the state components are more dependent in the new system with a non-diagonal $\mathbf{R}$, a situation for which the separability assumption of the filtering pdf is less favourable. One can further see that the VBMPF suggests lower errors $\mathcal{D}$ than the MPF, in contrast with the diagonal case. In the non-diagonal case, the VBMPF benefits from the VB optimization criteria in the density estimation stage compared

to the MPF. In addition, similarly to the diagonal case, for any partition dimension, the performances of both filters improve with an increasing number of particles, S. Beside the better performances of the VBMPF compared with the MPF, the former is also always computationally more relevant and it can be up to 100 times cheaper (when $S = 100$).

On the other hand, the values of $n_{\mathbf{x}^k}$ that achieve the lowest standard deviation (bold values) increase with S. As stated in Section III-D, this is due to the back-to-back approximation (first VB density approximation followed by a PF approximation) on which these filters are based. The VB approximation (or the free separability assumption in the MPF) is improved with increasing $n_{\mathbf{x}^k}$, while the PF approximation is improved with increasing S. Moreover, to mitigate the degeneracy phenomena (since each state partition corresponds to one PF-based algorithm), S should be very large when $n_{\mathbf{x}^k}$ becomes large enough. To illustrate, for $S = 5$, the minimum of $\mathcal{D}$ is reached when $n_{\mathbf{x}^k} = 1$, that is, when the PF-like algorithms are applied on scalar sub-systems only; a set of 5 particles is too small to prevent the degeneracy problem for the vectorial sub-systems with $n_{\mathbf{x}^k} = 2, 5, \cdots$. For $S = 10$, the minimum of $\mathcal{D}$ is reached when $n_{\mathbf{x}^k} = 5$, while more particles are needed for larger state partitions.

To compare with the PF, we display in Table V the minimum values of $\mathcal{D}$ given in Table IV (bold values) and the values of $\mathcal{D}$ resulting from the PF. In contrast with the case of $\mathbf{R}_n$ diagonal (Table II), the PF behaves better than the MPF for all tested numbers of particles, S. As already discussed, this is due to the fact that the state partitions are more dependent in the new system with a non-diagonal $\mathbf{R}_n$, a situation in which the free separability assumption of the filtering pdf is less favourable. The proposed VBMPF still suggests smaller errors than the PF when S is less than 100 (this is confirmed by other tests that have been done with $S = 50$ and 90). For instance, for $S = 20$, the VBMPF with $n_{\mathbf{x}^k} = 8$ outperforms the PF, which, in turn, outperforms the MPF in its "optimal condition" (when $n_{\mathbf{x}^k} = 5$). In other words, the PF needs to be implemented with at least about 100 particles (i.e., with about $2.5 \times n_{\mathbf{x}}$ particles) to behave better than the proposed VBMPF.

The benefit of the VB optimization criteria becomes more pronounced when the observation equation in (53) becomes linear as $y_n^k = 0.5x_n^k + v_n^k$. Indeed, as suggested in Tables VI and VII, which are similar to Tables IV and V, respectively, the proposed VBMPF with $S = 10, 20, 100$ particles significantly outpeforms the MPF. More precisely, although the VBMPF seems to need more particles for large partitions (for $n_{\mathbf{x}^k} = 10, 20$), compared with the MPF, the lowest error $\mathcal{D}$ suggested for each S by this filter is still much smaller than that resulting from the MPF. For instance, for $S = 10$, the minimum error

Table IV: Standard deviation $\mathcal{D}$ of the state estimates using the VBMPF and the MPF with different number of particles, S, and partitions dimensions, $n_{\mathbf{x}^k}$. The minimum value of $\mathcal{D}$, for each S (row), is designated in bold. The computational costs of these algorithms are compared using $\frac{\mathcal{C}_{\text{MPF}}}{\mathcal{C}_{\text{VBMPF}}}$.

$n_{\mathbf{x}^k} = \frac{n_{\mathbf{x}}}{K}$	1	2	5	8	10	20
VBMPF ($S = 5$)	**26.46**	26.94	26.81	26.74	27.26	27.27
MPF ($S = 5$)	**28.28**	28.66	28.74	28.67	28.96	29.05
$\frac{\mathcal{C}_{\text{MPF}}}{\mathcal{C}_{\text{VBMPF}}}$ ($S = 5$)	6.03	5.84	5.33	4.91	4.68	3.82
VBMPF ($S = 10$)	25.67	25.87	**25.52**	26.09	25.67	25.53
MPF ($S = 10$)	26.83	26.82	**26.68**	27.18	27	26.84
$\frac{\mathcal{C}_{\text{MPF}}}{\mathcal{C}_{\text{VBMPF}}}$ ($S = 10$)	10.99	10.86	10.5	10.19	9.96	9.05
VBMPF ($S = 20$)	24.96	24.92	24.92	**24.79**	24.85	25.07
MPF ($S = 20$)	25.9	25.98	**25.66**	25.84	25.81	25.87
$\frac{\mathcal{C}_{\text{MPF}}}{\mathcal{C}_{\text{VBMPF}}}$ ($S = 20$)	20.72	20.84	20.52	20.21	20	19.05
VBMPF ($S = 100$)	24.52	24.51	24.49	24.48	**24.47**	24.5
MPF ($S = 100$)	25.31	25.29	25.11	25.2	25.18	**25.09**
$\frac{\mathcal{C}_{\text{MPF}}}{\mathcal{C}_{\text{VBMPF}}}$ ($S = 100$)	100.79	100.52	99.73	98.94	98.43	95.49

suggested by the VBMPF ($\mathcal{D} = 15.67$) is smaller than the errors of the MPF associated to all possible[6] $n_{\mathbf{x}^k}$, in particular the smallest one ($\mathcal{D} = 23.2$). In addition, the VBMPF is computationally much cheaper than the MPF, with ratios $\frac{\mathcal{C}_{\text{MPF}}}{\mathcal{C}_{\text{VBMPF}}}$ in the same order as those indicated in Table IV.

Regarding the behavior of the VBMPF w.r.t. the PF, one can see from Table VII that the proposed filter significantly outperforms the PF for $S = 10, 20, 100$ particles. Furthermore, the PF requires much more than 40000 particles (i.e., $1000 \times n_{\mathbf{x}}$ particles) to reach the performances of the proposed VBMPF with only 10 particles (i.e., $\frac{1}{4} \times n_{\mathbf{x}}$ particles).

[6]Note that the minima in Table VI remain unchanged when including the errors associated to $n_{\mathbf{x}^k} = 2, 8$.

Table V: Standard deviation $\mathcal{D}$ of the state estimates for different number of particles. The values of $\mathcal{D}$ associated to VBMPF and MPF are the minima obtained in Table IV (values in bold); their corresponding $n_{\mathbf{x}^k}$ are also indicated.

S	5	10	20	100
VBMPF	26.46 ($n_{\mathbf{x}^k}=1$)	25.52 ($n_{\mathbf{x}^k}=5$)	24.79 ($n_{\mathbf{x}^k}=8$)	24.47 ($n_{\mathbf{x}^k}=10$)
MPF	28.28 ($n_{\mathbf{x}^k}=1$)	26.68 ($n_{\mathbf{x}^k}=5$)	25.66 ($n_{\mathbf{x}^k}=5$)	25.09 ($n_{\mathbf{x}^k}=20$)
PF	27.66	26.09	25.20	24.45

Table VI: Standard deviation $\mathcal{D}$ of the state estimates using the VBMPF and the MPF with different number of particles, S, and partitions dimensions, $n_{\mathbf{x}^k}$. The minimum value of $\mathcal{D}$, for each S (row), is designated in bold.

$n_{\mathbf{x}^k}=\frac{n_{\mathbf{x}}}{K}$	1	5	10	20
VBMPF ($S=10$)	**15.67**	20.1	23.51	27.47
MPF ($S=10$)	23.21	**23.2**	23.22	24.04
VBMPF ($S=20$)	**14.57**	17.38	21.66	25.55
MPF ($S=20$)	21.78	**21.36**	21.56	21.98
VBMPF ($S=100$)	14.18	**13.78**	18.04	22.28
MPF ($S=100$)	18.96	19.73	**18.25**	19.06

VI. Conclusion

We considered the Bayesian filtering problem for high-dimensional state-space systems. We adopted the variational Bayesian (VB) approach to propose a unified and general framework for approximating the filtering pdf of the full state as a product of independent marginal pdfs under the Kullback-Leibler divergence minimization criteria. Two classical random sampling techniques were then used to derive Monte Carlo (MC) representations of the approximate VB-like marginal distributions. While the proposed VB multiple particle filter (VBMPF) shares the same structure as the standard MPF, inasmuch as one PF is suggested for each state partition and that all PFs feed each other by exchanging information

Table VII: Standard deviation $\mathcal{D}$ of the state estimates for different number of particles. The values of $\mathcal{D}$ associated to VBMPF and MPF are the minima obtained in Table VI (values in bold); their corresponding $n_{\mathbf{x}^k}$ are also indicated.

S	10	20	100	40000
VBMPF	15.67 ($n_{\mathbf{x}^k} = 1$)	14.57 ($n_{\mathbf{x}^k} = 1$)	13.78 ($n_{\mathbf{x}^k} = 5$)	x
MPF	23.2 ($n_{\mathbf{x}^k} = 5$)	21.36 ($n_{\mathbf{x}^k} = 5$)	18.25 ($n_{\mathbf{x}^k} = 10$)	x
PF	28.86	27.9	26.08	20.23

through the particles, however, the mechanisms of exchanging information between PFs in both filters are different.

Starting from one (filtering) particle, the sampling step of the VBMPF generates one single (predicted) particle and thereby this filter is computationally more efficient than the MPF, for which a set of (predicted) particles needs to be generated. The performance of the proposed VBMPF was evaluated and compared against those of the standard PF and a MPF. The results suggest that the behavior of the three filters depend on the strength of dependencies between the state partitions. The MPF is more performant in weakly dependent scenarios, whereas the advantage of the VBMPF becomes more pronounced when the system partitions are more dependent, especially when the observation operator is linear. Indeed, for the studied 40-dimensional system, to reach the performances of the proposed VBMPF with 10 particles, the MPF requires more than 100 particles, while the PF requires more than 40000 particles.

The VB approach could be further used to address the unsupervised case for which one or some hyper-parameters of the system, like for instance the measurement noise covariance, is unknown. Applying the unsupervised scheme to a real-world problem, in which some (or all) of the hyper-parameters are not known, and thus need to be estimated along with the state, would then be of great interest. Another line of important research would be to work on establishing theoretical results on the convergence of the VBMPF.

APPENDIX A

The computational complexities we computed considered each addition, subtraction, multiplication, division, square-root, or exponential operation as one flop [45] [38]. They are based on:

- $2d-1$ flops to compute the inner product, $\mathbf{u}^T.\mathbf{v}$, of two $d\times 1$ vectors, $\mathbf{u}$ and $\mathbf{v}$;
- $\frac{d^3}{3}+d^2+\frac{2}{3}d$ flops to compute a square-root (and lower triangular or upper triangular) matrix of a $d\times d$ matrix using the Cholesky decomposition;
- $\frac{d^3}{3}+\frac{d^2}{2}+\frac{d}{6}$ flops to invert a lower (or upper) triangular matrix;
- assuming that drawing one sample from $\mathcal{N}(0,1)$ is equivalent to one flop.

APPENDIX B
DETAILS OF COMPUTATIONAL COMPLEXITIES (21), (29), (30)

We present here the main steps for computing the computational complexity of the sampling and weighting steps of the proposed VBMPF in nonlinear system (17) with $\mathbf{Q}_n$ diagonal; (32), (33), (36), (37), (39) and (40) can be obtained similarly.

A. Sampling step: $\mathcal{C}^{\mathsf{S}}_{\mathsf{VBMPF}}$ (21)

1) For $j=1,2,\cdots,K$, draw $\{\mathbf{x}_{n-1}^{j,(s_\ell)}\}_{\ell=1}^{J}\sim\sum_{s=1}^{S}\frac{1}{S}\delta(\mathbf{x}_{n-1}^{j}-\mathbf{x}_{n-1}^{j,(s)})$: The cost of sampling J particles from a discrete pmf is equivalent to $\mathcal{C}_{\mathsf{R}}(J)$; Repeating this K times requires a total sampling cost of $K\mathcal{C}_{\mathsf{R}}(J)$.
2) For $k=1,2,\cdots,K$, for $s=1,2,\cdots,S$, for $\ell=1,2,\cdots,J$, evaluate $\mathbf{f}_{n-1}^{k}(\mathcal{X}_{k,n-1}^{(s_\ell)})$: The total cost of this step is approximately $SJ\mathcal{C}_{\mathbf{f}}$ flops.
3) For $k=1,2,\cdots,K$, for $s=1,2,\cdots,S$, compute $\hat{\mu}_n^k(\mathbf{x}_{n-1}^{k,(s)})=\frac{1}{J}\sum_{\ell=1}^{J}\mathbf{f}_{n-1}^{k}(\mathcal{X}_{k,n-1}^{(s_\ell)})$: The vectors, $\mathbf{f}_{n-1}^{k}(\mathcal{X}_{k,n-1}^{(s_\ell)})$, with size $(n_{\mathbf{x}^k}\times 1)$, were already computed in 2). Evaluating their average for a given k and s requires $Jn_{\mathbf{x}^k}$ flops. The total cost for all k and s is therefore $SJn_{\mathbf{x}}$ flops.
4) The evaluation of the square-root, Σ_n, of the (diagonal) matrix, $\mathbf{Q}_{n-1}$, costs $n_{\mathbf{x}}$ flops.
5) For $k=1,2,\cdots,K$, for $s=1,2,\cdots,S$, draw $\tilde{\mathbf{x}}_n^{k,(s)}\sim\mathcal{N}\left(\hat{\boldsymbol{\mu}}_n^k(\mathbf{x}_{n-1}^{k,(s)}),\mathbf{Q}_{n-1}^k\right)$: For a given k and s, sampling $\mathbf{r}^{k,(s)}\sim\mathcal{N}(\mathbf{0},\mathbb{I}_{n_{\mathbf{x}^k}})$ is equivalent to $n_{\mathbf{x}^k}$ flops. Computing $\mathbf{p}^{k,(s)}=\Sigma_n^k\mathbf{r}^{k,(s)}$ and $\tilde{\mathbf{x}}_n^{k,(s)}=\hat{\mu}_n^k(\mathbf{x}_{n-1}^{k,(s)})+\mathbf{p}^{k,(s)}$, with Σ_n^k the k^{th} $(n_{\mathbf{x}^k}\times n_{\mathbf{x}^k})$ block of the (diagonal) matrix Σ_n, require $2n_{\mathbf{x}^k}$ flops. Accordingly, the total cost of these operations of all k and s is $3Sn_{\mathbf{x}}$ flops.

By summing the costs of the five steps above and using $K=\frac{n_{\mathbf{x}}}{n_{\mathbf{x}^k}}$, one eventually obtains (21).

B. Weighting step: $\mathcal{C}^{\mathsf{W}}_{\mathsf{VBMPF}}$ (29) in the case of $\mathbf{R}_n$ diagonal

1) For $s=1,2,\cdots,S$, compute $\tilde{w}_n^{(s)}=\exp\left(-\frac{1}{2}||\mathbf{y}_n-\mathbf{h}_n(\tilde{\mathbf{x}}_n^{(s)})||^2_{\mathbf{R}_n^{-1}}\right)$: For a given s, once $\mathbf{h}_n(\tilde{\mathbf{x}}_n^{(s)})$ is evaluated ($\mathcal{C}_{\mathbf{h}}$ flops), the operations $\mathbf{e}_n^{(s)}=\mathbf{y}_n-\mathbf{h}_n(\tilde{\mathbf{x}}_n^{(s)})$, $\boldsymbol{\alpha}_n^{(s)}=\mathbf{R}_n^{-1}\mathbf{e}_n^{(s)}$ and $\tilde{w}_n^{(s)}=$

$\exp\left(-\frac{1}{2}(\mathbf{e}_n^{(s)})^T\boldsymbol{\alpha}_n^{(s)}\right)$ are then performed with $n_{\mathbf{y}}$, $n_{\mathbf{y}}$ and $2n_{\mathbf{y}}+1$ flops, respectively. Repeating these for all s leads to a total cost of $(\mathcal{C}_{\mathbf{h}}+4n_{\mathbf{y}}+1)S$ flops.

2) For $s=1,2,\cdots,S$, normalize the weights as $w_n^{(s)}=\frac{\tilde{w}_n^{(s)}}{\sum_{s'=1}^{S}\tilde{w}_n^{(s')}}$: The total cost for all s is $2S-1$ flops.

3) For $k=1,2,\cdots,K$, for $s=1,2,\cdots,S$, for $s'=1,2,\cdots,S$, compute $\epsilon_{k,n}^{(s,s')}=w_n^{(s')}||\mathbf{y}_n-\mathbf{h}_n(\mathcal{Z}_{k,n}^{(s,s')})||^2_{\mathbf{R}_n^{-1}}$: Similarly to 1), the cost of computing $\epsilon_{k,n}^{(s,s')}$ for a given k, s and s' is $\mathcal{C}_{\mathbf{h}}+4n_{\mathbf{y}}$ flops; the total cost for all k, s and s' is therefore $(\mathcal{C}_{\mathbf{h}}+4n_{\mathbf{y}})KS^2$ flops. Then, one should subtract the cost of computing $||\mathbf{y}_n-\mathbf{h}_n(\mathcal{Z}_{k,n}^{(s,s')})||^2_{\mathbf{R}_n^{-1}}$ for all k and s with $s'=s$, which is $(\mathcal{C}_{\mathbf{h}}+4n_{\mathbf{y}}-1)KS$ flops, since these terms were already computed in 1) (note that $\mathcal{Z}_{k,n}^{(s,s)}=\tilde{\mathbf{x}}_n^{(s)}$). One eventually obtains $(\mathcal{C}_{\mathbf{h}}+4n_{\mathbf{y}})KS^2-(\mathcal{C}_{\mathbf{h}}+4n_{\mathbf{y}}-1)KS$ flops.

4) For $k=1,2,\cdots,K$, for $s=1,2,\cdots,S$, compute $\tilde{\lambda}_n^{k,(s)}=\exp\left(-\frac{1}{2}\sum_{s'=1}^{S}\epsilon_{k,n}^{(s,s')}\right)$: The total cost for all k and s is $KS(S+1)$ flops.

5) For $k=1,2,\cdots,K$, for $s=1,2,\cdots,S$, normalize the weights as $\lambda_n^{k,(s)}=\frac{\tilde{\lambda}_n^{k,(s)}}{\sum_{s'=1}^{S}\tilde{\lambda}_n^{k,(s')}}$: Similarly to 2) this requires $(2S-1)K$ flops.

By summing the costs of the five steps above and using $K=\frac{n_{\mathbf{x}}}{n_{\mathbf{x}^k}}$, one eventually obtains (29).

C. Weighting step: $\mathcal{C}^{\mathsf{W}}_{\mathsf{VBMPF}}$ (30) in the case of $\mathbf{R}_n$ non-diagonal

1) Performing the Cholesky decomposition of $\mathbf{R}_n$ requires approximately $\frac{n_{\mathbf{y}}^3}{3}+n_{\mathbf{y}}^2+\frac{2}{3}n_{\mathbf{y}}$ flops to compute an upper-triangular matrix, $\mathbf{U}_n$. Computing $\mathbf{V}_n=\mathbf{U}_n^{-1}$ requires approximately $\frac{n_{\mathbf{y}}^3}{3}+\frac{n_{\mathbf{y}}^2}{2}+\frac{n_{\mathbf{y}}}{6}$ flops.

2) For $s=1,2,\cdots,S$, compute $\tilde{w}_n^{(s)}=\exp\left(-\frac{1}{2}||\mathbf{y}_n-\mathbf{h}_n(\tilde{\mathbf{x}}_n^{(s)})||^2_{\mathbf{R}_n^{-1}}\right)$: For a given s, once $\mathbf{h}_n(\tilde{\mathbf{x}}_n^{(s)})$ is evaluated ($\mathcal{C}_{\mathbf{h}}$ flops), the operations $\mathbf{e}_n^{(s)}=\mathbf{y}_n-\mathbf{h}_n(\tilde{\mathbf{x}}_n^{(s)})$, $\boldsymbol{\alpha}_n^{(s)}=\mathbf{V}_n^T\mathbf{e}_n^{(s)}$ and $\tilde{w}_n^{(s)}=\exp\left(-\frac{1}{2}(\boldsymbol{\alpha}_n^{(s)})^T\boldsymbol{\alpha}_n^{(s)}\right)$ are then performed with approximately $n_{\mathbf{y}}$, $n_{\mathbf{y}}^2-\frac{n_{\mathbf{y}}}{2}$ and $2n_{\mathbf{y}}+1$ flops, respectively. Repeating these for all s leads to a total cost of $(\mathcal{C}_{\mathbf{h}}+n_{\mathbf{y}}^2+\frac{5}{2}n_{\mathbf{y}}+1)S$ flops.

3) The normalization of the weights, $\tilde{w}_n^{(s)}$, for all s, requires $2S-1$ flops.

4) For $k=1,2,\cdots,K$, for $s=1,2,\cdots,S$, for $s'=1,2,\cdots,S$, compute $\epsilon_{k,n}^{(s,s')}=w_n^{(s')}||\mathbf{y}_n-\mathbf{h}_n(\mathcal{Z}_{k,n}^{(s,s')})||^2_{\mathbf{R}_n^{-1}}$: As in 2), the cost of computing $\epsilon_{k,n}^{(s,s')}$ for a given k, s and s' is $\mathcal{C}_{\mathbf{h}}+n_{\mathbf{y}}^2+\frac{5}{2}n_{\mathbf{y}}$ flops; the total cost for all k, s and s' is therefore $(\mathcal{C}_{\mathbf{h}}+n_{\mathbf{y}}^2+\frac{5}{2}n_{\mathbf{y}})KS^2$ flops. Similarly to 3) in Appendix B-B, it remains to subtract the computing cost of $||\mathbf{y}_n-\mathbf{h}_n(\mathcal{Z}_{k,n}^{(s,s')})||^2_{\mathbf{R}_n^{-1}}$ for all k and s with $s'=s$, which is $(\mathcal{C}_{\mathbf{h}}+n_{\mathbf{y}}^2+\frac{5}{2}n_{\mathbf{y}}-1)KS$ flops, as already computed in 2). One eventually obtains $(\mathcal{C}_{\mathbf{h}}+n_{\mathbf{y}}^2+\frac{5}{2}n_{\mathbf{y}})KS^2-(\mathcal{C}_{\mathbf{h}}+n_{\mathbf{y}}^2+\frac{5}{2}n_{\mathbf{y}}-1)KS$ flops.

5) The last remaining steps are the computation of the weights, $\tilde{\lambda}_n^{k,(s)}$, for all k and s, then normalizing

them as in 4) and 5) of Appendix B-B in which one can see that these require $(S^2 + 3S - 1)K$ flops.

Based on the approximated costs of the five steps above and using $K = \frac{n_{\mathbf{x}}}{n_{\mathbf{x}^k}}$, one eventually obtains (30).

ACKNOWLEDGMENT

Research reported in this publication was supported by King Abdullah University of Science and Technology (KAUST). We would like to thank three anonymous reviewers for their constructive comments and suggestions.

REFERENCES

[1] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1518–69, June 2002.

[2] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. Springer-Verlag, 2005.

[3] B. Delyon, "Remarks on linear and nonlinear filtering," *IEEE Transactions on Information Theory*, vol. 41, no. 1, pp. 317–22, January 1995.

[4] H. Künsch, "State space and hidden markov models," in *Complex Stochastic Systems*, O. E. Barndorff-Nielsen, D. R. Cox, and C. Klüppelberg, Eds. CRC Press, 2001, ch. 3, pp. 109–173.

[5] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, ser. Statistics for Engineering and Information Science. New York: Springer Verlag, 2001.

[6] H. Tanizaki, *Nonlinear Filters, Estimation and Applications*, 2nd ed. Berlin: Springer, 1996.

[7] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice Hall, Englewood Cliffs, New Jersey, 1979.

[8] J. S. Liu and R. Chen, "Sequential monte carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–44, September 1998.

[9] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear / non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[10] P. Djuric, J. Kotecha, A. J. Zhang, A. Y. T. Ghirmai, and M. B. nand J.Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[11] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/ non-Gaussian Bayesian state estimation," *IEE Proceedings - F*, vol. 140, pp. 107–113, 1993.

[12] F. Desbouvries, Y. Petetin, and B. Ait-El-Fquih, "Direct, Prediction- and Smoothing-based Kalman and Particle Filter Algorithms," *Signal Processing*, vol. 91, no. 8, pp. 2064–2077, 2011.

[13] L. Tiancheng, M. Bolic, and P. Djuric, "Resampling Methods for Particle Filtering: Classification, implementation, and strategies," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 70–86, 2015.

[14] P. Djuric and M. Bugallo, "Particle Filtering for High-Dimensional Systems," in *Proceedings of the IEEE International Workshop on CAMSAP*, 2013.

[15] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles To High-Dimensional Particle Filtering," *Monthly Weather Review*, vol. 136, no. 12, pp. 4629–40, 2008.

[16] T. Bengtsson, P.Bickel, , and B. Li, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," *Probability and Statistics: Essays in Honor of David A. Freedman*, vol. 2, pp. 316–34, 2008.

[17] F. Daum and J. Huang, "Curse of dimensionality and particle filters," in *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, 2003.

[18] P. Djuric and M. Bugallo, "Multiple particle filtering with improved efficiency and performance," in *Proceedings of the IEEE International Conference on ICASSP*, Brisbane, Australia, 2015.

[19] M. Bugallo and P. Djuric, "Particle Filtering in High-Dimensional Systems with Gaussian Approximations," in *Proceedings of the IEEE International Conference on ICASSP*, Florence, Italy, 2014.

[20] ——, "Gaussian particle filtering in high-dimensional systems," in *Proceedings of the IEEE International Workshop on SSP*, Jupiters, Gold Coast, Australia, 2014.

[21] P. Closas and M. Bugallo, "Improving Accuracy by Iterated Multiple Particle Filtering," *IEEE Signal Processing Letters*, vol. 19, no. 8, pp. 531–34, 2012.

[22] P. Djuric, T. Lu, and M. Bugallo, "Multiple Particle Filtering," in *Proceedings of the IEEE International Conference on ICASSP*, New Orleans, USA, 2007.

[23] F. Septier and G. W. Peters, "Langevin and Hamiltonian Based Sequential MCMC for Efficient Bayesian Filtering in High-Dimensional Spaces," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 312–27, 2016.

[24] ——, "An Overview of Recent Advances in Monte-Carlo Methods for Bayesian Filtering in High-Dimensional Spaces," in *Theoretical Aspects of Spatial-Temporal Modeling*, G. W. Peters and T. Matsui, Eds. SpringerBriefs - JSS Research Series in Statistics, November 2015.

[25] T. Jaakkola, "Tutorial on variational approximation methods," in *Advanced Mean Field Methods - Theory and Practice*, M. Opper and D. Saad, Eds. MIT Press, 2001, pp. 129–59.

[26] V. Smidl and A. Quinn, *The Variational Bayes Method in Signal Processing*. Springer, 2006.

[27] ——, "Variational Bayesian Filtering," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5020–5030, 2008.

[28] S. Sarkka and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 596–600, March 2009.

[29] M. A. Chappell, A. R. Groves, B. Whitcher, and M. W. Woolrich, "Variational Bayesian Inference for a Nonlinear Forward Model," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 223–36, January 2009.

[30] B. Ait-El-Fquih and T. Rodet, "Variational Bayesian Kalman Filtering in Dynamical Tomography," in *Proceedings of the IEEE International Conference on ICASSP*, Prague, Czech Republic, 2011.

[31] J. Sun, J. Zhou, and X. Gu, "Variational Bayesian Two-Stage Kalman Fliter for Systems with Unknown Inputs," in *Proceedings of the International Workshop on IWIEE*, Harbin, China, 2012.

[32] J. Ting, E. Theodorou, and S. Schaal, "A Kalman filter for robust outlier detection," in *Proceedings of IEEE Interantional Conference on IROS*, San Diego, USA, 2007.

[33] B. Ait-El-Fquih and C. Gouy-Pailler, "Backward hidden Markov chain for outlier-robust filtering and fixed-interval smoothing," in *Proceedings of the IEEE International Conference on ICASSP*, Vancouver, Canada, 2013.

[34] M. Mansouri, H. Nounou, and M. Nounou, "State Estimation of a Chemical Reactor Process Model - A Comparative Study," in *Proceedings of the IEEE International Multi-Conference on SSD*, Hammamet, Tunisia, 2013.

[35] Z. Chen, S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Variational Bayesian Methods For Multimedia Problems," *IEEE Transactions on Multimedia*, vol. 16, no. 4, pp. 1000–17, June 2014.

[36] I. Mbalawata, S. Sarkka, M. Vihola, and H. Haario, "Adaptive Metropolis Algorithm Using Variational Bayesian Adaptive Kalman Filter," *Computational Statistics & Data Analysis*, vol. 83, pp. 101—15, 2015.

[37] Y. Zheng, A. Fraysse, and T. Rodet, "Efficient Variational Bayesian Approximation Method Based on Subspace Optimization," *IEEE Transactions on Image Processing*, vol. 24, no. 2, pp. 681–93, 2015.

[38] B. Ait-El-Fquih and I. Hoteit, "Fast Kalman-like Filtering in large-dimensional linear and Gaussian state-space models," *IEEE Transactions on Signal Processing*, vol. 63, no. 21, pp. 5853–67, 2015.

[39] C. Robert, *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. New York: Springer Science & Business Media, 2007.

[40] D. Rubin, "Using the SIR algorithm to simulate posterior distributions," in *Bayesian Statistics*, J. Bernardo, M. DeGroot, D. Lindley, and A. Smith, Eds. Oxford University Press, 1988, vol. 3, pp. 395–402.

[41] B. Ait-El-Fquih and I. Hoteit, "An efficient multiple particle filter based on the variational Bayesian approach," in *Proceedings of the IEEE International Symposium on ISSPIT*, Abu Dhabi, UAE, 2015.

[42] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, pp. 183–233, 1999.

[43] T. Jaakkola and M. Jordan, "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, pp. 25–37, 2000.

[44] M. Bolic, P. M. Djuric, and S. Hong, "Resampling Algorithms for Particle Filters: A Computational Complexity Perspective," *EURASIP Journal on Applied Signal Processing*, vol. 15, pp. 2267–77, 2004.

[45] L. N. Trefethen and D. Bau, *Numerical linear algebra*. Philadelphia: Society for Industrial and Applied Mathematics, 1997.

Dr. Boujemaa Ait-El-Fquih received the Ph.D. degree in signal processing from Télécom SudParis (formerly Institut national des télécommunications), Evry, France in 2007. He worked at the Institut Français du Pétrole (IFP), Lyon, as a postdoctoral fellow (2007), the University of Paris XI, Orsay, as a teaching and research assistant (2008), the University of Bordeaux I, jointly with Electricité de France (EDF), Chatou, as a research associate (2010), and the Commissariat à l'Energie Atomique (CEA), Saclay, as a research engineer (2011). Dr. Ait-El-Fquih joined the King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, in 2013, in which he is currently working as a research scientist. His research interests are in Bayesian estimation methods and their applications in geophysics and signal processing.

Dr. Ibrahim Hoteit is an associate professor in the Earth Sciences and Engineering program and the Applied Mathematics and Computational Sciences program at KAUST. Prior to joining KAUST in 2009, he was a research scientist at Scripps Institution of Oceanography in University of California. Dr. Hoteit's research interests are in modeling and simulation of environmental fluid systems on supercomputers, with specific interest in data assimilation and inverse problems and associated uncertainty quantification for large-scale systems. Dr. Hoteit co-authored more than 90 scientific papers and is a co-recipient of five best conference paper awards. He is a member of the American and European Geophysical Unions, the Society of Industrial and Applied Mathematics, and an elected member of the UNESCO Center of Pure and Applied Mathematics. Dr. Hoteit earned his B.S. (1997), M.S. (1998), and Ph.D. (2002) from the University of Joseph Fourier, Grenoble, France.