COOPERATIVE PARALLEL PARTICLE FILTERS FOR ON-LINE MODEL SELECTION AND APPLICATIONS TO URBAN MOBILITY

L. Martino[⋆], *J. Read*[†], *V. Elvira*[◊], *F. Louzada*[⋆]

[⋆] Institute of Mathematical Sciences and Computing, Universidade de São Paulo, São Carlos (Brazil).
[†] Dep. of Mathematics and Statistics, University of Helsinki, Helsinki (Finland).
[◊] Dep. of Signal Theory and Communic., Universidad Carlos III de Madrid, Leganés (Spain).

ABSTRACT

We design a sequential Monte Carlo scheme for the joint purpose of Bayesian inference and model selection, with application to urban mobility context where different modalities of transport and measurement devices can be employed. In this case, we have 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 the models given the data. The interaction occurs by a parsimonious distribution of the computational effort, adapting on-line 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.

Keywords: Sequential model selection; Modality detection; tracking; marginal likelihood estimation; parallel particle filters; distributed inference; adaptive complexity.

1. INTRODUCTION

Monte Carlo (MC) algorithms are very popular numerical techniques for the approximation of optimal a posteriori estimators [16, 11, 29], given an analytically intractable posterior probability density function (pdf). More specifically, Particle filters (PFs) have been extensively applied for sequential Bayesian inference [29] in machine learning [28, 31, 32], signal processing [4, 18] and statistics [13, 23]. 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 [22]. A finite set of candidate models are considered, and the true model may be included (or not) within this set. Thus, the goal is both sequential tracking and online model selection. This is exactly required in a urban mobility context where different modalities of transport can be employed (for instance bus, train, cycling etc.). The main contributions of this work, range of applicability, and related works are discussed below.

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 [19]. 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 for providing a global approximation of the posterior distribution of the variable of interest given the data and, at the 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 on-line 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 distributes the computational effort on-line, rewarding the filters addressing the most probable models given the 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 is described in Section 3. The sequential IS approach is presented in Section 3.1. The use of parallel particle filters appears natural following the theoretical derivation of MAPF. We have focused special attention in describing wthe 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 4.1. For instance, when all the candidate models share the 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 for obtaining 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 real data problem: a urban mobility context for detecting different modes of activity. For instance, one possible goal in urban mobility is to detect if a traveller is walking or riding a bus (or switching between these two modalities). The numerical results, with both synthetic and real data, show the efficiency and flexibility of the proposed algorithms (Section 6).

Range of applicability. The range of applicability of MAPF is clearly wider, not restricted only to the urban mobility case. Application of sequential model selection problem are very common in different fields where a stream of data is observed [8, 12, 21, 22]. For instance, in financial analysis where price fluctuations act differently under different regimes, as the ongoing instability in times of crisis [30]. They have also been used for fraud detection [20], 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 [36]. In marine tracking [37], vessels must be suitable monitored depending on the 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, 33]), selecting them within of a set of possible candidates (e.g., see Section 6.2).

Related works. The problem addressed in this work is strictly related the *switching model* [9, 34, 25, 6] and the *multiple model* approaches [1, 5, 8]. 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. MAPF is a more simple 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 of the dynamic and likelihood functions.[1] 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 always allowed, in general. This difference yields that MAPF provides 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 has been proposed in literature [17, 15], 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, 33]. In [12], 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 [13, 27, 26].

2. 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

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

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 $\mathcal{M}_k$,

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

with $k \in \{1, \dots, K\}$, formed by a transition pdf and a likelihood function, i.e., setting $q_{k,1}(\mathbf{x}_1|\mathbf{x}_0) = q_{k,1}(\mathbf{x}_1)$,

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

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 [19] 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

$$\begin{aligned} p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) &= \sum_{k=1}^{K} p(\mathbf{x}_{1:T}, \mathcal{M}_k|\mathbf{y}_{1:T}), && (3)\\ &= \sum_{k=1}^{K} p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \mathcal{M}_k) p(\mathcal{M}_k|\mathbf{y}_{1:T}), && (4)\\ &= \frac{1}{p(\mathbf{y}_{1:T})} \sum_{k=1}^{K} p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}, \mathcal{M}_k) p(\mathbf{y}_{1:T}|\mathcal{M}_k) p(\mathcal{M}_k), && (5) \end{aligned}$$

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:

- We cannot evaluate $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ in Eq. (5) completely, since we cannot evaluate

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

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

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

  for all $k = 1, \dots, K$. However, given an index $k \in \{1, \dots, K\}$ and $\mathbf{y}_{1:T}$, we are able to evaluate[2]

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

  so that we can also evaluate

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

  up to a normalizing constant.

- Moreover, often we 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 $\mathbf{x}_{1:T}$,

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

  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.

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).

[2]For the sake of simplicity, in the following mathematical elaborations, we consider both densities $q_{k,t}$ and $\ell_{k,t}$ 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.

3. MONTE CARLO APPROXIMATION VIA IMPORTANCE SAMPLING

For solving the issues described in the previous section, we consider the use of Monte Carlo techniques. First of all, fixing an index $k \in \{1, \dots, K\}$, we describe a *batch importance sampling* (IS) approach where 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 $\varphi_k : \mathcal{X}^T \to \mathbb{R}$, with $m = 1, \dots, M$. Namely, in the batch approach, we consider samples directly in the whole space $\mathcal{X}^T$. In the next section, we introduce the corresponding sequential scheme (working sequentially in the subspaces $\mathcal{X}$). 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 $\varphi_k : \mathcal{X}^T \to \mathbb{R}$, and assign to each sample the following importance weights

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

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 [29, 23] as[3]

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

We can also write $p(\mathbf{y}_{1:T}) \approx \sum_{k=1}^{K} \widehat{Z}_{k,T} p(\mathcal{M}_k)$. Furthermore, since , 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

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

where

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

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 $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$,

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

The last expression can summarized as

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

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

where we have denoted

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

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

It is particular interesting from a theoretical point of view that the weight $\bar{\gamma}_{k,T}^{(m)}$ can be decomposed[4] as

$$\begin{aligned} \bar{\gamma}_{k,T}^{(m)} &= \frac{w_{k,T}^{(m)}}{\sum_{j=1}^{M} w_{k,T}^{(j)}} \frac{\widehat{Z}_{k,T} p(\mathcal{M}_k)}{\sum_{j=1}^{K} \widehat{Z}_{j,T} p(\mathcal{M}_j)}. \\ &= \bar{w}_{k,T}^{(m)} \bar{\rho}_{k,T}, \end{aligned} \tag{18}$$

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

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

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

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

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

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

or with the equivalent two-stage formula

$$\begin{cases} \widetilde{\mathbf{I}}_{k,1:T} = \sum_{m=1}^{M} \bar{w}_{k,T}^{(m)} \mathbf{x}_{k,1:T}^{(m)}, \\ \widehat{\mathbf{I}}_{1:T} = \sum_{k=1}^{K} \bar{\rho}_{k,T} \widetilde{\mathbf{I}}_{k,1:T}, \end{cases} \tag{22}$$

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

3.1. Sequential Importance Sampling (SIS)

The IS method can be alternatively perfomed in a sequential manner, i.e., providing an approximation of the posterior pdf at each iteration t using the previous approximation ant 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 [11], respectively,

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

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

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

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

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

$$\varphi_k(\mathbf{x}_{k,1:T}) = \phi_{k,1}(\mathbf{x}_{k,1})\phi_{k,2}(\mathbf{x}_{k,2}|\mathbf{x}_{k,1})\cdots\phi_{k,T}(\mathbf{x}_{k,T}|\mathbf{x}_{k,1:T-1}), \tag{25}$$

with $\phi_{k,t} : \mathcal{X} \to \mathbb{R}$ for $t = 1, \ldots, T$. Given Eq. (23), we can infer a recursive relationship between two importance weights at consecutive iterations [11],

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

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 $m = 1, \ldots, M$. Therefore, given the recursive expression of $w_{k,t}^{(m)}$, we can rewrite the estimator $\widehat{Z}_{k,t}$ as

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

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),

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

where $\bar{w}_{k,t-1}^{(i)} = \frac{w_{k,t-1}^{(i)}}{\sum_{n=1}^{N} w_{k,t-1}^{(n)}}$ 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 $p(\mathbf{y}_{1:t}|\mathcal{M}_k)$,

$$\begin{aligned} \widetilde{Z}_{k,t} &= \prod_{\tau=1}^{t}\left[\sum_{j=1}^{M} \bar{w}_{k,\tau-1}^{(j)}\lambda_{k,\tau}^{(j)}\right], &(28)\\ &= \prod_{\tau=1}^{t}\left[\frac{\sum_{j=1}^{M} w_{k,\tau}^{(j)}}{\sum_{j=1}^{M} w_{k,\tau-1}^{(j)}}\right]. &(29) \end{aligned}$$

In Appendix B, we show a complete derivation of the estimator $\widetilde{Z}_{k,t}$ and that $\widetilde{Z}_{k,t} \equiv \widehat{Z}_{k,t}$, as one could easily realize from Eq. (29).

Remark 1. *Observe that, in SIS, there are two possible equivalent formulation of the estimator of $p(\mathbf{y}_{1:t}|\mathcal{M}_k)$, i.e., $\widehat{Z}_{k,t}$ in Eqs.* (27) *$\widetilde{Z}_{k,t}$ in Eq.* (28)*, and they are the equivalent, $\widetilde{Z}_{k,t} \equiv \widehat{Z}_{k,t}$ (see App. B).*

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

$$\bar{\rho}_{k,t} = \frac{\widehat{Z}_{k,t}p(\mathcal{M}_k)}{\sum_{j=1}^{M}\widehat{Z}_{j,t}p(\mathcal{M}_j)} \approx p(\mathcal{M}_k|\mathbf{y}_{1:t}). \tag{30}$$

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

3.2. Sequential Importance Resampling (SIR)

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 [11, 13]. 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 $\bar{w}_{k,t-1}^{(j)}$, $j = 1, \dots, M$, 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 [11, 13][5], i.e,

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

A proper value[6] for the (unnormalized) importance weight $\alpha_{1:t-1}^{(j)}$ associated with the j-th resampled particle is

$$\alpha_{k,t}^{(j)} = \widehat{Z}_{k,t} = \frac{1}{N} \sum_{i=1}^{N} w_{k,t}^{(i)}, \quad \forall j = 1, \dots, M. \tag{31}$$

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

$$\xi_{k,t}^{(m)} = \begin{cases} w_{k,t}^{(m)}, & \text{without resampling at } t\text{-th iteration,} \\ \alpha_{k,t}^{(m)}, & \text{with resampling at } t\text{-th iteration.} \end{cases} \tag{32}$$

then, in any case,

$$\frac{1}{N} \sum_{n=1}^{N} \xi_{k,t}^{(j)} = \widehat{Z}_{k,t},$$

as expected. In general, the resampling steps are not applied at each iterations, but only when some statistical criterion is satisfied [7, 11, 13] (e.g., see Section 4). The recursive expression of the weights for SIR becomes

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

Remark 2. *With the recursive definition of the weights $\xi_{k,t}^{(m)}$'s in Eq.* (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*

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

where $\bar{\xi}_{k,t-1}^{(j)} = \frac{\xi_{k,t-1}^{(j)}}{\sum_{m=1}^{M} \xi_{k,t-1}^{(m)}}$. They are equivalent and valid estimators for the reasons shown in App. B. Furthermore, if the resampling is applied at each iteration, observe that they become

$$\widetilde{Z}_{k,t} = \prod_{\tau=1}^{t} \left[\frac{1}{M} \sum_{j=1}^{M} \lambda_{k,\tau}^{(j)} \right], \tag{35}$$

and

$$\widehat{Z}_{k,t} = \widehat{Z}_{k,t-1} \left[\frac{1}{M} \sum_{m=1}^{M} \lambda_{k,t}^{(m)} \right] = \prod_{\tau=1}^{t} \left[\frac{1}{M} \sum_{j=1}^{M} \lambda_{k,\tau}^{(j)} \right]. \tag{36}$$

Note that, w.r.t. the estimator in Eq. (27) *(for SIS, i.e., without resampling), the operations of product and sum are inverted.*

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.

[5] This is a proper choice, but it is not unique; see "*Concept of weighted sample*" in [23, Chapter 2] or [29, Section 14.2].

[6] It is suitable in the sense of weighted sample described in [23, Chapter 2].

Table 1: **Main notation for MAPF.**

K	Number of parallel PFs.
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.
$\widetilde{Z}_{k,t}, \widehat{Z}_{k,t}$	estimator of $p(\mathbf{y}_{1:t}\|\mathcal{M}_k)$ (two 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 $p(\mathcal{M}_k\|\mathbf{y}_{1:t})$.
$\bar{\gamma}_{k,t}^{(m)}$	Global normalized weight ($\bar{\gamma}_{k,t}^{(m)} = \bar{w}_{k,t}^{(m)}\bar{\rho}_{k,t}$).
$\mathbf{I}_{1:t} = E[\mathbf{x}_{1:t}]$	MMSE estimator of the hidden state $\mathbf{x}_{1:t}$.
$\widetilde{\mathbf{I}}_{1:t}$	Partial approximation of $\mathbf{I}_{1:t}$ of the k-th PF.
$\widehat{\mathbf{I}}_{1:t}$	Global approximation of $\mathbf{I}_t$.

4. MODEL AVERAGING PARTICLE FILTERS

The factorization of the global normalized weights $\bar{\gamma}_{k,t}^{(m)} = \bar{w}_{k,t}^{(m)}\bar{\rho}_{k,t}$ suggests the use of K parallel IS schemes. Indeed, each IS scheme can compute independently $\bar{w}_{k,t}^{(m)}$ and $\widehat{Z}_{k,t}$, and then they merge all the information for calculating $\bar{\rho}_{k,t}$ (see Figure 1). Therefore, we consider K parallel particle filters [11, 7, 10] using the transition model as proposal pdf[7]

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

each one tailored to 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.

Table 2 provides an exhaustive description of the *Model Averaging Particle Filters* (MAPF). Table 1 summarizes the main notation used in 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 on-line, 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.,

$$M_{1,t} + M_{2,t} + \ldots + M_{K,t} = N, \qquad \text{for} \quad t = 1, \dots, T. \tag{37}$$

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 $\widehat{\mathbf{x}}_t$ in Eq. (10) can be approximated using the Eq. (21) or (22), i.e.,

$$\widehat{\mathbf{I}}_t = \sum_{k=1}^{K} \bar{\rho}_{k,t}\widetilde{\mathbf{I}}_{k,t} \quad \text{where} \quad \widetilde{\mathbf{I}}_{k,t} = \sum_{i_k=1}^{M_{k,t}} \bar{w}_{k,t}^{(i_k)}\mathbf{x}_{k,t}^{(i_k)}, \tag{38}$$

where $\widetilde{\mathbf{I}}_{k,t}$ is provided by each filter independently, as shown in Figure 1. We have denoted as $\bar{\mathbf{x}}_{k,1:t}^{(i_k)} \in \{\mathbf{x}_{k,1:t}^{(1)}, \dots, \mathbf{x}_{k,1:t}^{(M_{k,t})}\}$ 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 a well-known approximation of ESS [11, 18, 7] for our problem, so finally the used condition is

$$\widehat{ESS} = \frac{1}{\sum_{k=1}^{K}\sum_{m=1}^{M_{k,t}} \left(\bar{\gamma}_{k,t}^{(m)}\right)^2} \leq \epsilon N, \tag{39}$$

where $\bar{\gamma}_{k,t}^{(m)} = \bar{w}_{k,t}^{(m)}\bar{\rho}_{k,t}$. Namely, when $\widehat{ESS} \leq \epsilon N$, a resampling step is performed.

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

[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 constrain is not required.

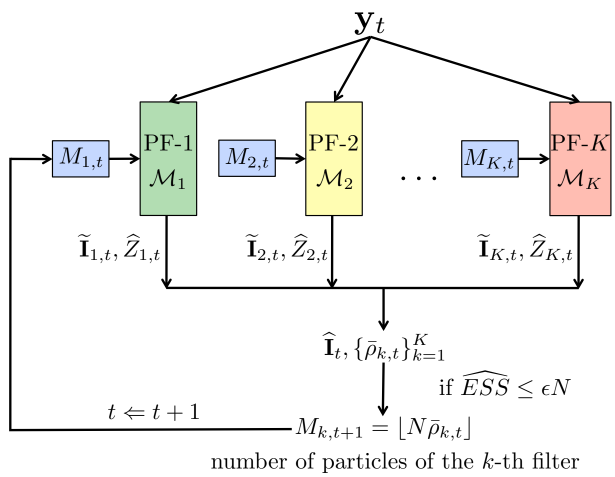


Fig. 1: Graphical overview of the MAPF scheme in Table 2 for approximating the MMSE estimator $\widehat{\mathbf{I}}_t$ and the posterior of each model given the observed data. Note that $\widehat{ESS}$ in Eq. (39) takes into account the (normalized) weights $\bar{\gamma}_{k,t}^{(i_k)} = \bar{w}_{k,t}^{(i_k)} \bar{\rho}_{k,t}$, for all i_k and k.

Table 2: **Model Averaging Particle Filters (MAPF)**.

1. **Initialization:** Set $M_{k,1} = \frac{N}{K} \geq 2$, $\xi_{k,0}^{(i_k)} = 1$, and choose the initial states $\bar{\mathbf{x}}_{k,0}^{(i_k)}$, for all $i_k = 1, \ldots, M_{k,1}$ and $k = 1, \ldots, K$.
2. For $t = 1, \ldots, T$:
   (a) **Propagation:** Draw $\mathbf{x}_{k,t}^{(i_k)} \sim q_{k,t}(\mathbf{x}|\bar{\mathbf{x}}_{k,t-1}^{(i_k)})$, for $i_k = 1, \ldots, M_{k,t}$ and $k = 1, \ldots, K$.
   (b) **Particle Weighting:** Compute the weights and normalized them,
   $$w_{k,t}^{(i_k)} = \xi_{k,t-1}^{(i_k)} \ell_{k,t}(\mathbf{y}_t|\mathbf{x}_{k,t}^{(i_k)}), \quad \text{and} \quad \bar{w}_{k,t}^{(i_k)} = \frac{w_{k,t}^{(i_k)}}{\sum_{j=1}^{M_{k,t}} w_{k,t}^{(j)}}, \tag{40}$$
   for $i_k = 1, ..., M_{k,t}$, $k = 1, \ldots, K$ and $\xi_{k,t-1}^{(i_k)}$ is defined in Eq. (33).
   (c) **Model Weighting:** Compute, for $k = 1, \ldots, K$,
   $$\widehat{Z}_{k,t} = \frac{1}{M_{k,\tau}} \sum_{i_k=1}^{M_{k,\tau}} w_{k,t}^{(i_k)}, \text{ and } \bar{\rho}_{k,t} = \frac{\widehat{Z}_{k,t} p(\mathcal{M}_k)}{\sum_{j=1}^K \widehat{Z}_{j,t} p(\mathcal{M}_j)}. \tag{41}$$
   Alternatively, the estimator $\widehat{Z}_{k,t}$ in Eq. (34) can be used.
   (d) **Adaptation and Resampling:** For each filter, $k = 1, \ldots, K$:
   - If the condition (39) is fulfilled,
     i. Set $M_{k,t+1} = \lfloor N\bar{\rho}_{k,t} \rfloor$, 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).
     ii. Draw $M_{k,t+1}$ samples, $\bar{\mathbf{x}}_{k,1:t}^{(1)}, \ldots, \bar{\mathbf{x}}_{k,1:t}^{(M_{k,t+1})}$, from
     $$\widehat{p}(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}, \mathcal{M}_k) = \sum_{i_k=1}^{M_{k,t}} \bar{w}_{k,t}^{(i_k)} \delta(\mathbf{x}_{1:t} - \mathbf{x}_{k,1:t}^{(i_k)}).$$
   - Otherwise, set $M_{k,t+1} = M_{k,t}$ and $\bar{\mathbf{x}}_{k,1:t}^{(i_k)} = \mathbf{x}_{k,1:t}^{(i_k)}$, for all i_k.
   (e) **Output:** Return $\{\mathbf{x}_{k,1:t}^{(i_k)}, w_{k,t}^{(i_k)}, \bar{\rho}_{k,t}\}$, $i_k = 1, \ldots, M_{k,t}$ and $k = 1, \ldots, K$.

Figure 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, Figure 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.

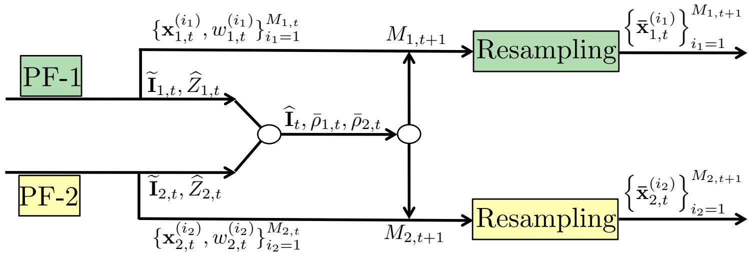


Fig. 2: Another graphical representation of the MAPF scheme, with $K = 2$ parallel particle filters. A fusion center returns, $\widehat{\mathbf{I}}_t$, $\bar{\rho}_{k,t}$, and the numbers of particles $M_{k,t+1}$, $k = 1, 2$, of the next iterations. Then, the resampling steps are performed separately.

Remark 3. *The step 2d of MAPF in Table 2 can be interpreted as a double resampling: a first resampling considering the model weights, $\bar{\rho}_{k,t}$, adapting the number of particles and the second one considering the normalized weights, $\bar{w}_{k,t}^{(m)}$, 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 weights $\bar{\gamma}_{k,t}^{(m)}$=$\bar{w}_{k,t}^{(m)}\bar{\rho}_{k,t}$, where the exchange of particles is possible.*

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

4.1. Interesting special cases

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 likelihood is

$$p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}) = \prod_{\tau=1}^{t} \ell_\tau(\mathbf{y}_\tau|\mathbf{x}_\tau), \tag{42}$$

and the complete prior pdf of k-th model, is[9]

$$p(\mathbf{x}_{1:t}|\mathcal{M}_k) = q_{k,1}(\mathbf{x}_1) \prod_{\tau=2}^{t} q_{k,\tau}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}). \tag{43}$$

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

$$p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}) \left[\sum_{k=1}^{K} \zeta_k \, p(\mathbf{x}_{1:t}|\mathcal{M}_k)\right], \tag{44}$$

where we have denoted $\zeta_k = p(\mathcal{M}_k|\mathbf{y}_{1:T})$. Namely, 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, $p(\mathbf{x}_{1:t}) = \sum_{k=1}^{K} \zeta_k \, p(\mathbf{x}_{1:t}|\mathcal{M}_k)$. Thus, in this case, MAPF can be interpreted as a *unique* filter with *adaptive* prior pdf where, $\bar{\rho}_{k,t} \approx \zeta_k$. Furthermore, if all the models are the same equal to the true one $\mathcal{T}$, i.e.,

$$\mathcal{M}_1 = \mathcal{M}_2 = \ldots = \mathcal{M}_K = \mathcal{T},$$

then MAPF described a distributed PF scheme [7, 10, 13] which K parallel PFs cooperate for providing a global estimator,

$$\widehat{\mathbf{I}}_{1:t} = \sum_{k=1}^{K} \bar{\rho}_{k,t} \widetilde{\mathbf{I}}_{k,1:t}. \tag{45}$$

The computational effort is distributed in order to foster the filters which are providing the best performance, in the specific run.

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

5. MAPF FOR TIME-VARYING MODELS

The MAPF scheme can be easily modified for applying it in a switching 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 traveller 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 $\widehat{Z}_{k,t}$, for $k = 1, \dots, , K$. In this case we modify the computation of $\widehat{Z}_{k,t}$: the simplest possibility is to refresh all the $\widehat{Z}_{k,t}$'s each T_V iterations considering only the incremental weights (as forcing $\xi_{k,t-1} = \widehat{Z}_{k,t-1} = 1$)[10], i.e.,

$$\widehat{Z}_{k,t^*} = \frac{1}{M_{k,t^*}} \sum_{m=1}^{M_{k,t^*}} \lambda_{k,t^*}^{(m)}, \quad t^* = rT_V, \quad r \in \mathbb{N}. \tag{46}$$

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, each T_V iterations, the numbers of particles $M_{k,t}$'s should be refreshed, setting $M_{k,t} = \frac{N}{K}$ for all k. Moreover, in order to recover lost filters, a joint resampling should be performed, considering the weights $\bar{\gamma}_{k,t}^{(i_k)} = \bar{w}_{k,t}^{(i_k)} \bar{\rho}_{k,t}$, $i_k = 1, \dots, M_{k,t}$ and $k = 1, \dots, K$. 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),

$$\widehat{p}(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = \sum_{k=1}^{K} \sum_{i_k=1}^{M_{k,t}} \bar{\gamma}_{k,t}^{(i_k)} \delta(\mathbf{x}_{1:t} - \mathbf{x}_{k,1:t}^{(i_k)}), \tag{47}$$

where $\bar{\gamma}_{k,t}^{(i_k)}$ are in Eqs. (16)-(18). Figure 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.

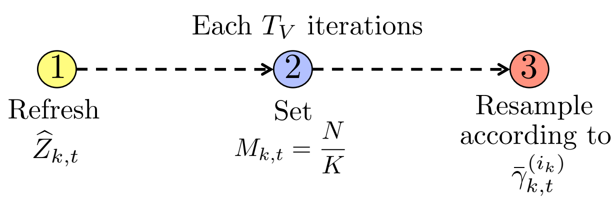


Fig. 3: Refreshing strategy for time-varying model settings.

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/`).

6. NUMERICAL SIMULATIONS

6.1. Online model selection for time series

As 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 recognize on-line which the

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

model generates the received measurements, $y_{1:T}$, between

$$\mathcal{M}_1 : \begin{cases} x_{1,t} = \frac{a x_{1,t}}{1+b x_{1,t}^2} + v_{1,t}, \\ y_t = x_{1,t} + u_{1,t}, \end{cases} \tag{48}$$

$$\mathcal{M}_2 : \begin{cases} x_{2,t} = x_{2,t-1} + v_{2,t}, \\ y_t = \exp\{-c\, x_{2,t}\} + u_{2,t}, \end{cases} \tag{49}$$

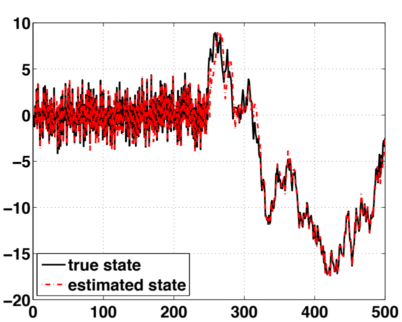
with, $a = -10$, $b = 3$, $c = 0.2$ and $t = 1, \ldots, T = 500$. The variables $v_{k,t} \sim \mathcal{N}(0,1)$ and $u_{k,t} \sim \mathcal{N}(0, \frac{1}{2})$, $k = 1, 2$ represent Gaussian perturbations.

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 have a change in the model at the iteration $t^* = 250$.

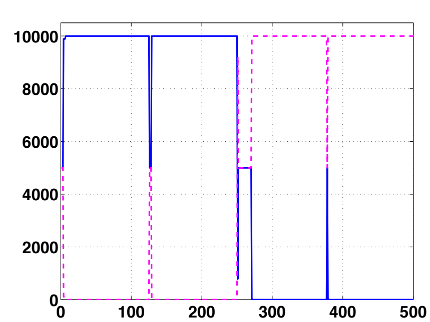
We apply MAPF with $N = 10^5$ total number of particles, $\epsilon = 0.1$ and refreshing window $T_V = 125$ (clearly, $K = 2$). Given the previous assumptions, we have averaged the results over 10^4 independent simulations, where the hidden states and the data are generated at each run, according to the model $\mathcal{T}_t$. Figure 4(a) shows the true and the estimated sequence of states in one specific run. Figure 4(b) depicts the evolution of the number of particles of each filters, $M_{k,t}$, $k = 1, 2$, as function of the the iterations t, in one particular run. Note that, the algorithm is able to detect quickly the true model and assigns adequately all the computational effort to the filter addressing the true model, at the corresponding iteration t. Figure 4(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.

PF true model	**MAPF**	**PF - $\mathcal{M}_1$**	**PF - $\mathcal{M}_2$**	**PF wrong model**
6.64	6.91	95.09	106.21	115.44

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 $N = 10^5$.



(a) True and estimated path in one run.



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

Fig. 4: (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.

6.2. Parameter selection

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

$$\mathcal{M}_k : \begin{cases} x_{k,t} = a_k |x_{k,t-1}| + v_{k,t} \\ y_t = b_k \log(x_{k,t}^2) + u_{k,t} \end{cases}, \qquad t = 1, \ldots, T. \tag{50}$$

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 scenarios:

• **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,

$$a_k = \frac{k}{K}, \qquad k = 1, \ldots, K. \tag{51}$$

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 $k = 1, \ldots, K-1$. Moreover, we have another grid for b_k,

$$b_k = \frac{1}{3} + \frac{10(k-1)}{K}, \qquad k = 1, \ldots, K-1, \tag{52}$$

and $b_K = 1$.
• **Common likelihood function (S2):** the parameters a_k are selected are in Eq. (51), 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 $k = 1, \ldots, K$. Namely, the models share the same likelihood function.
• **Common dynamic model (S3):** in this case, $a_k = 1$ and $\sigma_{1,k} = 1$ for all $k = 1, \ldots, K$, whereas the b_k's are set as in Eq. (52) (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 are $N = 10^5$, 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 "no-refreshing" setup) and number of considered models $K \in \{5, 20, 50, 100\}$. Table 4 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 highly frequent refreshing steps. Figures 5 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 an Mean Square Error (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, in particular to offer real-time location-based services such as advising the traveller when it is time to change bus, or suggesting any other kind of services.

Location is obtained by the devices GPS. 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 incorporated to the sensor network. A volunteer group of researchers collected GPS measurements recorded continuously during two weeks by using Android smartphones running the ContextLogger application[11]. 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 $\mathbf{y}_t = \mathbf{x}_t + \mathbf{r}_t$ where $\mathbf{x}_t$

[11] See `http://contextlogger.org/`.

Table 4: 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

K	5	20	50	100
S1	98.55	98.60	98.60	98.59
S2	98.48	97.53	97.49	97.26
S3	98.50	98.57	98.51	98.58

(b) With refreshing $T_V = 20$

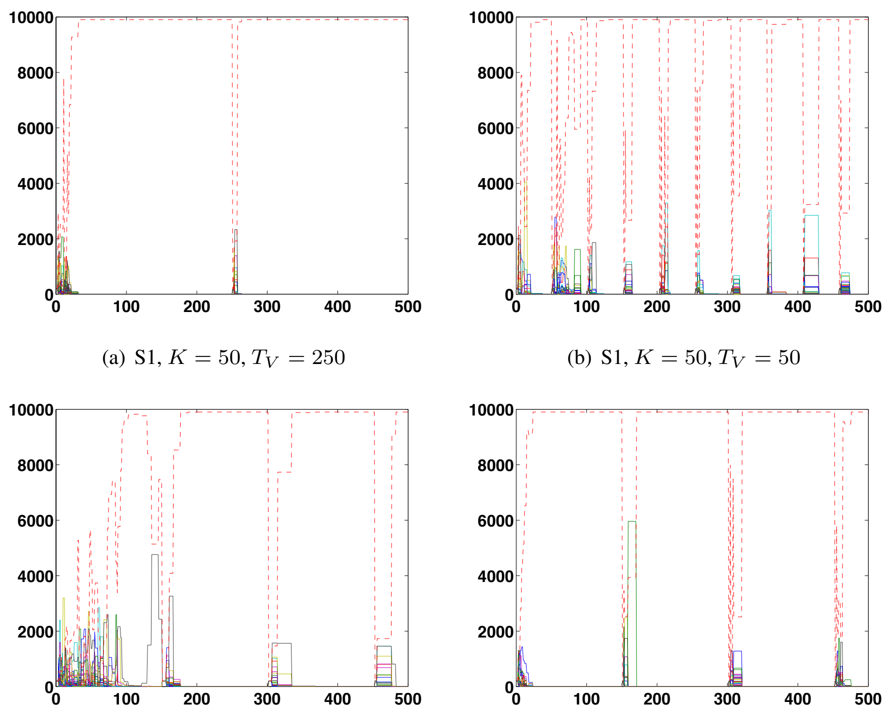
K	5	20	50	100
S1	62.54	91.12	91.80	91.68
S2	62.06	79.04	81.99	81.26
S3	64.54	91.04	91.90	92.40

(c) With refreshing $T_V = 40$

K	5	20	50	100
S1	71.81	94.78	95.00	95.50
S2	82.65	89.37	89.65	90.36
S3	82.78	94.78	95.42	95.36

(d) With refreshing $T_V = 100$

K	5	20	50	100
S1	78.47	97.98	98.08	98.02
S2	92.46	95.83	95.68	96.15
S3	94.46	97.76	98.06	97.93



(a) S1, $K = 50$, $T_V = 250$

(b) S1, $K = 50$, $T_V = 50$

(c) S2, $K = 50$, $T_V = 150$

(d) S3, $K = 50$, $T_V = 150$

Fig. 5: 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.

is the current position and $\mathbf{r}_t$ 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,[12] and, in the case of Helsinki which we have considered, from the urban planner.[13]. As an example, we plot all stops in the Helsinki region in Fig. 6. 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:

[12] See `http://www.openstreetmap.org`
[13] See `http://hsl.fi`

(a) (b) (c)

Fig. 6: **(a)** All bus stops in Helsinki region (left). **(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.

Dynamic Model for Train-Bus. Let us define the subset $\mathcal{S} \subseteq \mathbb{R}^2$, which is a piecewise linear approximation the route of the corresponding train or bus. An example is shown in Figure 7(a). We consider the following equation,

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{h}_t(\mathcal{S}), \tag{53}$$

where, $\mathbf{x}_t \in \mathbb{R}^2$, and $\mathbf{h}_t(\mathcal{S})$ is random Gaussian perturbation depending on the subset $\mathcal{S} \subseteq \mathbb{R}^2$. In this case, $\mathbf{x}_t \sim \mathcal{N}(\mathbf{x}_{t-1}, \mathbf{\Sigma}_t(\mathcal{S}))$, where the covariance matrix depends on $\mathcal{S}$: the slope of the eigenvector associated to the greatest eigenvalue λ_1 is the same of the current linear piece of $\mathcal{S}$ (see Figure 7(a)). Moreover, denoting as λ_2 the second eigenvalue, we design the covariance matrix $\mathbf{\Sigma}_t(\mathcal{S})$ in order to have $\lambda_1 >> \lambda_2$, 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 $\mathcal{S}$. Furthermore, the covariance matrix $\mathbf{\Sigma}_t(\mathcal{S})$ can also contain other kind of information about, for instance, 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 [24, 29] for discarded the samples outside the route $\mathcal{S}$ (in this case, the noise is a truncated Gaussian pdf, restricted within $\mathcal{S}$). 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.

Dynamic Model for Car, Cycling and Walk. In this case, we consider a simple model of type

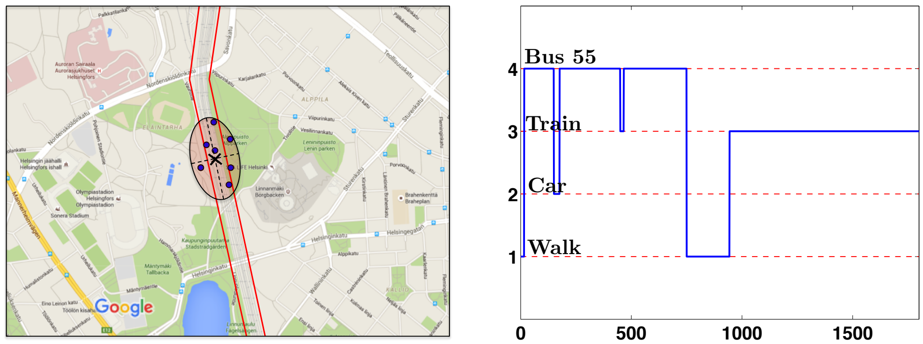
$$\mathbf{x}_t = \begin{cases} \mathbf{x}_{t-1} + b_1 \mathbf{v}_t, & \text{with prob. } \frac{1}{3}, \\ \mathbf{x}_{t-1} + b_2 \mathbf{v}_t, & \text{with prob. } \frac{1}{3}, \\ \mathbf{x}_{t-1} + b_3 \mathbf{v}_t, & \text{with prob. } \frac{1}{3}, \end{cases} \tag{54}$$

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 conditionated 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. Figure 7(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. Considering only buses and trains, we obtain an MSE of 1.17 meters.

7. CONCLUSIONS

We have designed an interacting parallel sequential Monte Carlo scheme for inference in state space models and on-line 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



(a) (b)

Fig. 7: **(a)** Example of generation of particles (shown as circles) using the dynamic model of type in Eq. (53). **(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).

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.

8. REFERENCES

[1] K. Achutegui, L. Martino, J. Rodas, C.J. Escudero, and J. Miguez. A multi-model particle filtering algorithm for indoor tracking of mobile terminals using rss data. In *IEEE Intelligent Control and Control Applications (CCA)*, pages 1702–1707, 2009.

[2] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society B*, 72(3):269–342, 2010.

[3] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić. Particle methods for change detection, system identification and control. *Proceedings of the IEEE*, 92(3):423–438, March 2004.

[4] M. S. Arulumpalam, S. Maskell, N. Gordon, and T. Klapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions Signal Processing*, 50(2):174–188, February 2002.

[5] L. Bazzani, D. Bloisi, and 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*, pages 47–54, Miami, Florida, 2009.

[6] Y Boers and JN Driessen. Interacting multiple model particle filter. *IEEE Proceedings-Radar, Sonar and Navigation*, 150(5):344–349, 2003.

[7] M. Bolić, P. M. Djurić, and S. Hong. Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions Signal Processing*, 53(7):2442–2450, July 2005.

[8] F. Caron, M. Davy, E. Duflos, and P. Vanheeghe. Particle filtering for multisensor data fusion with switching observation models: Application to land vehicle positioning. *Signal Processing, IEEE Transactions on*, 55(6):2703–2719, 2007.

[9] F. Caron, Manuel Davy, E. Duflos, and P. Vanheeghe. Particle filtering for multisensor data fusion with switching observation models: Application to land vehicle positioning. *IEEE Transactions on Signal Processing*, 55(6):2703–2719, June 2007.

[10] O. Demirel, I. Smal, W. Niessen, E. Meijering, and I. F. Sbalzarini. A parallel particle filtering library. *arXiv:1310.5045*, 2013.

[11] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, September 2003.

[12] P. M. Djurić and J. Míguez. Assessment of nonlinear dynamic models by Kolmogorov–Smirnov statistics. *IEEE Transactions on Signal Processing*, 58(10):5069–5079, 2010.

[13] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York (USA), 2001.

[14] V. Elvira, L. Martino, D. Luengo, and M. Bugallo. Efficient multiple importance sampling estimators. *IEEE Signal Processing Letters*, 22(10):1757–1761, 2015.

[15] V. Elvira, J. Míguez, and P. M. Djurić. Adapting the number of particles in sequential monte carlo methods through an online scheme for convergence assessment. *arXiv preprint arXiv:1509.04879*, 2015.

[16] W. J. Fitzgerald. Markov chain Monte Carlo methods with applications to signal processing. *Signal Processing*, 81(1):3–18, January 2001.

[17] D. Fox. Kld-sampling: Adaptive particle filters. In *Advances in neural information processing systems*, pages 713–720, 2001.

[18] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions Signal Processing*, 50(2):425–437, February 2002.

[19] J. A. Hoeting, D. Madigan, A. E. Raftery, and Chris T. Volinsky. Bayesian model averaging: a tutorial. *Statistical Science*, 14(4):382–417, 1999.

[20] J. Hollmén and V. Tresp. Call-based fraud detection in mobile communications networks using a hierarchical regime-switching model. In M. Kearns, S. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11: Proceedings of the 1998 Conference (NIPS'11)*, pages 889–895. MIT Press, 1999.

[21] M. Hong, M. F. Bugallo, and P. M. Djurić. Joint model selection and parameter estimation by population monte carlo simulation. *Selected Topics in Signal Processing, IEEE Journal of*, 4(3):526–539, 2010.

[22] D.S. Lee and N.K.K. Chia. A particle algorithm for sequential Bayesian parameter estimation and model selection. *IEEE Transactions on Signal Processing*, 50(2):326–336, 2002.

[23] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[24] L. Martino and J. Míguez. A generalization of the adaptive rejection sampling algorithm. *Statistics and Computing*, 21(4):633–647, July 2011.

[25] S. McGinnity and G. W. Irwin. Manoeuvring target tracking using a multiple-model bootstrap filter. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, chapter 23, pages 479–496. Springer, 2001.

[26] J. Míguez. Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization. *Digital Signal Processing*, 17:787–807, 2007.

[27] J. Míguez, M. F. Bugallo, and P. M. Djurić. A new class of particle filters for random dynamical systems with unknown statistics. *EURASIP Journal on Advances in Signal Processing*, 2004(15):2278–2294, November 2004.

[28] M. Morshidi and T. Tjahjadi. Gravity optimised particle filter for hand tracking. *Pattern Recognition*, 47(1):194 – 207, 2014.

[29] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[30] K. Salhi, M. Deaconu, A. Lejay, N. Champagnat, and N. Navet. Regime switching model for financial data: empirical risk analysis. 2015.

[31] Y. Salih and A. S. Malik. Comparison of stochastic filtering methods for 3d tracking. *Pattern Recognition*, 44(10-11):2711–2737, 2011.

[32] B. Shin, J. Tao, and R. Klette. A super-particle filter for lane detection. *Pattern Recognition*, 48(11):3333–3345, 2015.

[33] G. Storvik. Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions Signal Processing*, 50(2):281–289, February 2002.

[34] B. Tian, C. M. Chew, H. Tang, and M. Yuan. An intuitive and efficient switching particle filter for real-time vision-based localization. In *Cybernetics and Intelligent Systems (CIS), IEEE Conference on*, pages 7–11, Nov 2013.

[35] E. Veach and L. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. *In SIGGRAPH 1995 Proceedings*, pages 419–428, 1995.

[36] W. Wu, M. J. Black, D. Mumford, Y. Gao, E. Bienenstock, and J. P. Donoghue. Modeling and decoding motor cortical activity using a switching Kalman filter. *IEEE Trans. Biomedical Engineering*, 51(6):933–942, June 2004.

[37] N. Zorbas, D. Zissis, K. Tserpes, and 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.

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 $p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}, \mathcal{M}_1) = p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}|\mathcal{M}_1)$. Given the assumptions which we have consider for our model, i.e., $p(\mathbf{x}_t|\mathbf{x}_{1:t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $p(\mathbf{y}_t|\mathbf{x}_{1:t}, \mathbf{y}_{1:t-1}) = p(\mathbf{y}_t|\mathbf{x}_t)$, we can write easily the following recursive formula for the joint pdf $p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t})$ [11, 13],

$$p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = \ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1})\, p(\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}). \tag{55}$$

Given our assumptions, note that we can always evaluate the joint pdf $p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t})$. However, we are interested in a similar recursive expression which involves the posterior $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$. Thus, starting be the definition and replacing the expression above, we obtain

$$\begin{aligned} p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) &= \frac{p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t})}{p(\mathbf{y}_{1:t})} && (56)\\ &= \frac{\ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_{1:t})} p(\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}). && (57) \end{aligned}$$

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

$$p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = \ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{y}_{1:t})} p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}), \tag{58}$$

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

$$\begin{aligned} p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) &= \left[\ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1})\right] \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) p(\mathbf{y}_{1:t-1})} p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}), \\ &= \frac{\ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}). && (59) \end{aligned}$$

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

B. DERIVATION OF THE ESTIMATOR $\widetilde{Z}_T$

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., $\widehat{Z}$ in Eq. (27) and $\widetilde{Z}$ given in Eq. (28). The alternative formulation $\widetilde{Z}$ in Eq. (28) can be derived as follows [13]. Recall that

$$\begin{aligned} Z_t = p(\mathbf{y}_{1:t}) &= \int_{\mathcal{X}_{1:t}} p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) d\mathbf{x}_{1:t} \\ &\approx \widehat{Z}_t = \frac{1}{M} \sum_{m=1}^{M} w_t^{(m)}, && (60) \end{aligned}$$

and consider the following integral

$$\begin{aligned}
&\int_{\mathcal{X}_{1:t}} \ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{1:t} = \\
&= \frac{1}{p(\mathbf{y}_{1:t-1})} \int_{\mathcal{X}_{1:t}} \ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}) d\mathbf{x}_{1:t}, \\
&= \frac{1}{p(\mathbf{y}_{1:t-1})} \int_{\mathcal{X}_{1:t}} p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) d\mathbf{x}_{1:t}, \\
&= \frac{p(\mathbf{y}_{1:t})}{p(\mathbf{y}_{1:t-1})} = \frac{Z_t}{Z_{t-1}} = p(\mathbf{y}_t|\mathbf{y}_{1:t-1}).
\end{aligned} \tag{61}$$

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

$$\int_{\mathcal{X}_{1:t}} \ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{1:t} = \frac{Z_t}{Z_{t-1}}. \tag{62}$$

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

$$\begin{aligned}
\int_{\mathcal{X}_{1:t}} \ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) \widehat{p}(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{1:t} &= \sum_{m=1}^{M} \bar{w}_{t-1}^{(m)} \int_{\mathcal{X}_t} \ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}) d\mathbf{x}_t, \\
&\approx \frac{Z_t}{Z_{t-1}}.
\end{aligned} \tag{63}$$

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,

$$\begin{aligned}
\int_{\mathcal{X}_{1:t}} \ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) \widehat{p}(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{1:t} &= \sum_{m=1}^{M} \bar{w}_{t-1}^{(m)} \int_{\mathcal{X}_t} \ell_t(\mathbf{y}_t|\mathbf{x}_t) q_t(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}) d\mathbf{x}_t, \\
&\approx \sum_{m=1}^{M} \bar{w}_{t-1}^{(m)} \frac{\ell_t(\mathbf{y}_t|\mathbf{x}_t^{(m)}) q_t(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{\phi_t(\mathbf{x}_t^{(m)}|\mathbf{x}_{1:t-1}^{(m)})}, \\
&= \sum_{m=1}^{M} \bar{w}_{t-1}^{(m)} \lambda_t^{(m)} \approx \frac{Z_t}{Z_{t-1}}.
\end{aligned} \tag{64}$$

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

$$\begin{aligned}
\sum_{m=1}^{M} \bar{w}_{t-1}^{(m)} \lambda_t^{(m)} &= \frac{1}{\sum_{m=1}^{M} w_{t-1}^{(m)}} \sum_{m=1}^{M} w_{t-1}^{(m)} \lambda_t^{(m)}, \\
&= \frac{1}{\sum_{m=1}^{M} w_{t-1}^{(m)}} \sum_{m=1}^{M} w_t^{(m)}, \\
&= \frac{\frac{1}{M}\sum_{m=1}^{M} w_t^{(m)}}{\frac{1}{M}\sum_{m=1}^{M} w_{t-1}^{(m)}} = \frac{\widehat{Z}_t}{\widehat{Z}_{t-1}} \approx \frac{Z_t}{Z_{t-1}}.
\end{aligned} \tag{65}$$

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

$$\begin{aligned}
\widetilde{Z}_t &= \prod_{\tau=1}^{t} \left[\sum_{m=1}^{M} \bar{w}_{\tau-1}^{(m)} \lambda_\tau^{(m)} \right] \\
&= \prod_{\tau=1}^{T} \frac{\widehat{Z}_\tau}{\widehat{Z}_{\tau-1}} = \widehat{Z}_1 \frac{\widehat{Z}_2}{\widehat{Z}_1} \cdots \frac{\widehat{Z}_{t-1}}{\widehat{Z}_{t-2}} \frac{\widehat{Z}_t}{\widehat{Z}_{t-1}} \\
&= \widehat{Z}_t \approx Z,
\end{aligned} \tag{66}$$

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. (63) via importance sampling. In this case, we assume to draw independent samples $\mathbf{x}_t^{(1)}$,. . ., $\mathbf{x}_t^{(M)}$ from the a different proposal pdf $\varphi(\mathbf{x}_{1:t})$, defined as

$$\begin{aligned}\varphi(\mathbf{x}_{1:t}) &= \phi_t(\mathbf{x}_t|\mathbf{x}_{1:t-1})\widehat{p}(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}),\\ &= \sum_{m=1}^{M}\bar{w}_{t-1}^{(m)}\phi_t(\mathbf{x}_t|\mathbf{x}_{1:t-1}^{(m)}).\end{aligned}$$

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

$$\begin{aligned}\int_{\mathcal{X}_{1:t}}\ell_t(\mathbf{y}_t|\mathbf{x}_t)q_t(\mathbf{x}_t|\mathbf{x}_{t-1})\widehat{p}(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{1:t} &\approx \frac{1}{M}\sum_{m=1}^{M}\frac{\ell_t(\mathbf{y}_t|\mathbf{x}_t^{(m)})q_t(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{\phi_t(\mathbf{x}_t^{(m)}|\mathbf{x}_{1:t-1}^{(m)})}\\ &= \frac{1}{M}\sum_{m=1}^{M}\lambda_t^{(m)} \approx \frac{Z_t}{Z_{t-1}}. \qquad (67)\end{aligned}$$

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

$$\xi_t^{(m)} = \begin{cases} w_t^{(m)}, & \text{without resampling at the } t\text{-th iteration,}\\ \widehat{Z}_t, & \text{after resampling at the } t\text{-th iteration,}\end{cases} \qquad (68)$$

we can ensure that

$$\widehat{Z}_t = \sum_{m=1}^{M}\xi_t^{(m)}\lambda_t^{(m)},$$

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 $\xi_{t-1}^{(m)} = \widehat{Z}_{t-1}$, then $\widehat{Z}_t = \frac{\widehat{Z}_{t-1}}{M}\sum_{m=1}^{M}\lambda_t^{(m)}$. Since $\widehat{\frac{Z_t}{Z_{t-1}}} = \frac{1}{M}\sum_{m=1}^{M}\lambda_t^{(m)}$, we obtain

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

Finally, note that $\widehat{Z}_t$ and $\widetilde{Z}_t = \prod_{t=1}^{T}\left[\sum_{m=1}^{M}\bar{\xi}_{t-1}^{(m)}\lambda_t^{(m)}\right]$ 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 $\xi_t^{(m)}$.

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 $\widehat{Z}_{k,t}$ of the marginal likelihood. Then, this information is properly merged following the approximation of $\widehat{p}(\mathbf{x}|\mathbf{y})$ 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) [14]. An important related observation is remarked below.

Remark 4. *Consider that, at the t-th iteration, the condition in Eq.* (39) *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)*,*

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

using the so-called deterministic mixture (DM) procedure [35, 14]. 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 $k = 1, \ldots, K$.
2. For $n = 1, \ldots, N$:
   (a) Select a model k with probability $\bar{\rho}_{k,t}$, $k = 1, \ldots, K$.
   (b) Set $i_k = i_k + 1$ resample $\bar{\mathbf{x}}_{k,t}^{(i_k)} = \mathbf{x}_{k,t}^{(j)} \in \{\mathbf{x}_{k,t}^{(1)}, \ldots, \mathbf{x}_{k,t}^{(M_{k,t})}\}$ with probability $\bar{w}_{k,t}^{(j)}$, $j = 1, \ldots, M_{k,t}$.
3. Set $M_{k,t+1} = i_k$, for all $k = 1, \ldots, K$, 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.