

Cooperative parallel particle filters for online model selection and applications to urban mobility



Luca Martino^{a,*}, Jesse Read^{b,c}, Víctor Elvira^d, Francisco Louzada^a

^a Inst. of Math. Sciences and Computing, Universidade de São Paulo, Brazil

^b Dep. of Information and Computer Science, Aalto University, Finland

^c LTCI, CNRS, Télécom ParisTech, Université Paris-Saclay, France

^d Dep. of Signal Theory and Communic., Universidad Carlos III de Madrid, Spain

ARTICLE INFO

Article history:

Available online 28 September 2016

Keywords:

Sequential model selection
Modality detection
Marginal likelihood estimation
Parallel particle filters
Distributed inference
Urban mobility

ABSTRACT

We design a sequential Monte Carlo scheme for the dual purpose of Bayesian inference and model selection. We consider the application context of urban mobility, where several modalities of transport and different measurement devices can be employed. Therefore, we address the joint problem of online tracking and detection of the current modality. For this purpose, we use interacting parallel particle filters, each one addressing a different model. They cooperate for providing a global estimator of the variable of interest and, at the same time, an approximation of the posterior density of each model given the data. The interaction occurs by a parsimonious distribution of the computational effort, with online adaptation for the number of particles of each filter according to the posterior probability of the corresponding model. The resulting scheme is simple and flexible. We have tested the novel technique in different numerical experiments with artificial and real data, which confirm the robustness of the proposed scheme.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Monte Carlo (MC) algorithms are very popular numerical techniques for the approximation of optimal a posteriori estimators [9, 18, 13, 30]. Particle filters (PFs) are well-known MC methods that have been extensively applied in different fields, in order to handle analytically intractable posterior probability density functions (pdfs) [4, 10, 15, 20]. In this work, we consider the problem of tracking a variable of interest within a state-space model, where the dynamic and observation equations are unknown. A finite set of candidate models is considered, and the true model may be included (or not) within this set. Thus, the goal is both sequential tracking and online model selection [14, 24]. This problem is particularly of interest in an urban mobility context where different modalities of transport can be employed (e.g., bus, train, cycling). In the following, we discuss the main contributions of this work, the range of their applicability, and also related work.

Contributions and organization of the paper. In this work, we present a simple approach involving parallel particle filters (PFs), called *model averaging parallel particle filters* (MAPF), based on the

Bayesian model averaging (BMA) principle [21]. The proposed PF solution performs the inference by running as many filters as candidate models, i.e., each PF is tailored to a different states-space model (Section 4). The parallel PFs in MAPF cooperate to provide a global approximation of the posterior distribution of the variable of interest given the data and, at the same time, also provide particle approximations of the posterior distributions of the models given the data. The interaction among the filters occurs by a dynamic allocation of the computational effort, i.e., distributing a portion of the total number of particles to each filter, proportionally to the posterior pdf of the corresponding model. Namely, the parallel filters in MAPF exchange information adapting online the number of particles employed in each filter. However, the total number of particles remains fixed, chosen in advance by the user. Hence, the novel scheme is able to distribute the computational effort online, rewarding the filters which are considering the most probable models given the observed data.

An exhaustive theoretical derivation of MAPF is provided in Sections 2–3. In Section 2, we introduce the general Bayesian formulation for tackling our problem. The posterior distribution which we study is *doubly* intractable, in the sense that it cannot be evaluated analytically and the computation of the related moments is analytically intractable. Then, the importance sampling (IS) technique for approximating the corresponding optimal Bayesian estimators

* Corresponding author.

E-mail address: lukatotal@gmail.com (L. Martino).

is described in Section 3. The sequential IS approach is presented in Section 3.1. The use of parallel particle filters naturally appears following the theoretical derivation of MAPF. We have focused special attention in describing the sequential approximation of the marginal likelihood (a.k.a., Bayesian evidence), in Sections 3.1, 3.2 and further material is also given in different appendices. Different possible formulations are derived and discussed in detail, thus providing a concise review which could be of value to interested practitioners and researchers. Interesting special cases of MAPF are discussed in Section 4.2. For instance, when all the candidate models share the same likelihood function, MAPF can be seen as a unique PF with adaptive prior density. Moreover, when all the candidate models are equal and coincide with the true model, then MAPF can be interpreted as a distributed PF scheme where the filters compete in order to obtain more particles (according to the performance at each specific run). The application of MAPF in the case of time-varying models is described in Section 5.

We test MAPF in different experimental scenarios and apply the proposed scheme in a real data problem: an urban mobility context for detecting different modes of activity. For instance, one possible goal in urban mobility is to detect if a traveler is walking or riding a bus (or switching between these two modalities). The numerical results in Section 6, with both synthetic and real data, show the efficiency and flexibility of the proposed algorithms.¹

Range of applicability. The range of applicability of MAPF is wide, clearly not restricted only to the urban mobility case. Applications of sequential model selection problem are very common in different fields where a stream of data is observed [11,14,23,24]. For instance, in financial analysis where price fluctuations act differently under different regimes, as the ongoing instability in times of crisis [34]. They have also been used for fraud detection [22], explicitly modeling the switch to a regime of fraudulent activity from ordinary activity. In the medical domain, patients may need to be modeled for possible complications [37]. In marine tracking [38], vessels must be suitably monitored depending on different conditions of the environment. Another general application, where MAPF can be also applied, is the change detection and system identification problem [3]. Furthermore, MAPF can be used for optimizing the tuning of one or several parameters (problem specifically tackled in [2,35]), selecting them within a set of possible candidates (e.g., see Section 6.2).

Related works. The problem addressed in this work is related to the *switching model* [11,36,31,7] and the *multiple model* approaches [1,5,11]. The first main difference is that, in general, this kind of methods requires the definition and tuning of a transition probability matrix among the models (formed by, at least, K^2 parameters). MAPF is a simpler scheme since the transition matrix is not required so that the design effort (in terms of construction and tuning) can be focused completely on a proper choice of the dynamic and likelihood functions.² In a time-varying model scenario, MAPF only requires the tuning of an additional scalar parameter (denoted as T_V in the rest of the work). We also remark that, in a standard tracking and model selection setup (when the true model is unknown and is not varying with the time), this additional scalar parameter is even not needed. Another main difference is that MAPF does not allow exchange of particles among the filters (with the exception of the refreshing steps; see Section 5), whereas in the switching model approach this exchange is usually allowed.

This difference yields that MAPF can provide more accurate estimates (taking also advantage of the parallelization), since each filter employs its own particles without hybrid mixes which can jeopardize the tracking. Moreover, MAPF automatically adapts the number of particles of each PF keeping fixed the overall computational cost. PF schemes with adaptive number of particles have been proposed in literature [19,17], but in the context of a unique filter. Moreover unlike in these works, in MAPF, the overall computational cost is kept constant, chosen in advance by the user. Other related and well-known approaches for the joint purpose of sequential tracking and parameter estimation are given in [2,35]. In [14], the Kolmogorov–Smirnov test is applied for testing a dynamic equation in a state-space model. Finally, it is important to mention that specific particle filters addressing models with unknown statistics have been also designed [15,32].

2. Bayesian inference and model selection in state-space models

Let us denote the unknown state $\mathbf{x}_t \in \mathcal{X}$ with $\mathcal{X} \in \mathbb{R}^{d_x}$ (continuous space) or $\mathcal{X} \in \mathbb{N}^{d_x}$ (discrete space), $t \in \mathbb{N}$, and the current observed data as $\mathbf{y}_t \in \mathcal{Y} \in \mathbb{R}^{d_y}$. We assume that the hidden sequence $\mathbf{x}_{1:T} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$ is generated with a transition pdf $g(\mathbf{x}_t|\mathbf{x}_{t-1})$. At the t -th iteration, we observe \mathbf{y}_t with probability $f(\mathbf{y}_t|\mathbf{x}_t)$, so that after T iterations we have $\mathbf{y}_{1:T} = [\mathbf{y}_1, \dots, \mathbf{y}_T]$. The previous two pdfs, g_t and f_t , jointly compose the true model indicated as \mathcal{T} , which we consider unknown. Namely, setting $g_1(\mathbf{x}_1|\mathbf{x}_0) = g_1(\mathbf{x}_1)$, we have

$$\mathcal{T}: \begin{cases} g_t(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ f_t(\mathbf{y}_t|\mathbf{x}_t) \end{cases}, \quad t = 1, \dots, T. \quad (1)$$

For the sake of simplicity, we consider the model \mathcal{T} fixed over the time t , however the case of time-varying model \mathcal{T}_t is also tackled in Section 5.

We are interested in inferring the hidden states $\mathbf{x}_{1:T}$ given all the observed measurements $\mathbf{y}_{1:T}$. Both, $\mathbf{x}_{1:T}$ and $\mathbf{y}_{1:T}$, are generated according to the model \mathcal{T} in Eq. (1). Since \mathcal{T} is unknown, we consider K different possible models, denoted as \mathcal{M}_k , with $k \in \{1, \dots, K\}$, formed by a transition pdf and a likelihood function, i.e., setting $q_{k,1}(\mathbf{x}_1|\mathbf{x}_0) = q_{k,1}(\mathbf{x}_1)$,

$$\mathcal{M}_k: \begin{cases} q_{k,t}(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ \ell_{k,t}(\mathbf{y}_t|\mathbf{x}_t) \end{cases}, \quad t = 1, \dots, T. \quad (2)$$

We denote the set of all considered models as $\mathcal{F} = \{\mathcal{M}_1, \dots, \mathcal{M}_K\}$. The true model \mathcal{T} could be contained in \mathcal{F} , i.e., $\mathcal{T} \in \mathcal{F}$, but in general we have $\mathcal{T} \notin \mathcal{F}$. However, even in the case $\mathcal{T} \notin \mathcal{F}$, we apply the Bayesian model averaging (BMA) approach [21] which provides a coherent mechanism for taking in account the model uncertainty, improving the overall filtering performance.

We assume a prior probability mass function (pmf), $p(\mathcal{M}_k)$, $k = 1, \dots, K < \infty$ over the different models. Thus, the goal is to make inference about the sequence $\mathbf{x}_{1:T}$ and the K different possible models, given the set of received measurements $\mathbf{y}_{1:T}$. Therefore, we study the following posterior density

$$p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) = \sum_{k=1}^K p(\mathbf{x}_{1:T}, \mathcal{M}_k|\mathbf{y}_{1:T}), \quad (3)$$

$$= \sum_{k=1}^K p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \mathcal{M}_k)p(\mathcal{M}_k|\mathbf{y}_{1:T}), \quad (4)$$

$$= \frac{1}{p(\mathbf{y}_{1:T})} \sum_{k=1}^K p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \mathcal{M}_k)p(\mathbf{y}_{1:T}|\mathcal{M}_k)p(\mathcal{M}_k), \quad (5)$$

¹ A preliminary MATLAB code of MAPF is provided at <https://www.mathworks.com/matlabcentral/fileexchange/58597-model-averaging-particle-filter>.

² We remark that, in MAPF, every model can differ from the other ones for both dynamic and measurement equations.

where $p(\mathbf{y}_{1:T}) = \sum_j^K p(\mathbf{y}_{1:T}|\mathcal{M}_j)p(\mathcal{M}_j)$. In general, the study of the posterior pdf above is (doubly) analytically intractable because of the following reasons. On the one hand, we cannot evaluate $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ in Eq. (5) completely, since we cannot evaluate

$$p(\mathcal{M}_k|\mathbf{y}_{1:T}) = \frac{Z_{k,T} p(\mathcal{M}_k)}{\sum_{j=1}^K Z_{j,T} p(\mathcal{M}_j)}, \quad (6)$$

owing to, in general, we are not able to compute

$$Z_{k,T} = p(\mathbf{y}_{1:T}|\mathcal{M}_k) = \int_{\mathcal{X}^T} p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}|\mathcal{M}_k) d\mathbf{x}_{1:T}, \quad (7)$$

for all $k = 1, \dots, K$. However, given an index $k \in \{1, \dots, K\}$ and $\mathbf{y}_{1:T}$, we are able to evaluate³

$$p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}|\mathcal{M}_k) = \left[q_{k,1}(\mathbf{x}_1) \prod_{t=2}^T q_{k,t}(\mathbf{x}_t|\mathbf{x}_{t-1}) \right] \left[\prod_{t=1}^T \ell_{k,t}(\mathbf{y}_t|\mathbf{x}_t) \right], \quad (8)$$

so that we can also evaluate

$$p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \mathcal{M}_k) = \frac{p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}|\mathcal{M}_k)}{Z_{k,T}} \propto p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}|\mathcal{M}_k), \quad (9)$$

up to a normalizing constant. On the other hand, we often cannot compute analytically integrals involving the function $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$. For instance, one can be interested the computation of the Minimum Mean Square Error (MMSE) estimator of the sequence of hidden states $\mathbf{x}_{1:T}$,

$$\mathbf{I}_{1:t} = E[\mathbf{x}_{1:t}] = \int_{\mathcal{X}^T} \mathbf{x}_{1:t} p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) d\mathbf{x}_{1:T}. \quad (10)$$

More generally, the computation of moments of $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ are not analytically intractable, and its approximation is computationally demanding. In the next section, we employ Monte Carlo schemes for approximating both $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \mathcal{M}_k)$ and $p(\mathcal{M}_k|\mathbf{y}_{1:T})$, for $k = 1, \dots, K$. As a consequence, we also approximate the complete posterior pdf $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ in Eq. (5).

3. Monte Carlo approximation via importance sampling

In order to approximate efficiently the posterior pdfs in Eqs. (3), (6) and (9) (solving the computational challenges described in the previous section), we consider the use of Monte Carlo techniques. First, we describe a *batch importance sampling* (IS) approach where, for each model $k \in \{1, \dots, K\}$, we consider to draw M possible sequences

$$\mathbf{x}_{k,1:T}^{(m)} = [\mathbf{x}_{k,1}^{(m)}, \mathbf{x}_{k,2}^{(m)}, \dots, \mathbf{x}_{k,T}^{(m)}] \sim \varphi_k(\mathbf{x}_{1:T}),$$

from a proposal pdf $\varphi_k: \mathcal{X}^T \rightarrow \mathbb{R}$, with $m = 1, \dots, M$. Namely, in the batch approach, we sample directly in the whole space \mathcal{X}^T . In the next section, we introduce the corresponding sequential scheme (working sequentially in the subspace \mathcal{X}). The batch IS technique is described as follows. For each index $k \in \{1, \dots, K\}$, draw M samples $\mathbf{x}_{k,1:T}^{(1)}, \dots, \mathbf{x}_{k,1:T}^{(M)}$ from a proposal pdf $\varphi_k(\mathbf{x}_{1:T})$, where $\varphi_k: \mathcal{X}^T \rightarrow \mathbb{R}$, and assign to each sample the following importance weights

$$w_{k,T}^{(m)} = \frac{p(\mathbf{x}_{k,1:T}^{(m)}, \mathbf{y}_{1:T}|\mathcal{M}_k)}{\varphi_k(\mathbf{x}_{k,1:T}^{(m)})}, \quad (11)$$

with $m = 1, \dots, M$, and $k = 1, \dots, K$. Note that the total number of samples are $N = KM$. We can approximate

$$Z_{k,T} = p(\mathbf{y}_{1:T}|\mathcal{M}_k) = \int_{\mathcal{X}^T} p(\mathbf{x}_{k,1:T}, \mathbf{y}_{1:T}|\mathcal{M}_k) d\mathbf{x}_{k,1:T},$$

using basic IS arguments [33,25] as⁴

$$\hat{Z}_{k,T} = \frac{1}{M} \sum_{m=1}^M w_{k,T}^{(m)} \approx p(\mathbf{y}_{1:T}|\mathcal{M}_k). \quad (12)$$

We can also write $p(\mathbf{y}_{1:T}) \approx \sum_{k=1}^K \hat{Z}_{k,T} p(\mathcal{M}_k)$. Furthermore, we can approximate the measure of $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \mathcal{M}_k) = \frac{p(\mathbf{x}_{k,1:T}, \mathbf{y}_{1:T}|\mathcal{M}_k)}{Z_{k,T}}$ by the following particle approximation

$$\hat{p}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \mathcal{M}_k) = \sum_{m=1}^M \bar{w}_{k,T}^{(m)} \delta(\mathbf{x}_{1:T} - \mathbf{x}_{k,1:T}^{(m)}), \quad (13)$$

where

$$\bar{w}_{k,T}^{(m)} = \frac{w_{k,T}^{(m)}}{\sum_{j=1}^M w_{k,T}^{(j)}} = \frac{w_{k,T}^{(m)}}{M \hat{Z}_{k,T}}, \quad (14)$$

is the normalized weight of the m -th sample of the k -th model (normalized considering all the samples associated to the k -th model). Thus, given Eq. (5), we also obtain an approximation of $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$,

$$\begin{aligned} \hat{p}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) &= \frac{1}{\hat{p}(\mathbf{y}_{1:T})} \sum_{k=1}^K \hat{p}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \mathcal{M}_k) \hat{p}(\mathbf{y}_{1:T}|\mathcal{M}_k) p(\mathcal{M}_k) \\ &= \frac{1}{\sum_{j=1}^K \hat{Z}_{j,T} p(\mathcal{M}_j)} \\ &\quad \times \sum_{k=1}^K \left[\left(\frac{1}{M \hat{Z}_{k,T}} \sum_{m=1}^M w_{k,T}^{(m)} \delta(\mathbf{x}_{1:T} - \mathbf{x}_{k,1:T}^{(m)}) \right) \hat{Z}_{k,T} p(\mathcal{M}_k) \right] \\ &= \sum_{k=1}^K \sum_{m=1}^M \left(\frac{w_{k,T}^{(m)} p(\mathcal{M}_k)}{\sum_{j=1}^K \sum_{i=1}^M w_{j,T}^{(i)} p(\mathcal{M}_j)} \right) \delta(\mathbf{x}_{1:T} - \mathbf{x}_{k,1:T}^{(m)}). \end{aligned}$$

The last expression can be summarized as

$$\hat{p}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) = \sum_{k=1}^K \sum_{m=1}^M \bar{\gamma}_{k,T}^{(m)} \delta(\mathbf{x}_{1:T} - \mathbf{x}_{k,1:T}^{(m)}), \quad (15)$$

where we have denoted

$$\bar{\gamma}_{k,T}^{(m)} = \frac{w_{k,T}^{(m)} p(\mathcal{M}_k)}{\sum_{j=1}^K \sum_{i=1}^M w_{j,T}^{(i)} p(\mathcal{M}_j)}, \quad (16)$$

$$= \frac{1}{M} \frac{w_{k,T}^{(m)} p(\mathcal{M}_k)}{\sum_{j=1}^K \hat{Z}_{j,T} p(\mathcal{M}_j)}. \quad (17)$$

Since $\hat{Z}_{k,T} = \frac{1}{M} \sum_{j=1}^M w_{k,T}^{(j)}$, it is of a particular interest from a theoretical point of view that the weight $\bar{\gamma}_{k,T}^{(m)}$ can be decomposed as⁵

³ For the sake of simplicity, in the following mathematical elaborations, we consider both densities $q_{k,t}$ and $\ell_{k,t}$ to be normalized w.r.t. \mathbf{x} and \mathbf{y} respectively, i.e., $\int_{\mathcal{X}} q_{k,t}(\mathbf{x}|\mathbf{z}) d\mathbf{x} = 1$ and $\int_{\mathcal{Y}} \ell_{k,t}(\mathbf{y}|\mathbf{z}) d\mathbf{y} = 1$. As a consequence, the joint pdf $p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}|\mathcal{M}_k)$ in Eq. (8) is also normalized.

⁴ We consider that $\varphi_k(\mathbf{x}_{1:T})$ is normalized.

⁵ Note also that $\sum_{k=1}^K \sum_{m=1}^M \bar{\gamma}_{k,T}^{(m)} = 1$.

$$\tilde{\gamma}_{k,T}^{(m)} = \frac{w_{k,T}^{(m)}}{\sum_{j=1}^M w_{k,T}^{(j)}} \frac{\hat{Z}_{k,T} p(\mathcal{M}_k)}{\sum_{j=1}^K \hat{Z}_{j,T} p(\mathcal{M}_j)} = \tilde{w}_{k,T}^{(m)} \tilde{\rho}_{k,T}, \quad (18)$$

where $\tilde{w}_{k,T}^{(m)}$ is given in Eq. (14), and $\tilde{\rho}_{k,T}$ is an estimator of the posterior of the k -th model $p(\mathcal{M}_k | \mathbf{y}_{1:T})$ in Eq. (6),

$$\tilde{\rho}_{k,T} = \frac{\hat{Z}_{k,T} p(\mathcal{M}_k)}{\sum_{j=1}^K \hat{Z}_{j,T} p(\mathcal{M}_j)} \approx p(\mathcal{M}_k | \mathbf{y}_{1:T}). \quad (19)$$

It is important to remark that the use of K parallel IS schemes seems to appear naturally from the factorization $\tilde{\gamma}_{k,T}^{(m)} = \tilde{w}_{k,T}^{(m)} \tilde{\rho}_{k,T}$. Indeed, we can also rewrite the approximation in Eq. (15) as the convex combination

$$\hat{p}(\mathbf{x}_{1:T} | \mathbf{y}_{1:T}) = \sum_{k=1}^K \tilde{\rho}_{k,T} \hat{p}(\mathbf{x}_{1:T} | \mathbf{y}_{1:T}, \mathcal{M}_k), \quad (20)$$

where $\tilde{\rho}_{k,T}$ is the normalized weight of the k -th model. Finally, for instance, the computation of the MMSE estimator $\hat{\mathbf{x}}_{1:T}$ in Eq. (10) is approximated as

$$\hat{\mathbf{I}}_{1:T} \approx \hat{\mathbf{I}}_{1:T} = \sum_{k=1}^K \sum_{m=1}^M \tilde{\gamma}_{k,T}^{(m)} \mathbf{x}_{k,1:T}^{(m)}, \quad (21)$$

or with the equivalent two-stage formula

$$\begin{cases} \tilde{\mathbf{I}}_{k,1:T} = \sum_{m=1}^M \tilde{w}_{k,T}^{(m)} \mathbf{x}_{k,1:T}^{(m)}, \\ \hat{\mathbf{I}}_{1:T} = \sum_{k=1}^K \tilde{\rho}_{k,T} \tilde{\mathbf{I}}_{k,1:T}, \end{cases} \quad (22)$$

where $\tilde{\mathbf{I}}_{k,1:T}$ represents the approximated MMSE estimator considering only the k -th model. The IS procedure above can easily reformulated within a sequential framework, as we describe in the next section.

3.1. Sequential importance sampling (SIS) and marginal likelihood estimation

The IS method can be alternatively performed in a sequential manner, i.e., providing an approximation of the posterior pdf at each iteration t using the previous approximation at the iteration $t-1$. Let us consider an index $k \in \{1, \dots, K\}$. Observing the following recursive relationship between the posterior pdfs at $t-1$ and t [13],

$$p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}, \mathcal{M}_k) = \frac{\ell_{k,t}(\mathbf{y}_t | \mathbf{x}_t) q_{k,t}(\mathbf{x}_t | \mathbf{x}_{t-1})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \mathcal{M}_k)} p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}, \mathcal{M}_k) \quad (23)$$

(see Appendix A for further details), we can build the empirical approximation $\hat{p}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}, \mathcal{M}_k)$ as

$$\hat{p}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}, \mathcal{M}_k) = \frac{1}{M \hat{Z}_{k,t}} \sum_{i=1}^M w_{k,t}^{(i)} \delta(\mathbf{x}_{1:t} - \mathbf{x}_{k,1:t}^{(i)}), \quad (24)$$

given the previous $\hat{p}(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}, \mathcal{M}_k)$. Recall that, we also obtain an estimator of $p(\mathbf{y}_{1:t} | \mathcal{M}_k)$, as $\hat{Z}_{k,t} = \frac{1}{M} \sum_{m=1}^M w_{k,t}^{(m)}$. Let us consider a proposal density $\phi_k : \mathcal{X}^T \rightarrow \mathbb{R}$ factorizes as

$$\phi_k(\mathbf{x}_{k,1:T}) = \phi_{k,1}(\mathbf{x}_{k,1}) \prod_{t=2}^T \phi_{k,t}(\mathbf{x}_{k,t} | \mathbf{x}_{k,t-1}), \quad (25)$$

with $\phi_{k,t} : \mathcal{X} \rightarrow \mathbb{R}$ for $t = 1, \dots, T$. Given Eq. (23), we can infer a recursive relationship between two importance weights at consecutive iterations [13],

$$\begin{aligned} w_{k,t}^{(m)} &= w_{k,t-1}^{(m)} \frac{\ell_{k,t}(\mathbf{y}_t | \mathbf{x}_{k,t}^{(m)}) q_{k,t}(\mathbf{x}_{k,t}^{(m)} | \mathbf{x}_{k,t-1}^{(m)})}{\phi_{k,t}(\mathbf{x}_{k,t}^{(m)} | \mathbf{x}_{k,1:t-1}^{(m)})} \\ &= w_{k,t-1}^{(m)} \lambda_{k,t}^{(m)} = \prod_{\tau=1}^t \lambda_{k,\tau}^{(m)}, \end{aligned} \quad (26)$$

where

$$\lambda_{k,t}^{(m)} = \frac{\ell_{k,t}(\mathbf{y}_t | \mathbf{x}_{k,t}^{(m)}) q_{k,t}(\mathbf{x}_{k,t}^{(m)} | \mathbf{x}_{k,t-1}^{(m)})}{\phi_{k,t}(\mathbf{x}_{k,t}^{(m)} | \mathbf{x}_{k,1:t-1}^{(m)})},$$

and $\mathbf{x}_{k,t}^{(m)} \sim \phi_{k,t}(\mathbf{x}_{k,t}^{(m)} | \mathbf{x}_{k,1:t-1}^{(m)})$, and $m = 1, \dots, M$. Therefore, given the recursive expression of $w_{k,t}^{(m)}$, we can rewrite the estimator $\hat{Z}_{k,t}$ as

$$\hat{Z}_{k,t} = \frac{1}{M} \sum_{m=1}^M w_{k,t}^{(m)} = \frac{1}{M} \sum_{m=1}^M \left[\prod_{\tau=1}^t \lambda_{k,\tau}^{(m)} \right]. \quad (27)$$

However, the estimator of $p(\mathbf{y}_{1:t} | \mathcal{M}_k)$ above has not a unique formulation. Indeed, it is possible to obtain an approximation of the denominator in Eq. (23) (see Appendix B),

$$\hat{p}(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \mathcal{M}_k) = \sum_{i=1}^M \tilde{w}_{k,t-1}^{(i)} \lambda_{k,t}^{(i)},$$

where $\tilde{w}_{k,t-1}^{(i)} = \frac{w_{k,t-1}^{(i)}}{\sum_{n=1}^M w_{k,t-1}^{(n)}}$ are the normalized weights at $t-1$ -th iteration. As a consequence, since $p(\mathbf{y}_{1:t} | \mathcal{M}_k) = \prod_{\tau=1}^t p(\mathbf{y}_\tau | \mathbf{y}_{1:\tau-1}, \mathcal{M}_k)$, we have a second possible formulation of the estimator of $p(\mathbf{y}_{1:t} | \mathcal{M}_k)$,

$$\tilde{Z}_{k,t} = \prod_{\tau=1}^t \left[\sum_{j=1}^M \tilde{w}_{k,\tau-1}^{(j)} \lambda_{k,\tau}^{(j)} \right] = \prod_{\tau=1}^t \left[\frac{\sum_{j=1}^M w_{k,\tau}^{(j)}}{\sum_{j=1}^M w_{k,\tau-1}^{(j)}} \right]. \quad (28)$$

In Appendix B, we show a complete derivation of the estimator $\tilde{Z}_{k,t}$ and that $\tilde{Z}_{k,t} \equiv \hat{Z}_{k,t}$, as one could easily realize from Eq. (28).

Remark 1. Note that, in SIS, there are two possible equivalent formulations of the estimator of $p(\mathbf{y}_{1:t} | \mathcal{M}_k)$, i.e., $\hat{Z}_{k,t}$ in Eqs. (27), $\tilde{Z}_{k,t}$ in Eq. (28), and they are equivalent, $\tilde{Z}_{k,t} \equiv \hat{Z}_{k,t}$.

In any case, since $\hat{p}(\mathbf{y}_{1:t}) = \sum_{j=1}^M Z_{j,t} p(\mathcal{M}_j)$, the model weight $\tilde{\rho}_{k,t} = \frac{\hat{p}(\mathbf{y}_{1:t}, \mathcal{M}_k)}{\hat{p}(\mathbf{y}_{1:t})}$ employed in Eqs. (19)–(20) can be expressed as

$$\tilde{\rho}_{k,t} = \frac{\hat{Z}_{k,t} p(\mathcal{M}_k)}{\sum_{j=1}^M \hat{Z}_{j,t} p(\mathcal{M}_j)} \approx p(\mathcal{M}_k | \mathbf{y}_{1:t}). \quad (29)$$

The weights $\tilde{\rho}_{k,t}$ above are then used for computing the global estimator at the t -th iteration, similarly in Eq. (22).

3.2. Sequential importance resampling (SIR) and marginal likelihood estimation

In several algorithms, such as the *sequential Monte Carlo* (SMC) methods, resampling steps are performed within SIS schemes for avoiding the degeneracy of the weights [13,15]. Let us denote as $\bar{\mathbf{x}}_{k,1:t}^{(j)} \in \{\mathbf{x}_{k,1:t}^{(1)}, \dots, \mathbf{x}_{k,1:t}^{(M)}\}$, a resampled particle at the iteration t (resampled according to the normalized weights $\tilde{w}_{k,t-1}^{(j)}$, $j = 1, \dots, M$, at the t -th iteration). The unnormalized importance weights of the resampled particles, denoted as $\alpha_{k,t}^{(j)}$, $j = 1, \dots, M$, are set to the same value [13,15,27], i.e.,⁶

⁶ This is a proper choice, but it is not unique; see *The concept of weighted samples* in [25, Chapter 2] or [33, Section 14.2].

$$\alpha_{k,t}^{(1)} = \alpha_{k,t}^{(2)} = \dots = \alpha_{k,t}^{(M)}.$$

A proper value for the (unnormalized) importance weight $\alpha_{1:t-1}^{(j)}$ associated with the j -th resampled particle [27] is

$$\alpha_{k,t}^{(j)} = \hat{Z}_{k,t} = \frac{1}{N} \sum_{i=1}^N w_{k,t}^{(i)}, \quad \forall j = 1, \dots, M. \quad (30)$$

One reason why this seems a suitable choice, for instance, is that defining the following weights

$$\xi_{k,t}^{(m)} = \begin{cases} w_{k,t}^{(m)}, & \text{without resampling at } t\text{-th iteration,} \\ \alpha_{k,t}^{(m)}, & \text{with resampling at } t\text{-th iteration,} \end{cases} \quad (31)$$

then, in any case, $\frac{1}{N} \sum_{n=1}^N \xi_{k,t}^{(j)} = \hat{Z}_{k,t}$, as expected. In general, the resampling steps are not applied at each iteration, but only when some statistical criterion is satisfied [8,13,15] (e.g., see Section 4). The recursive expression of the weights for SIR becomes

$$\xi_{k,t}^{(m)} = \xi_{k,t-1}^{(m)} \lambda_{k,t}^{(m)}, \quad (32)$$

where

$$\xi_{k,t-1}^{(m)} = \begin{cases} \xi_{k,t-1}^{(m)}, & \text{without res. at } (t-1)\text{-th iter.,} \\ \hat{Z}_{k,t-1}, & \text{with res. at } (t-1)\text{-th iter.} \end{cases} \quad (33)$$

Remark 2. With the recursive definition of the weights $\xi_{k,t}^{(m)}$'s in Eqs. (32)–(33), the two possible estimators of the marginal likelihood $p(\mathbf{y}_{1:t}|\mathcal{M}_k)$ are the same within SIR, as well. The estimators are

$$\hat{Z}_{k,t} = \frac{1}{M} \sum_{m=1}^M \xi_{k,t-1}^{(m)} \lambda_{k,t}^{(m)}, \quad \tilde{Z}_{k,t} = \prod_{\tau=1}^t \left[\sum_{j=1}^M \xi_{k,\tau-1}^{(j)} \lambda_{k,\tau}^{(j)} \right] \quad (34)$$

where $\tilde{\xi}_{k,t-1}^{(j)} = \frac{\xi_{k,t-1}^{(j)}}{\sum_{m=1}^M \xi_{k,t-1}^{(m)}}$. They are equivalent and valid estimators [27] as shown in Appendix B.

So far, in this section, we have considered a specific value of the index $k \in \{1, \dots, K\}$. However, for our purpose, we need of all the model weights $\bar{\rho}_{k,t}$, for all values $k = 1, \dots, K$. Thus, a practical implementation employing K parallel particle filters appears natural. In the next section, we describe the proposed scheme in details.

4. Model averaging particle filters

The factorization of the global normalized weights $\tilde{\gamma}_{k,t}^{(m)} = \bar{w}_{k,t}^{(m)} \bar{\rho}_{k,t}$ suggests the use of K parallel IS schemes. Indeed, each IS scheme can compute independently $\bar{w}_{k,t}^{(m)}$ and $\hat{Z}_{k,t}$, and then they merge all the information for calculating $\bar{\rho}_{k,t}$ (see Fig. 1). Therefore, we consider K parallel particle filters (as in [8,12,13], for instance) using the transition model as proposal pdf⁷

$$\phi_{k,t}(\mathbf{x}_{k,t}|\mathbf{x}_{k,1:t-1}) = q_{k,t}(\mathbf{x}_{k,t}|\mathbf{x}_{k,t-1}),$$

each one tracking a different states-space model \mathcal{M}_k , for $k = 1, \dots, K$. The K parallel PFs cooperate for providing a unique global approximation of the complete posterior, as described in the previous sections. As a consequence, an approximation $\bar{\rho}_{k,T}$ of the model posterior densities $p(\mathcal{M}_k|\mathbf{y}_{1:T})$ is also provided.

⁷ More sophisticated PF techniques could be also employed, each one addressing a different model.

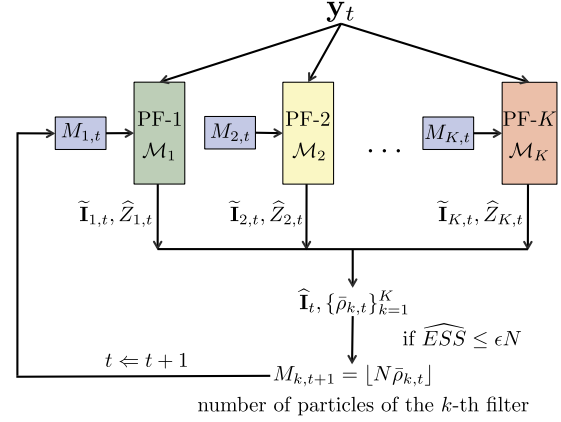


Fig. 1. Graphical overview of the MAPF scheme in Table 2 for approximating the MMSE estimator $\hat{\mathbf{I}}_t$ and the posterior of each model given the observed data. Note that both \widehat{ESS} formulas in Eq. (37) take into account the (normalized) weights $\tilde{\gamma}_{k,t}^{(i_k)} = \bar{w}_{k,t}^{(i_k)} \bar{\rho}_{k,t}$, for all $i_k = 1, \dots, M_{k,t}$ and k .

Table 1

Main notation for MAPF.

K	Number of parallel PFs (and different models).
T	Total number of iterations of each PF.
N	Total a number of particles distributed among the PFs.
$M_{k,t}$	Number of particles of the k -th PF at the iteration t .
T_V	Refreshing parameter.
$\tilde{Z}_{k,t}, \hat{Z}_{k,t}$	Estimator of $p(\mathbf{y}_{1:t} \mathcal{M}_k)$ (two equivalent formulations).
$w_{k,t}^{(m)}$	Importance weight assigned to the sample $\mathbf{x}_{k,t}^{(m)}$.
$\bar{w}_{k,t}^{(m)}$	Weight assigned to the sample $\mathbf{x}_{k,t}^{(m)}$, normalized w.r.t. the $M_{k,t}$ weights of k -th filter.
$\bar{\rho}_{k,t}$	Approximation of $p(\mathcal{M}_k \mathbf{y}_{1:t})$.
$\rho_{k,t}$	Approximation of $\bar{p}(\mathcal{M}_k \mathbf{y}_{1:t})$, considering an updated prior $\bar{p}(\mathcal{M}_k)$.
$\tilde{\gamma}_{k,t}^{(m)}$	Global normalized weight ($\tilde{\gamma}_{k,t}^{(m)} = \bar{w}_{k,t}^{(m)} \bar{\rho}_{k,t}$ or $\tilde{\gamma}_{k,t}^{(m)} = \bar{w}_{k,t}^{(m)} \rho_{k,t}$).
$\mathbf{I}_{1:t} = E[\mathbf{x}_{1:t}]$	MMSE estimator of the hidden state $\mathbf{x}_{1:t}$.
$\hat{\mathbf{I}}_{1:t}$	Partial approximation of $\mathbf{I}_{1:t}$ of the k -th PF.
$\tilde{\mathbf{I}}_{1:t}$	Global approximation of \mathbf{I}_t .

Table 1 summarizes the main notation used in the *Model Averaging Particle Filter* (MAPF) and Table 2 provides an exhaustive description of MAPF. The computational effort is distributed adaptively among the parallel PFs, proportionally to the (approximated) posterior pdf of the model \mathcal{M}_k given the current set of data $\mathbf{y}_{1:t}$, i.e., $\bar{\rho}_{k,t} \approx p(\mathcal{M}_k|\mathbf{y}_{1:t})$. More specifically, when a resampling step is performed, the number of particles of each filter is adapted online, taking into account an approximation of the probability $p(\mathcal{M}_k|\mathbf{y}_{1:t})$. However, the overall computational cost remains invariant (pre-established by the user in advance) since total number of particles denote as N does not vary, i.e.,

$$\sum_{j=1}^K M_{j,t} = M_{1,t} + M_{2,t} + \dots + M_{K,t} = N, \quad \text{for } t = 1, \dots, T. \quad (35)$$

Since $M_{k,t+1} = \lfloor \bar{\rho}_{k,t} \rfloor$, when a resampling step is applied (or $M_{k,t+1} = M_{k,t}$, without) in general we have $N' = N - \sum_{j=1}^K M_{j,t+1} \geq 0$. When $N' > 0$, the remaining N' particles can be assigned to the filters in different ways: for instance, with probability $\bar{\rho}_{k,t}$, i.e., set $M_{k,t+1} = M_{k,t+1} + N'$ so that $\sum_{j=1}^K M_{j,t+1} = N$. Alternatively, the use of a minimum number of particles for each filter can be considered (avoiding, in this way, the stop of some filter with no

particles assigned).⁸ The MMSE estimator $\hat{\mathbf{x}}_t$ in Eq. (10) can be approximated using Eq. (21) or (22), i.e.,

$$\hat{\mathbf{I}}_t = \sum_{k=1}^K \tilde{\rho}_{k,t} \tilde{\mathbf{I}}_{k,t} \quad \text{where} \quad \tilde{\mathbf{I}}_{k,t} = \sum_{i_k=1}^{M_{k,t}} \tilde{w}_{k,t}^{(i_k)} \mathbf{x}_{k,t}^{(i_k)}, \quad (36)$$

where $\tilde{\mathbf{I}}_{k,t}$ is provided by each filter independently, as shown in Fig. 1. We have denoted as $\tilde{\mathbf{x}}_{k,1:t}^{(i_k)} \in \{\mathbf{x}_{k,1:t}^{(1)}, \dots, \mathbf{x}_{k,1:t}^{(M_{k,t})}\}$ the samples after a resampling step. A resampling step is applied at each iteration that the Effective Sample Size (ESS) is smaller than a threshold value (ϵN where $|\epsilon| \leq 1$). We adapt some approximations of ESS proposed in literature [13,20,28,26] to the MAPF context, hence for instance the possible conditions are

$$\widehat{ESS}(\bar{\gamma}) = \frac{1}{\sum_{k=1}^K \sum_{m=1}^{M_{k,t}} (\bar{\gamma}_{k,t}^{(m)})^2} \leq \epsilon N, \quad \text{or} \quad (37)$$

$$\widehat{ESS}(\bar{\gamma}) = \frac{1}{\max_{m,k} \bar{\gamma}_{k,t}^{(m)}} \leq \epsilon N$$

where $\bar{\gamma}_{k,t}^{(m)}$ are the global weights in Eq. (16). Namely, when $\widehat{ESS} \leq \epsilon N$, a resampling step is performed.

Remark 3. Note that ESS approximations in Eq. (37), $\widehat{ESS}(\bar{\gamma})$, involve the use of the global weights, $\bar{\gamma}_{k,t}^{(m)}$, so that the decision of applying resampling in a specific iteration is also a cooperative task carried out by the parallel filters.

Fig. 2 depicts another graphical representation of the MAPF algorithm in Table 2 with $K = 2$ filters, when a resampling step is employed. The filters interact through the adaptation of numbers of particles of each filter, $M_{k,t}$ (and also providing jointly the global MMSE estimator). However, Fig. 2 shows that the resampling steps are performed independently by each filter (in the figure $K = 2$), i.e., no particles are exchanged among the filters. Further considerations about MAPF are provided in Appendix C.

Remark 4. The step 4 of MAPF in Table 2 can be interpreted as a double resampling: a first resampling considering the model weights, $\tilde{\rho}_{k,t}$, adapting the number of particles and the second one considering the normalized weights, $\tilde{w}_{k,t}^{(m)}$ within a filter. However, in this scheme the exchange of particles among the filters is not allowed. In Section 5 another resampling procedure is considered, using directly the global weights $\bar{\gamma}_{k,t}^{(m)}$, where the exchange of particles is possible.

The remark above is strictly connected to the considerations in Appendix C.

4.1. Factorization of the global weights in MAPF: empirical Bayes approach

Recalling the definition in Eq. (16),

$$\bar{\gamma}_{k,t}^{(m)} = \frac{w_{k,t}^{(m)} p(\mathcal{M}_k)}{\sum_{j=1}^K \sum_{i=1}^{M_{j,t}} w_{j,t}^{(i)} p(\mathcal{M}_j)},$$

note that in this case we have (due to the use of different number of particles in the parallel filters)

Table 2

Model Averaging Particle Filters (MAPF).

– **Initialization:** Set $M_{k,1} = \frac{N}{K} \geq 2$, $\xi_{k,0}^{(i_k)} = 1$, and choose the initial states $\tilde{\mathbf{x}}_{k,0}^{(i_k)}$, for all $i_k = 1, \dots, M_{k,1}$ and $k = 1, \dots, K$.

– **For** $t = 1, \dots, T$:

1. **Propagation:** Draw $\mathbf{x}_{k,t}^{(i_k)} \sim q_{k,t}(\mathbf{x} | \tilde{\mathbf{x}}_{k,t-1}^{(i_k)})$, for $i_k = 1, \dots, M_{k,t}$ and $k = 1, \dots, K$.

2. **Particle Weighting:** Compute the weights and normalized them,

$$w_{k,t}^{(i_k)} = \xi_{k,t-1}^{(i_k)} \ell_{k,t}(\mathbf{y}_t | \mathbf{x}_{k,t}^{(i_k)}), \quad \text{and} \quad \tilde{w}_{k,t}^{(i_k)} = \frac{w_{k,t}^{(i_k)}}{\sum_{j=1}^{M_{k,t}} w_{k,t}^{(j)}}, \quad (38)$$

for $i_k = 1, \dots, M_{k,t}$, $k = 1, \dots, K$ and $\xi_{k,t-1}^{(i_k)}$ is defined in Eqs. (32)–(33).

3. **Model Weighting:** Compute, for $k = 1, \dots, K$,

$$\hat{Z}_{k,t} = \frac{1}{M_{k,t}} \sum_{j=1}^{M_{k,t}} w_{k,t}^{(j)}, \quad \text{and} \quad \tilde{\rho}_{k,t} = \frac{\hat{Z}_{k,t} p(\mathcal{M}_k)}{\sum_{j=1}^K \hat{Z}_{j,t} p(\mathcal{M}_j)}. \quad (39)$$

Alternatively, the estimator $\hat{Z}_{k,t}$ in Eq. (34) can be used.

4. **Adaptation and Resampling:** For each filter, $k = 1, \dots, K$:

– If the condition (37) is fulfilled,

(a) Set $M_{k,t+1} = \lfloor N \tilde{\rho}_{k,t} \rfloor$, and distribute the remaining $N' = N - \sum_{j=1}^K M_{j,t+1}$ particles among the K filters according to some pre-established criterion (see Section 4).

(b) Draw $M_{k,t+1}$ samples, $\tilde{\mathbf{x}}_{k,1:t}^{(1)}, \dots, \tilde{\mathbf{x}}_{k,1:t}^{(M_{k,t+1})}$, from

$$\hat{p}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}, \mathcal{M}_k) = \sum_{i_k=1}^{M_{k,t}} \tilde{w}_{k,t}^{(i_k)} \delta(\mathbf{x}_{1:t} - \mathbf{x}_{k,1:t}^{(i_k)}).$$

– Otherwise, set $M_{k,t+1} = M_{k,t}$ and $\tilde{\mathbf{x}}_{k,1:t} = \mathbf{x}_{k,1:t}^{(i_k)}$, for all i_k .

5. **Output:** Return $\{\mathbf{x}_{k,1:t}^{(i_k)}, w_{k,t}^{(i_k)}, \tilde{\rho}_{k,t}, \tilde{\gamma}_{k,t}^{(i_k)}\}$, $i_k = 1, \dots, M_{k,t}$ and $k = 1, \dots, K$.

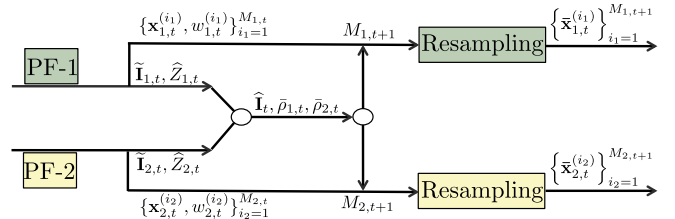


Fig. 2. Another graphical representation of the MAPF scheme, with $K = 2$ parallel particle filters. A fusion center returns, $\hat{\mathbf{I}}_t$, $\tilde{\rho}_{k,t}$, and the numbers of particles $M_{k,t+1}$, $k = 1, 2$, of the next iterations. Then, the resampling steps are performed separately.

$$\begin{aligned} \bar{\gamma}_{k,t}^{(m)} &= \frac{w_{k,t}^{(m)} p(\mathcal{M}_k)}{\sum_{j=1}^K M_{j,t} \hat{Z}_{j,t} p(\mathcal{M}_j)} \\ &= \frac{w_{k,t}^{(m)}}{\sum_{j=1}^{M_{k,t}} w_{k,t}^{(j)}} \frac{M_{k,t} \hat{Z}_{k,t} p(\mathcal{M}_k)}{\sum_{j=1}^K M_{j,t} \hat{Z}_{j,t} p(\mathcal{M}_j)} = \tilde{w}_{k,t}^{(m)} \tilde{\rho}_{k,t}, \end{aligned} \quad (40)$$

where

$$\tilde{\rho}_{k,t} = \frac{\hat{Z}_{k,t} M_{k,t} p(\mathcal{M}_k)}{\sum_{j=1}^K \hat{Z}_{j,t} M_{j,t} p(\mathcal{M}_j)} = \frac{\hat{Z}_{k,t} \tilde{p}(\mathcal{M}_k)}{\sum_{j=1}^K \hat{Z}_{j,t} \tilde{p}(\mathcal{M}_j)}, \quad (41)$$

and $\tilde{p}(\mathcal{M}_j) = M_{j,t} p(\mathcal{M}_j)$ plays the role of an updated prior pdf as in an empirical Bayes approach [6].

4.2. Sub-cases of interest

It is interesting to remark that, if the likelihood function is common to all the filters, i.e., $\ell_{k,t}(\mathbf{y}_t | \mathbf{x}_t) = \ell_t(\mathbf{y}_t | \mathbf{x}_t)$ for all k , then MAPF can be seen as a PF with adaptive prior pdf. Thus, the complete

⁸ Observe that for the algorithm in Table 2, it is necessary that $M_{k,t} \geq 2$, $\forall k, t$. In the equivalent MAPF formulation given in Appendix C, this constraint is not required.

likelihood is $p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}) = \prod_{\tau=1}^t \ell_{\tau}(\mathbf{y}_{\tau}|\mathbf{x}_{\tau})$, and the complete prior pdf of k -th model, is⁹

$$p(\mathbf{x}_{1:t}|\mathcal{M}_k) = q_{k,1}(\mathbf{x}_1) \prod_{\tau=2}^t q_{k,\tau}(\mathbf{x}_{\tau}|\mathbf{x}_{\tau-1}). \quad (42)$$

Namely, in this setup, the model selection problem becomes as the problem of a suitable choice of a prior pdf for our model. This specific case is particularly interesting from a practical point of view, as we discuss below. Since in this case $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}, \mathcal{M}_k) = p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t})p(\mathbf{x}_{1:t}|\mathcal{M}_k)$, the complete posterior $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$ in Eq. (4) can be rewritten as

$$p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}) \left[\sum_{k=1}^K \zeta_k p(\mathbf{x}_{1:t}|\mathcal{M}_k) \right], \quad (43)$$

where we have denoted $\zeta_k = p(\mathcal{M}_k|\mathbf{y}_{1:t})$. Noting that $\sum_{k=1}^K \zeta_k = 1$, we can interpret this framework as using a unique model with the dynamic-prior pdf defined by the following mixture, $p(\mathbf{x}_{1:t}) = \sum_{k=1}^K \zeta_k p(\mathbf{x}_{1:t}|\mathcal{M}_k)$. In this case, when the resampling is applied, the MAPF approach can also be interpreted as a *unique* particle filter with an *adaptive* prior pdf approximating the prior mixture $\sum_{k=1}^K \zeta_k p(\mathbf{x}_{1:t}|\mathcal{M}_k)$, using the approximated weights $\tilde{\rho}_{k,t} \approx \zeta_k$. Furthermore, if all the models are the same equal to the true one \mathcal{T} , i.e., $\mathcal{M}_1 = \mathcal{M}_2 = \dots = \mathcal{M}_K = \mathcal{T}$, then MAPF described a distributed PF scheme [8,12,15] which K parallel PFs cooperate for providing a global estimator, $\hat{\mathbf{I}}_{1:t} = \sum_{k=1}^K \tilde{\rho}_{k,t} \tilde{\mathbf{I}}_{k,1:t}$. The computational effort is distributed in order to foster the filters that are providing the best performance, in the specific run.

5. MAPF for time-varying models

The MAPF scheme can be easily modified for applying it in a time-varying model setting, where at some unknown iterations $t_1^* \leq t_2^* \leq t_3^* \dots$ the true model \mathcal{T} generating \mathbf{x} 's and \mathbf{y} 's changes, i.e., we have $\mathcal{T}_1 \rightarrow \mathcal{T}_2 \rightarrow \mathcal{T}_3 \dots$ etc. For example, a traveler may switch from walking to riding the bus, and both the dynamics and observations will vary accordingly. The change detection problem [3] can be also considered as an interesting particular case. For instance, see the numerical example in Section 6.1.

The simplest way for handling this scenario is to consider a window of $T_V \geq 1$ iterations for computing the values $\hat{\mathbf{Z}}_{k,t}$, for $k = 1, \dots, K$. In this case we modify the computation of $\hat{\mathbf{Z}}_{k,t}$: the simplest possibility is to refresh all the $\hat{\mathbf{Z}}_{k,t}$'s each T_V iterations considering only the incremental weights (as forcing $\xi_{k,t-1} = \hat{\mathbf{Z}}_{k,t-1} = 1$),¹⁰ i.e.,

$$\hat{\mathbf{Z}}_{k,t^*} = \frac{1}{M_{k,t^*}} \sum_{m=1}^{M_{k,t^*}} \lambda_{k,t^*}^{(m)}, \quad t^* = rT_V, \quad r \in \mathbb{N}. \quad (44)$$

On the one hand, with a small T_V , the algorithm is able to detect quickly the change in the model \mathcal{T} , although the approximation of the posteriors of the model given the data become more unstable (since the posterior takes into account a smaller number of observations). On the other hand, with a bigger T_V , the algorithm is more stable but the detection of model updates is slower. In this scenario, every T_V iterations, the number of particles of each filter should be refreshed, setting $M_{k,t} = \frac{N}{K}$ for all $k = 1, \dots, K$. Moreover, in order to recover lost filters, a joint resampling should be performed, considering the weights $\tilde{\gamma}_{k,t}^{(i_k)} =$

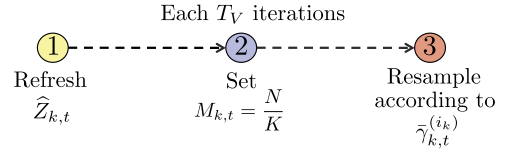


Fig. 3. Refreshing strategy for time-varying model settings.

$\tilde{w}_{k,t}^{(i_k)} \tilde{\rho}_{k,t}$, $i_k = 1, \dots, M_{k,t}$ and $k = 1, \dots, K$ (instead of considering only $\tilde{w}_{k,t}^{(i_k)}$; see Remark 4). This allows the exchange of particles among the different filters. Namely, one could draw N particles from the global approximation at the t -th iteration in Eq. (15),

$$\hat{p}(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = \sum_{k=1}^K \sum_{i_k=1}^{M_{k,t}} \tilde{\gamma}_{k,t}^{(i_k)} \delta(\mathbf{x}_{1:t} - \mathbf{x}_{k,1:t}^{(i_k)}), \quad (45)$$

where $\tilde{\gamma}_{k,t}^{(i_k)}$ are in Eqs. (16)–(18). Fig. 3 summarizes the suggested approach for handling the selection of time-varying models. The application of this refreshing strategy could be useful also in the standard model selection setting, without changes in the true model: it avoids numerical problems and can increase the robustness of the MAPF technique.

Namely, $M_{k,t} = \frac{N}{K}$ particles, for each filters, are obtained sampling $\frac{N}{K}$ times from $\hat{p}(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$ in Eq. (45). Finally, let us consider for simplicity the sub-case of common likelihood function. The improvement in the performance provided by MAPF with respect to a standard filter can be considerable in tracking applications. This is owing to, for instance, riding a bus hugely constraints both the dynamic and observation model. Indeed, simply knowing the bus routes beforehand, the movement can be restricted by a whole dimension, since the trajectory of the bus is fixed. The mathematical descriptions of different modalities of mobility can be calibrated in advance from publicly available data. In the numerical simulations in Section 6.3, for example, we use shows the real-time position of buses in the city of Helsinki (<http://live.mattersoft.fi/hsl/>).

Automatic adaptive selection of the refreshing steps. The parameter T_V can also be adaptively and automatically tuned. Indeed, an automatic approach can be used in order to decide when applying the refreshing steps. For instance, we can consider to apply the refreshing procedure, summarized in Fig. 3 (including the global resampling that involves $\tilde{\gamma}_{k,t}^{(m)}$'s), as alternative of the standard resampling steps (that employ $\tilde{w}_{k,t}^{(m)}$'s). More specifically, when one of the chosen ESS condition in (37) is fulfilled, i.e., $\widehat{ESS}(\tilde{\gamma}) \leq \epsilon N$, with probability p_r we apply the refreshing strategy (with the global resampling) otherwise, with probability $1 - p_r$, we apply a standard resampling within each filter.¹¹ Fig. 4 summarizes this procedure. Finally, note also that a mix strategy can be also employed: the adaptive procedure can be applied and the refreshing can be forced at some specific iterations (in order to increase the robustness of the method).

6. Numerical simulations

We consider three different numerical experiments in order to test the MAPF algorithm. The first two numerical examples employ artificial data whereas the last one involves real data.¹² More specifically, in the first example we study the ability of MAPF in

¹¹ Our experience suggests the choice $0.05 \leq p_r \leq 0.2$.

¹² A preliminary MATLAB code of MAPF related to the first two numerical examples is provided at <https://www.mathworks.com/matlabcentral/fileexchange/58597-model-averaging-particle-filter>.

⁹ Note that, in a Bayesian setting, the dynamic models play the role of prior pdf.

¹⁰ Alternatively a sliding window of iterations can be considered, i.e., $\hat{\mathbf{Z}}_{k,t} \equiv \tilde{\mathbf{Z}}_{k,t} = \prod_{\tau=t-T_V+1}^t \left[\sum_{j=1}^{M_{k,\tau}} \tilde{\xi}_{k,\tau-1}^{(j)} \lambda_{k,\tau}^{(j)} \right]$.

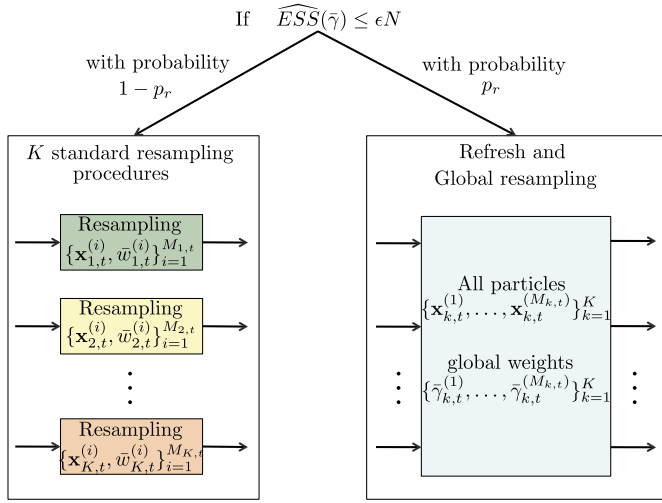


Fig. 4. Graphical representation of the automatic and adaptive procedure in order to select the refreshing steps.

a time-varying model scenario. The second example considers the parameter selection and tracking problem in a state-space model (the parameter selection can be interpreted as a sub-case of the model selection problem). The last numerical experiment applies MAPF in a urban mobility context, considering real-world data involving smartphone-based tracking.

6.1. Online model selection for time series

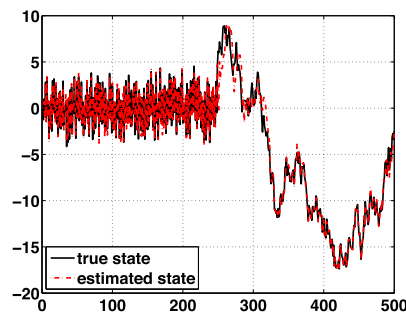
As a first example, we consider an inference problem given two possible systems of stochastic equations for modeling a time series ($x_t \in \mathbb{R}$, $t \in \mathbb{N}$). The goal is to estimate sequentially the hidden sequence $x_{1:T}$ and also to recognize, in an online fashion, which model generates the received measurements $y_{1:T}$, between

$$\mathcal{M}_1 : \begin{cases} x_{1,t} = \frac{ax_{1,t}}{1+bx_{1,t}^2} + v_{1,t}, \\ y_t = x_{1,t} + u_{1,t}, \end{cases} \quad (46)$$

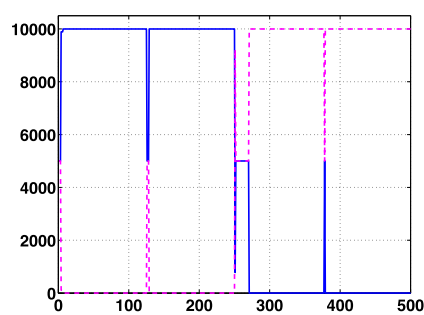
$$\mathcal{M}_2 : \begin{cases} x_{2,t} = x_{2,t-1} + v_{2,t}, \\ y_t = \exp\{-c x_{2,t}\} + u_{2,t}, \end{cases} \quad (47)$$

with, $a = -10$, $b = 3$, $c = 0.2$ and $t = 1, \dots, T = 500$. The variables $v_{k,t} \sim \mathcal{N}(0, 1)$ and $u_{k,t} \sim \mathcal{N}(0, \frac{1}{2})$, $k = 1, 2$ represent Gaussian perturbations.

We set that the first 250 observations are generated from \mathcal{M}_1 and the remaining from \mathcal{M}_2 . Namely, for $t \leq 250$ the true model is $\mathcal{T}_{t \leq 250} \equiv \mathcal{M}_1$ whereas, for $t > 250$, we have $\mathcal{T}_{t > 250} \equiv \mathcal{M}_2$. That is, we change the model at iteration $t^* = 250$.



(a) True and estimated path in one run.



(b) Num. of particles, $M_{1,t}$, $M_{2,t}$, as function of t .

Fig. 5. (a) True (solid line) and estimated (dashed) sequence of states $x_{1:T}$ in one specific run, obtained by MAPF. (b) Evolution of the numbers of particles $M_{1,t}$, $M_{2,t}$ as function of the iterations t , in one specific run. A refreshing step is applied at each $T_V = 125$ iterations.

Table 3

MSE of MAPF in estimation of $x_{1:T}$, compared with the MSE of different unique PFs using the same number of particles $N = 10^5$.

PF true model	MAPF ($T_V = 125$)	PF - \mathcal{M}_1	PF - \mathcal{M}_2	PF wrong model
6.64	6.91	95.09	106.21	115.44

We apply MAPF with $N = 10^5$ total number of particles, $\epsilon = 0.1$, considering $\widehat{ESS}(\tilde{\gamma}) = \frac{1}{\sum_{k=1}^K \sum_{m=1}^{M_{k,t}} (\tilde{\gamma}_{k,t}^{(m)})^2}$, and refreshing window $T_V = 125$ (clearly, $K = 2$). Given the previous assumptions, we have averaged the results over 10^4 independent simulations, where the hidden states and the data are generated at each run, according to the model \mathcal{T}_t . Fig. 5(a) shows the true and the estimated sequence of states in one specific run. Fig. 5(b) depicts the evolution of the number of particles of each filters, $M_{k,t}$, $k = 1, 2$, as function of the iterations t , in one particular run. Note that, the algorithm quickly detects the true model and assigns adequately all the computational effort to the filter addressing the true model, at the corresponding iteration t . Fig. 5(b) also illustrates that MAPF recovers quickly after a refreshing step, each $T_V = 125$ iterations. We have also computed the Mean Square Error (MSE) in the estimation of $x_{1:T}$, and then averaged it over the 10^4 independent runs. We compare the MSE obtained by MAPF with different unique particle filters with $N = 10^5$ particles: one PF considers the true model (best case), a second PF considers always \mathcal{M}_1 , the third PF always deals with \mathcal{M}_2 and the last one addresses always the wrong model, i.e., \mathcal{M}_2 for $t \leq 250$ and \mathcal{M}_1 for $t > 250$. The results are shown in Table 3. MAPF provides an MSE virtually identical to the MSE of the best case, obtained using a unique PF addressing always the true model.

We also test MAPF considering different values of T_V and using the adaptive procedure for selecting the refreshing steps described in Section 5, with $p_r = 0.1$. As suggested in Section 5, we use a mix strategy forcing also the refreshing procedure at the iteration 350, 410 and 450 (note that this time steps are far from the time change of the model at the iteration 250). The results are shown in Table 4. We can observe that, in any case, MAPF provides MSE values much closer to the a single PF considering a unique true model (with an MSE of 6.64) than a single PF considering only \mathcal{M}_1 or \mathcal{M}_2 (with MSEs of 95.09 and 106.21, respectively). The good results obtained with $T_V \leq 50$ show that the application of the global resampling can be beneficial improving the global estimation (note that the MSE is small even if T_V is not a divisor of 250). The results obtained with a $T_V > 250$ show the ability of MAPF of recovering the right model after a previous time change. Finally, observe that the automatic adaptive procedure is a reasonable option as robust tuning, i.e., when a specific selection by user (in advance) of T_V is problematic.

Table 4

MSE of MAPF in estimation of $x_{1:T}$ with different values of T_V and using the adaptive procedure in Section 5 with $p_r = 0.1$ ($N = 10^5$).

T_V	17	20	35	50	100	125	250	260	300	adaptive
MSE	6.95	6.80	7.12	6.84	15.78	6.91	6.88	7.08	21.68	8.03

Table 5

Percentage (averaged over the T iterations and 500 independent runs) of the perfect match between the estimated model and the true model, in different settings (S1, S2 and S3) and number of models K .

(a) Without refreshing					(b) With refreshing $T_V = 20$				
K	5	20	50	100	K	5	20	50	100
S1	98.55	98.60	98.60	98.59	S1	62.54	91.12	91.80	91.68
S2	98.48	97.53	97.49	97.26	S2	62.06	79.04	81.99	81.26
S3	98.50	98.57	98.51	98.58	S3	64.54	91.04	91.90	92.40

(c) With refreshing $T_V = 40$					(d) With refreshing $T_V = 100$				
K	5	20	50	100	K	5	20	50	100
S1	71.81	94.78	95.00	95.50	S1	78.47	97.98	98.08	98.02
S2	82.65	89.37	89.65	90.36	S2	92.46	95.83	95.68	96.15
S3	82.78	94.78	95.42	95.36	S3	94.46	97.76	98.06	97.93

6.2. Parameter selection

Consider the system of equations defining different models, with $x_{k,t} \in \mathbb{R}$,

$$\mathcal{M}_k : \begin{cases} x_{k,t} = a_k |x_{k,t-1}| + v_{k,t} \\ y_t = b_k \log(x_{k,t}^2) + u_{k,t} \end{cases}, \quad t = 1, \dots, T, \quad (48)$$

where $v_{k,t} \sim \mathcal{N}(0, \sigma_{1,k}^2)$ and $u_{k,t} \sim \mathcal{N}(0, \sigma_{2,k}^2)$. We consider that true model \mathcal{T} coincides with \mathcal{M}_K with parameters $a_K = b_K = 1$, and $\sigma_{1,k} = \sigma_{2,k} = 1$. Thus, at each run, all the data $y_{1:T}$ are generated according to \mathcal{T} . We consider 3 different experimental settings:

• **General setup (S1):** in this case, each model \mathcal{M}_k has both different dynamic equations and different likelihood functions. Specifically, we consider a grid of parameters,

$$a_k = \frac{k}{K}, \quad k = 1, \dots, K. \quad (49)$$

Note that $a_K = 1$. The first $K - 1$ values of $\sigma_{1,k}$ are chosen randomly at each run, more precisely, $\sigma_{i,k} \sim \mathcal{U}([0.1, 10])$ for $i = 1, 2$ and $k = 1, \dots, K - 1$. Moreover, we have another grid for b_k ,

$$b_k = \frac{1}{3} + \frac{10(k-1)}{K}, \quad k = 1, \dots, K - 1, \quad (50)$$

and $b_K = 1$.

• **Common likelihood function (S2):** the parameters a_k are selected as in Eq. (49), and again $\sigma_{1,k} \sim \mathcal{U}([0.1, 10])$, whereas in this case, we set $b_k = 1$, $\sigma_{2,k} = 1$ for all $k = 1, \dots, K$. In other words, the models share the same likelihood function.

• **Common dynamic model (S3):** in this case, $a_k = 1$ and $\sigma_{1,k} = 1$ for all $k = 1, \dots, K$, whereas the b_k s are set as in Eq. (50) (with $b_K = 1$) and $\sigma_{2,k} \sim \mathcal{U}([0.1, 10])$.

We apply MAPF performing 500 independent runs for every scenario, S1, S2 and S3. In each experiment, new hidden sequences and observed data are generated from the true model $\mathcal{T} \equiv \mathcal{M}_K$. In all cases, we set $T = 500$, the number of particles is $N = 10^5$, and the resampling parameter is $\epsilon = 0.1$. We also test different values of $T_V \in \{20, 40, 100, T + 1\}$ (the case $T_V = T + 1$ corresponds to a “non-refreshing” setup) and number of considered models $K \in \{5, 20, 50, 100\}$. Table 5 provides the percentage of the perfect match between the estimated model obtained by the maximum a posteriori (MAP) estimator and the true model. This percentage is

averaged over the T iterations, at each run. The results show that MAPF is able to detect the true model in different scenarios, even with very frequent refreshing steps. Figs. 6 show the variable number of particles (in one specific run) of $K = 50$ filters as function of different iterations t in different experimental setup. These figures confirm that MAPF is able to recover quickly the true model after several refreshing steps. Finally, considering $K = 5$, the numerical simulations also show that MAPF obtains a MSE approximately 10 times smaller than a unique filter with $N = 10^5$ particles employed for targeting a wrong model, without taking into account “catastrophic” runs when the filter is completely lost.

6.3. Real data: urban mobility

In this section, we conduct a study on real-world data involving smartphone-based tracking. The knowledge of the current position is useful for many applications, for example to offer real-time location-based services such as advising the traveler when it is time to change bus, or providing other kinds of services such as warnings about traffic interruptions; or, simply collecting accurate travel paths for later analysis.

Location is obtained by the device’s GPS receiver. However, this causes a relatively strong drain on the battery, so infrequent use of the GPS receiver is strongly desired. Furthermore, GPS reception does not work in some parts of the transport system, for example in the metro system (when it is underground) and on the train (due to unfavorable reception conditions caused by the metal cabin and overhead electrical lines). Hence, in this case, other kind of sensors should be incorporated to the sensor network. A volunteer group of researchers collected GPS measurements, recorded continuously during two weeks by their Android smartphones running the CONTEXTLOGGER application.¹³ To obtain a relatively accurate ground truth, we obtained readings every 10 seconds. In correspondence with our recorded data, we considered the time step as 10 seconds (so that the refreshing time T_V must take into account this consideration). We assume a linear observation model $y_t = x_t + r_t$ where x_t is the current position and r_t is a Gaussian noise with a standard deviation error of ± 10.62 meters, as recorded in the true data (accuracy is recorded on each datum). Geographical coordinates of interactions (crossings) and bus stops can be obtained from OPENSTREETMAPS,¹⁴ and, in the case of

¹³ See <http://contextlogger.org/>.

¹⁴ See <http://www.openstreetmap.org>.

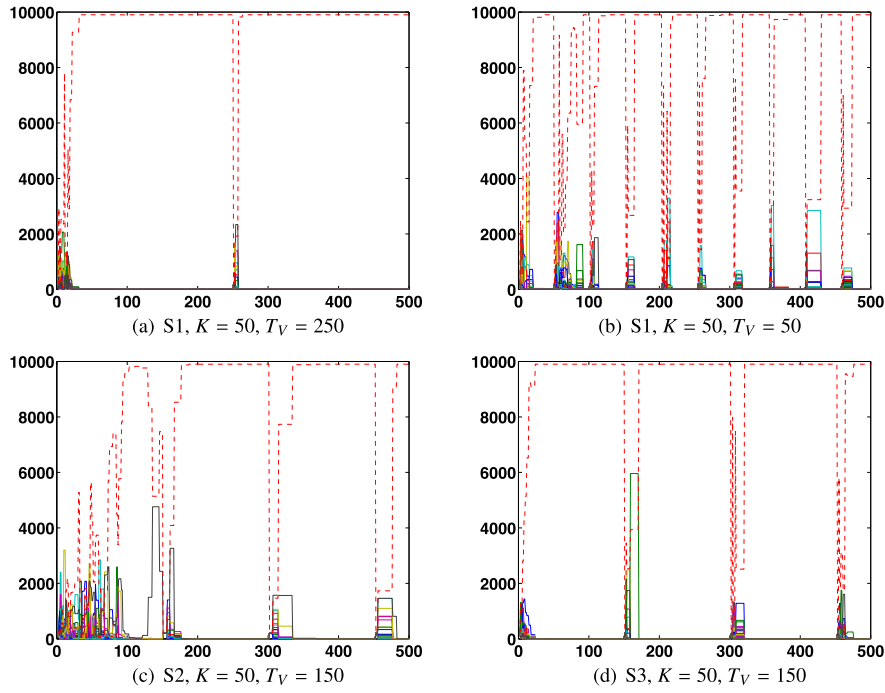


Fig. 6. Number of particles $M_{k,t}$ (with $K = 50$ filters and $N = 10^5$) as function of the iterations t , in one specific run. The number of particles of the filter corresponding to the true model is depicted with a dashed line. Figures (a) and (b) correspond to the setup S1 and $T_V = 50, 150$, respectively; figures (c) and (d) correspond to the setup S2 and S3 with $T_V = 150$, in both cases.

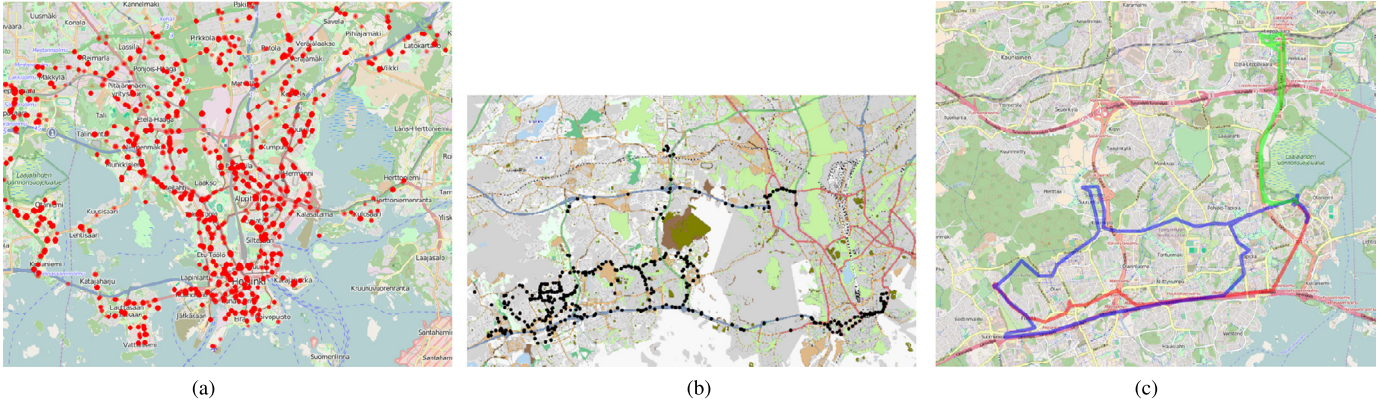


Fig. 7. (a) All bus stops in Helsinki region. (b) All intersections crossed during a 10-day period by one participant. (c) A small section of trajectories labeled under different transportation modes. Blue is car, red is bus, and green is cycling. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Helsinki which we have considered, from the urban planner.¹⁵ We plot all stops in the Helsinki region in Fig. 7. There are several possibilities for modeling the different possible modes in urban mobility problem. Here, we provide some very simple and efficient possibilities, which can be easily implemented within a commercial application:

Dynamic model for train–bus. Let us define the subset $S \subseteq \mathbb{R}^2$ as a piecewise linear approximation of the route of the corresponding train or bus. Namely, the train tracks or the bus routes can be interpreted geometrically as curves embedded in \mathbb{R}^2 . One set S is an approximation of one of these curves, obtained by a piecewise linear interpolation the corresponding route. An example is shown in Fig. 8(a). We consider the following equation,

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{h}_t(S), \quad (51)$$

where, $\mathbf{x}_t \in \mathbb{R}^2$, and $\mathbf{h}_t(S)$ is random Gaussian perturbation depending on the subset $S \subseteq \mathbb{R}^2$. In this case, $\mathbf{x}_t \sim \mathcal{N}(\mathbf{x}_{t-1}, \Sigma_t(S))$, where the covariance matrix depends on S : the slope of the eigenvector associated to the greatest eigenvalue λ_1 is the same of the current linear piece of S (see Fig. 8(a)). Moreover, denoting as λ_2 the second eigenvalue, we design the covariance matrix $\Sigma_t(S)$ in order to have $\lambda_1 \gg \lambda_2$, i.e., the generated particles are highly correlated in the direction of the route. Clearly, the different trains, trams or buses are discriminated by the different routes, i.e. the set S . Furthermore, the covariance matrix $\Sigma_t(S)$ can also contain other kinds of information, such as the velocity; indeed, it is possible to choose properly the values of the two eigenvalues λ_1, λ_2 . One possibility to improve the particle generation is to apply the rejection sampling principle [29,33] for discarded the samples outside the route S (in this case, the noise is a truncated Gaussian pdf, restricted within S). The values λ_1, λ_2 are tuned using a Least Squares pre-processing according to the data, depending go the specific vehicle (train, tram, bus) and route.

¹⁵ See <http://hsl.fi>.

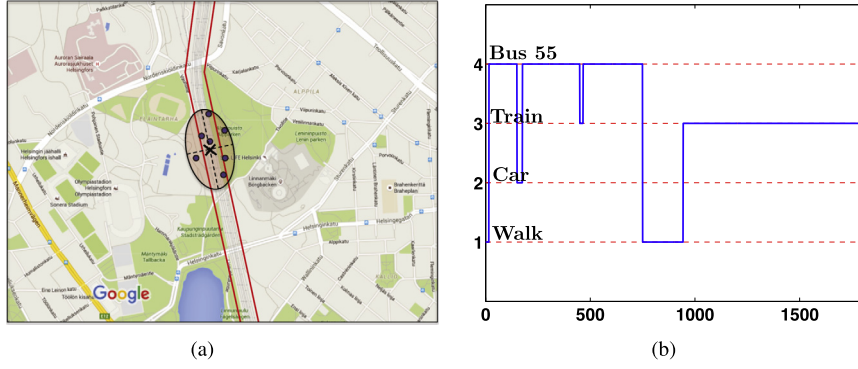


Fig. 8. (a) Example of generation of particles (shown as circles) using the dynamic model of type in Eq. (51). (b) Detection of transition between Bus 55, Walk and Train (MAP estimation) at the Helsinki central railway station, considering 4 possible modalities with real data ($T \approx 1900$ sec and $T_V = 150$ sec).

Dynamic model for car, cycling and walk. In this case, we consider a simple model of type

$$\mathbf{x}_t = \begin{cases} \mathbf{x}_{t-1} + b_1 \mathbf{v}_t, & \text{with prob. } \frac{1}{3}, \\ \mathbf{x}_{t-1} + b_2 \mathbf{v}_t, & \text{with prob. } \frac{1}{3}, \\ \mathbf{x}_{t-1} + b_3 \mathbf{v}_t, & \text{with prob. } \frac{1}{3}, \end{cases} \quad (52)$$

where $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $b_1 > b_2 > b_3 = 0.05$ are scalar parameters representing fast, moderate, and slow movements, respectively. The parameter $b_3 = 0.05$ corresponds to waiting in a bus stop or waiting for a traffic light switch. In our experiments, in a city environment, we have set $b_1 = 3.1$, $b_2 = 1.6$, for motorized vehicles, whereas $b_1 = 2$, $b_2 = 0.8$ for cycling and $b_1 = 1$, $b_2 = 0.3$ for modeling walkers. However, we note that the results are not strongly conditioned on these choices.

Our experience suggest of using $50 \leq T_V \leq 200$ seconds. With this range of values $50 \leq T_V \leq 200$, we obtain a percentage greater of 82% in the estimation of the true modality, in the experiments considering the collected data. Fig. 8(b) shows the results of a specific test taking into account 4 different modalities. Furthermore, we obtain an averaged MSE of 3.14 meters in the tracking estimation, considering all the different modalities. With the automatic adaptive procedure in Section 5 ($p_r = 0.1$), we obtain an MSE of 3.27 meters. Considering only buses and trains, we obtain an MSE of 1.17 meters (1.38 with the automatic-adaptive procedure). We have also implemented and tested the *multiple model bootstrap particle filter* proposed in [31] considering a uniform transition matrix for the underlying Markov process, required for modeling the transitions among the different modalities. Considering all the modalities, we have in this case an MSE of 4.76 meters, whereas considering only buses and trains an MSE of 1.59 meters. In any case, these MSEs are greater than the MSEs obtained by MAPF. The multiple model bootstrap particle filter probably requires a more accurate tuning of all the entries of the transition matrix of the underlying Markov process (they are always more than 1, unlike in MAPF where we have only T_V).

7. Conclusions

We have designed an interacting parallel sequential Monte Carlo scheme for inference in state space models and online model selection. The parallel particle filters collaborate for providing a global efficient estimate of the hidden states and an approximation of the probability of the models given the received measurements. The exchange of information among the filters takes place through the adaptation of the numbers of particles of each filter. A exhaustive theoretical derivation has been provided. The proposed technique has been applied successfully in different experimental scenarios, including a real data experiment for the mode detection in a urban mobility problem.

Acknowledgments

This work has been supported by the Grant 2014/23160-6 of São Paulo Research Foundation (FAPESP), the Grant 305361/2013-3 of National Council for Scientific and Technological Development (CNPq), and the Aalto University AEF research programme <http://energyefficiency.aalto.fi/en/>.

Appendix A. Recursive formulas for sequential inference

For simplifying the notation, in this section, we consider working only with a unique model \mathcal{M}_1 , so that $p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t} | \mathcal{M}_1)$. Thus, we can write easily the following recursive formula for the joint pdf $p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t})$ [13,15],

$$p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = \ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}).$$

However, we are interested in a similar recursive expression which involves the posterior $p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) = \frac{p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t})}{p(\mathbf{y}_{1:t})}$. Thus, starting be the definition and replacing the expression above, we obtain

$$p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) = \frac{\ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1})}{p(\mathbf{y}_{1:t})} p(\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}). \quad (\text{A.1})$$

Replacing $p(\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}) = p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) p(\mathbf{y}_{1:t-1})$, we have

$$p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) = \ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1}) \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{y}_{1:t})} p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}),$$

and since we can write $p(\mathbf{y}_{1:t}) = p(\mathbf{y}_t | \mathbf{y}_{1:t-1}) p(\mathbf{y}_{1:t-1})$, finally we obtain

$$\begin{aligned} p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) &= [\ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1})] \\ &\quad \times \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1}) p(\mathbf{y}_{1:t-1})} p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) \\ &= \frac{\ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1})} p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}). \end{aligned}$$

The last expression involves the posterior at the t -th iteration, $p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$, as function of the posterior at the $t - 1$ -th iteration, $p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1})$.

Appendix B. Derivation of the estimator \tilde{Z}_t

For simplicity, we again assume working only with a unique model \mathcal{M}_1 , so that $K = 1$ and $p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t} | \mathcal{M}_1) = p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t} | \mathcal{M}_1)$. Within the SIS framework, there are two possible formulations of the estimator of Z , i.e., \tilde{Z} in Eq. (27) and \tilde{Z} given in Eq. (28). The alternative formulation \tilde{Z} in Eq. (28) can be derived as follows [15]. Recall that

$$Z_t = \int_{\mathcal{X}_{1:t}} p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) d\mathbf{x}_{1:t} \approx \hat{Z}_t = \frac{1}{M} \sum_{m=1}^M w_t^{(m)},$$

and consider the following integral

$$\begin{aligned} & \int_{\mathcal{X}_{1:t}} \ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{1:t} \\ &= \frac{1}{p(\mathbf{y}_{1:t-1})} \int_{\mathcal{X}_{1:t}} \ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}) d\mathbf{x}_{1:t} \\ &= \frac{1}{p(\mathbf{y}_{1:t-1})} \int_{\mathcal{X}_{1:t}} p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) d\mathbf{x}_{1:t} \\ &= \frac{p(\mathbf{y}_{1:t})}{p(\mathbf{y}_{1:t-1})} = \frac{Z_t}{Z_{t-1}} = p(\mathbf{y}_t | \mathbf{y}_{1:t-1}), \end{aligned}$$

where in the last expression we have used $Z_t = p(\mathbf{y}_{1:t})$ and $p(\mathbf{y}_{1:t}) = p(\mathbf{y}_t | \mathbf{y}_{1:t-1}) p(\mathbf{y}_{1:t-1})$. Summarizing, we have obtained

$$\int_{\mathcal{X}_{1:t}} \ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{1:t} = \frac{Z_t}{Z_{t-1}}. \quad (\text{B.1})$$

Now we replace above $p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1})$ the particle approximation $\hat{p}(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) = \sum_{m=1}^M \bar{w}_{t-1}^{(m)} \delta(\mathbf{x}_{1:t-1} - \mathbf{x}_{1:t-1}^{(m)})$, so we can write

$$\begin{aligned} & \int_{\mathcal{X}_{1:t}} \ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1}) \hat{p}(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{1:t} \\ & \times \sum_{m=1}^M \bar{w}_{t-1}^{(m)} \int_{\mathcal{X}_t} \ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1}^{(m)}) d\mathbf{x}_t \approx \frac{Z_t}{Z_{t-1}}. \end{aligned} \quad (\text{B.2})$$

Moreover, approximating the M integrals $\int_{\mathcal{X}_t} \ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1}^{(m)}) d\mathbf{x}_t$ via Monte Carlo using *only one sample*, $\mathbf{x}_t^{(m)} \sim \phi_t(\mathbf{x}_t^{(m)} | \mathbf{x}_{1:t-1}^{(m)})$, for each one,

$$\begin{aligned} & \sum_{m=1}^M \bar{w}_{t-1}^{(m)} \int_{\mathcal{X}_t} \ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1}^{(m)}) d\mathbf{x}_t \\ & \approx \sum_{m=1}^M \bar{w}_{t-1}^{(m)} \frac{\ell_t(\mathbf{y}_t | \mathbf{x}_t^{(m)}) q_t(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(m)})}{\phi_t(\mathbf{x}_t^{(m)} | \mathbf{x}_{1:t-1}^{(m)})} \\ & = \sum_{m=1}^M \bar{w}_{t-1}^{(m)} \lambda_t^{(m)} \approx \frac{Z_t}{Z_{t-1}}. \end{aligned} \quad (\text{B.3})$$

Alternative derivation. Note that we can also deduce the past expression as following

$$\begin{aligned} \sum_{m=1}^M \bar{w}_{t-1}^{(m)} \lambda_t^{(m)} &= \frac{1}{\sum_{m=1}^M w_{t-1}^{(m)}} \sum_{m=1}^M w_{t-1}^{(m)} \lambda_t^{(m)} \\ &= \frac{1}{\sum_{m=1}^M w_{t-1}^{(m)}} \sum_{m=1}^M w_t^{(m)} = \frac{\frac{1}{M} \sum_{m=1}^M w_t^{(m)}}{\frac{1}{M} \sum_{m=1}^M w_{t-1}^{(m)}} \\ &= \frac{\hat{Z}_t}{\hat{Z}_{t-1}} \approx \frac{Z_t}{Z_{t-1}}. \end{aligned}$$

Equivalence with \hat{Z}_t . Setting $\hat{Z}_0 = 1$, we can obtain that

$$\tilde{Z}_t = \prod_{\tau=1}^t \left[\sum_{m=1}^M \bar{w}_{\tau-1}^{(m)} \lambda_{\tau}^{(m)} \right] = \hat{Z}_1 \frac{\hat{Z}_2}{\hat{Z}_1} \dots \frac{\hat{Z}_{t-1}}{\hat{Z}_{t-2}} \frac{\hat{Z}_t}{\hat{Z}_{t-1}} = \hat{Z}_t \approx Z, \quad (\text{B.4})$$

namely, the estimator in Eq. (28) is exactly equivalent the estimator (27).

B.1. Application of resampling

Consider again to approximate the integral in Eq. (B.2) via importance sampling. In this case, we assume to draw independent samples $\mathbf{x}_t^{(1)}, \dots, \mathbf{x}_t^{(M)}$ from the a different proposal pdf $\varphi(\mathbf{x}_{1:t})$, defined as

$$\varphi(\mathbf{x}_{1:t}) = \phi_t(\mathbf{x}_t | \mathbf{x}_{1:t-1}) \hat{p}(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) = \sum_{m=1}^M \bar{w}_{t-1}^{(m)} \phi_t(\mathbf{x}_t | \mathbf{x}_{1:t-1}^{(m)}).$$

Note that, this is equivalent to apply a resampling at the $(t-1)$ -th iteration. Thus, we can write

$$\begin{aligned} & \int_{\mathcal{X}_{1:t}} \ell_t(\mathbf{y}_t | \mathbf{x}_t) q_t(\mathbf{x}_t | \mathbf{x}_{t-1}) \hat{p}(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{1:t} \\ & \approx \frac{1}{M} \sum_{m=1}^M \frac{\ell_t(\mathbf{y}_t | \mathbf{x}_t^{(m)}) q_t(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(m)})}{\phi_t(\mathbf{x}_t^{(m)} | \mathbf{x}_{1:t-1}^{(m)})} = \frac{1}{M} \sum_{m=1}^M \lambda_t^{(m)} \\ & \approx \frac{Z_t}{Z_{t-1}}, \end{aligned} \quad (\text{B.5})$$

where $\mathbf{x}_t^{(m)} \sim \phi_t(\mathbf{x}_t | \mathbf{x}_{1:t-1}) \hat{p}(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1})$, with $m = 1, \dots, M$. Recalling the definition of $\xi_t^{(m)}$,

$$\xi_t^{(m)} = \begin{cases} w_t^{(m)}, & \text{without resampling at the } t\text{-th iteration,} \\ \hat{Z}_t, & \text{after resampling at the } t\text{-th iteration,} \end{cases} \quad (\text{B.6})$$

we can ensure that $\hat{Z}_t = \sum_{m=1}^M \xi_t^{(m)} \lambda_t^{(m)}$, is still a valid estimator of Z . When no resampling is performed, $\xi_{t-1}^{(m)} = w_{t-1}^{(m)}$, we come back to the standard IS estimator of Z . When the resampling is applied, we have $\xi_{t-1}^{(m)} = \hat{Z}_{t-1}$, then $\hat{Z}_t = \frac{\hat{Z}_{t-1}}{M} \sum_{m=1}^M \lambda_t^{(m)}$. Since $\frac{\hat{Z}_t}{\hat{Z}_{t-1}} = \frac{1}{M} \sum_{m=1}^M \lambda_t^{(m)}$, we obtain

$$\hat{Z}_t = \hat{Z}_{t-1} \left[\frac{1}{M} \sum_{m=1}^M \lambda_t^{(m)} \right] = \hat{Z}_{t-1} \frac{\hat{Z}_t}{\hat{Z}_{t-1}} \approx Z.$$

Finally, note that \hat{Z}_t and $\tilde{Z}_t = \prod_{\tau=1}^t \left[\sum_{m=1}^M \bar{w}_{\tau-1}^{(m)} \lambda_{\tau}^{(m)} \right]$ are two equivalent formulations of the same estimator. It can be shown exactly as described above for the SIS framework, replacing $w_t^{(m)}$ with $\xi_t^{(m)}$. Furthermore, if the resampling is applied at each iteration, observe that both estimators become

$$\tilde{Z}_t = \prod_{\tau=1}^t \left[\frac{1}{M} \sum_{j=1}^M \lambda_{\tau}^{(j)} \right], \quad (\text{B.7})$$

and

$$\hat{Z}_t = \hat{Z}_{t-1} \left[\frac{1}{M} \sum_{m=1}^M \lambda_t^{(m)} \right] = \prod_{\tau=1}^t \left[\frac{1}{M} \sum_{j=1}^M \lambda_{\tau}^{(j)} \right]. \quad (\text{B.8})$$

Note that, with respect to the estimator in Eq. (27) (for SIS, i.e., without resampling), the operations of product and sum are inverted.

Appendix C. Further considerations about MAPF

The validity of the MAPF scheme relies on each filter performs separately a proper SIR estimation of the hidden states using the normalized weights $\bar{w}_{k,t}^{(i_k)}$. Each filter also provides a consistent estimator $\hat{Z}_{k,t}$ of the marginal likelihood. Then, this information is

properly merged following the approximation of $\hat{p}(\mathbf{x}|\mathbf{y})$ given in Eq. (15) or (20). The variable numbers of particles does not provide any theoretical issues, since we obtain always valid IS estimators (clearly, it affects the efficiency of these estimators) [16]. An important related observation is remarked below.

Remark 5. Consider that, at the t -th iteration, the condition in Eq. (37) is satisfied. Namely, (a) the numbers of particles are updated and (b) the resampling applied. Considering jointly (a) and (b), we can be interpreted that in MAPF we are drawing samples from the mixture in Eq. (20),

$$\hat{p}(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = \sum_{k=1}^K \bar{\rho}_{k,t} \hat{p}(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}, \mathcal{M}_k),$$

using the so-called deterministic mixture (DM) procedure [16]. The DM is performed through the adaptation of the number of particles.

That is, the adaptation of number of particles can be seen as a way of selecting (i.e., using) more times one model than other. The previous remark suggests an alternative equivalent resampling scheme:

1. Set $i_k = 0$, for all $k = 1, \dots, K$.
2. For $n = 1, \dots, N$:
 - (a) Select a model k with probability $\bar{\rho}_{k,t}$, $k = 1, \dots, K$.
 - (b) Set $i_k = i_k + 1$ resample $\bar{\mathbf{x}}_{k,t}^{(i_k)} = \mathbf{x}_{k,t}^{(j)} \in \{\mathbf{x}_{k,t}^{(1)}, \dots, \mathbf{x}_{k,t}^{(M_{k,t})}\}$ with probability $\bar{w}_{k,t}^{(j)}$, $j = 1, \dots, M_{k,t}$.
3. Set $M_{k,t+1} = i_k$, for all $k = 1, \dots, K$, so that $\sum_{k=1}^K M_{k,t+1} = N$.

Clearly, with the procedure above, some model could be completely discarded, with no particles assigned.

Appendix D. Supplementary material

Supplementary material related to this article can be found online at <http://dx.doi.org/10.1016/j.dsp.2016.09.011>.

References

- [1] K. Achutegui, L. Martino, J. Rodas, C.J. Escudero, J. Míguez, A multi-model particle filtering algorithm for indoor tracking of mobile terminals using rss data, in: IEEE Intelligent Control and Control Applications, CCA, 2009, pp. 1702–1707.
- [2] C. Andrieu, A. Doucet, R. Holenstein, Particle Markov chain Monte Carlo methods, *J. R. Stat. Soc. B* 72 (3) (2010) 269–342.
- [3] C. Andrieu, A. Doucet, S.S. Singh, V.B. Tadić, Particle methods for change detection, system identification and control, *Proc. IEEE* 92 (3) (March 2004) 423–438.
- [4] M.S. Arulampalam, S. Maskell, N. Gordon, T. Klapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Trans. Signal Process.* 50 (2) (February 2002) 174–188.
- [5] L. Bazzani, D. Bloisi, V. Murino, A comparison of multi-hypothesis Kalman filter and particle filter for multi-target tracking, in: Performance Evaluation of Tracking and Surveillance Workshop at CVPR 2009, Miami, Florida, 2009, pp. 47–54.
- [6] J.M. Bernardo, A.F.M. Smith, Bayesian Theory, Wiley & Sons, 1994.
- [7] Y. Boers, J.N. Driessen, Interacting multiple model particle filter, *IEE Proc. Radar Sonar Navig.* 150 (5) (2003) 344–349.
- [8] M. Bolić, P.M. Djurić, S. Hong, Resampling algorithms and architectures for distributed particle filters, *IEEE Trans. Signal Process.* 53 (7) (July 2005) 2442–2450.
- [9] M.F. Bugallo, L. Martino, J. Corander, Adaptive importance sampling in signal processing, *Digit. Signal Process.* 47 (2015) 36–49.
- [10] M.F. Bugallo, S. Xu, P.M. Djurić, Performance comparison of EKF and particle filtering methods for maneuvering targets, *Digit. Signal Process.* 17 (October 2007) 774–786.
- [11] F. Caron, Manuel Davy, E. Duflos, P. Vanheeghe, Particle filtering for multisensor data fusion with switching observation models: application to land vehicle positioning, *IEEE Trans. Signal Process.* 55 (6) (June 2007) 2703–2719.
- [12] O. Demirel, I. Smal, W. Niessen, E. Meijering, I.F. Sbalzarini, A parallel particle filtering library, arXiv:1310.5045, 2013.
- [13] P.M. Djurić, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, J. Míguez, Particle filtering, *IEEE Signal Process. Mag.* 20 (5) (September 2003) 19–38.
- [14] P.M. Djurić, J. Míguez, Assessment of nonlinear dynamic models by Kolmogorov–Smirnov statistics, *IEEE Trans. Signal Process.* 58 (10) (2010) 5069–5079.
- [15] A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer, New York, USA, 2001.
- [16] V. Elvira, L. Martino, D. Luengo, M. Bugallo, Efficient multiple importance sampling estimators, *IEEE Signal Process. Lett.* 22 (10) (2015) 1757–1761.
- [17] V. Elvira, J. Míguez, P.M. Djurić, Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment, arXiv preprint, arXiv:1509.04879, 2015.
- [18] W.J. Fitzgerald, Markov chain Monte Carlo methods with applications to signal processing, *Signal Process.* 81 (1) (January 2001) 3–18.
- [19] D. Fox, Kld-sampling: adaptive particle filters, in: Advances in Neural Information Processing Systems, 2001, pp. 713–720.
- [20] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Jansson, R. Karlsson, P.-J. Nordlund, Particle filters for positioning, navigation and tracking, *IEEE Trans. Signal Process.* 50 (2) (February 2002) 425–437.
- [21] J.A. Hoeting, D. Madigan, A.E. Raftery, Chris T. Volinsky, Bayesian model averaging: a tutorial, *Stat. Sci.* 14 (4) (1999) 382–417.
- [22] J. Hollmén, V. Tresp, Call-based fraud detection in mobile communications networks using a hierarchical regime-switching model, in: M. Kearns, S.olla, D.A. Cohn (Eds.), Proceedings of the 1998 Conference, NIPS’11, in: Adv. Neural Inf. Process. Syst., vol. 11, MIT Press, 1999, pp. 889–895.
- [23] M. Hong, M.F. Bugallo, P.M. Djurić, Joint model selection and parameter estimation by population Monte Carlo simulation, *IEEE J. Sel. Top. Signal Process.* 4 (3) (2010) 526–539.
- [24] D.S. Lee, N.K.K. Chia, A particle algorithm for sequential Bayesian parameter estimation and model selection, *IEEE Trans. Signal Process.* 50 (2) (2002) 326–336.
- [25] J.S. Liu, Monte Carlo Strategies in Scientific Computing, Springer, 2004.
- [26] L. Martino, V. Elvira, F. Louzada, Alternative Effective Sample Size measures for importance sampling, in: IEEE Statistical Signal Processing Workshop, SSP, 2016.
- [27] L. Martino, V. Elvira, F. Louzada, Weighting a resampled particle in Sequential Monte Carlo, in: IEEE Statistical Signal Processing Workshop, SSP, 2016.
- [28] L. Martino, V. Elvira, F. Louzada, Effective Sample Size for importance sampling based on discrepancy measures, *Signal Process.* 131 (2017) 386–401.
- [29] L. Martino, J. Míguez, A generalization of the adaptive rejection sampling algorithm, *Stat. Comput.* 21 (4) (July 2011) 633–647.
- [30] L. Martino, H. Yang, D. Luengo, J. Kanninen, J. Corander, A fast universal self-tuned sampler within Gibbs sampling, *Digit. Signal Process.* 47 (2015) 68–83.
- [31] S. McGinnity, G.W. Irwin, Maneuvering target tracking using a multiple-model bootstrap filter, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer, 2001, pp. 479–496, chapter 23.
- [32] J. Míguez, M.F. Bugallo, P.M. Djurić, A new class of particle filters for random dynamical systems with unknown statistics, *EURASIP J. Adv. Signal Process.* 2004 (15) (November 2004) 2278–2294.
- [33] C.P. Robert, G. Casella, Monte Carlo Statistical Methods, Springer, 2004.
- [34] K. Salhi, M. Deaconu, A. Lejay, N. Champagnat, N. Navet, Regime switching model for financial data: empirical risk analysis, *Physica A* 461 (2016) 148–157, <http://dx.doi.org/10.1016/j.physa.2016.05.002>.
- [35] G. Storvik, Particle filters for state-space models with the presence of unknown static parameters, *IEEE Trans. Signal Process.* 50 (2) (February 2002) 281–289.
- [36] B. Tian, C.M. Chew, H. Tang, M. Yuan, An intuitive and efficient switching particle filter for real-time vision-based localization, in: IEEE Conference on Cybernetics and Intelligent Systems, CIS, Nov. 2013, pp. 7–11.
- [37] W. Wu, M.J. Black, D. Mumford, Y. Gao, E. Bienenstock, J.P. Donoghue, Modeling and decoding motor cortical activity using a switching Kalman filter, *IEEE Trans. Biomed. Eng.* 51 (6) (June 2004) 933–942.
- [38] N. Zorbas, D. Zissis, K. Tserpes, D. Anagnostopoulos, Predicting object trajectories from high-speed streaming data, in: 9th IEEE International Conference on Big Data Science and Engineering, IEEE BDSE-15, 2015.

Luca Martino obtained his Ph.D. in Statistical Signal Processing from Universidad Carlos III de Madrid, Spain, in 2011. He was an Assistant Professor in the Department of Signal Theory and Communications at Universidad Carlos III de Madrid since then. In August 2013 he joined the Department of Mathematics and Statistics at the University of Helsinki. In March 2015, he joined the Universidade de São Paulo (USP). Currently, he is a post-doctoral researcher at the University of Valencia. His research interests include Bayesian inference, Monte Carlo methods and exact methods for the generation of random variables.

Jesse Read obtained his Ph.D. in Computer Science from the University of Waikato, New Zealand, in 2010. He then worked as a postdoc researcher

in the Department of Signal Theory and Communications at the Carlos III University of Madrid, and then in the Department of Information and Computer Science at Aalto University in Finland. Currently he is working in the Computer Science and Networks Department in Telecom ParisTech in France.

Víctor Elvira obtained his M.Sc. and Ph.D. degrees in Electrical Eng. from Universidad de Cantabria (Spain) in 2007, 2008 and 2011, respectively. Since 2012 he is postdoctoral researcher at Universidad Carlos III de Madrid (Spain). He has been a visiting scholar at the IHP Institute (Frankfurt Oder, Germany), University of Helsinki (Finland), and Stony Brook

University (NY, USA). His research interests include statistical signal processing, and biomedical signal processing.

Francisco Louzada is Professor of Statistics in the Institute of Mathematical Science and Computing at the University of São Paulo (USP) in Brazil, and Director of Technology Transfer of the Center for Research, Innovation and Dissemination of Mathematical Science in Industry. He did his Ph.D. in Statistics from University of Oxford (1998), Master in Computer Science and Computational Mathematics from USP (1991) and Bachelor of Statistics from UFSCar (1988). His main Areas of Interest are Big Data Analysis, Statistical Inference and Survival Analysis.