A Review of Multiple Try MCMC algorithms for Signal Processing

Luca Martino

Image Processing Lab., Universitat de València (Spain)

Universidad Carlos III de Madrid, Leganes (Spain)

Abstract

Many applications in signal processing require the estimation of some parameters of interest given a set of observed data. More specifically, Bayesian inference needs the computation of *a-posteriori* estimators which are often expressed as complicated multi-dimensional integrals. Unfortunately, analytical expressions for these estimators cannot be found in most real-world applications, and Monte Carlo methods are the only feasible approach. A very powerful class of Monte Carlo techniques is formed by the Markov Chain Monte Carlo (MCMC) algorithms. They generate a Markov chain such that its stationary distribution coincides with the target posterior density. In this work, we perform a thorough review of MCMC methods using multiple candidates in order to select the next state of the chain, at each iteration. With respect to the classical Metropolis-Hastings method, the use of multiple try techniques foster the exploration of the sample space. We present different Multiple Try Metropolis schemes, Ensemble MCMC methods, Particle Metropolis-Hastings algorithms and the Delayed Rejection Metropolis technique. We highlight limitations, benefits, connections and differences among the different methods, and compare them by numerical simulations.

Keywords: Markov Chain Monte Carlo, Multiple Try Metropolis, Particle Metropolis-Hastings, Particle Filtering, Monte Carlo methods, Bayesian inference

1 Introduction

Bayesian methods have become very popular in signal processing over the last years [1, 2, 3, 4]. They require the application of sophisticated Monte Carlo techniques, such as Markov chain Monte Carlo (MCMC) and particle filters, for the efficient computation of *a-posteriori* estimators [5, 6, 2]. More specifically, the MCMC algorithms generate a Markov chain such that its stationary distribution coincides with the posterior probability density function (pdf) [7, 8, 9]. Typically, the only requirement is to be able to evaluate the target function, where the knowledge of the normalizing constant is usually not needed.

The most popular MCMC method is undoubtedly the Metropolis-Hastings (MH) algorithm [10, 11]. The MH technique is a very simple method, easy to be applied: this is the reason of

its success. In MH, at each iteration, one new candidate is generated from a proposal pdf and then is properly compared with the previous state of the chain, in order to decide the next state. However, the performance of MH are often not satisfactory. For instance, when the posterior is multimodal, or when the dimension of the space increases, the correlation among the generated samples is usually high and, as a consequence, the variance of the resulting estimators grows. To speed up the convergence and reduce the "burn-in" period of the MH chain, several extensions have been proposed in literature.

In this work, we provide an exhaustive review of more sophisticated MCMC methods that, at each iteration, consider different candidates as possible new state of the chain. More specifically, at each iteration different samples are compared by certain weights and then one of them is selected as possible future state. The main advantage of these algorithms is that they foster the exploration of a larger portion of the sample space, decreasing the correlation among the states of the generated chain. In this work, we describe different algorithms of this family, independently introduced in literature. The main contribution is to present them under the same the framework and notation, remarking differences, relationships, limitations and strengths. All the discussed techniques yield an ergodic chain converging to the posterior density of interest (in the following, referred also as target pdf).

The first scheme of this MCMC class, called Orientational Bias Monte Carlo (OBMC) [12, Chapter 13], was proposed in the context of molecular simulation. Later on a more general algorithm, called Multiple Try Metropolis (MTM), was introduced [13].[1] The MTM algorithm has been extensively studied and generalized in different ways [14, 15, 16, 17, 18]. Other techniques, alternative to the MTM schemes, are the so-called the Ensemble MCMC (EnMCMC) methods [19, 20, 21, 22]. They follow a similar approach to MTM but employ a different acceptance function for selecting the next state of the chain. With respect to (w.r.t.) a generic MTM scheme, EnMCMC does not require any generation of auxiliary samples (as in a MTM scheme employing a generic proposal pdf) and hence, in this sense, EnMCMC are less costly.

In all the previous techniques, the candidates are drawn in a batch way and compared jointly. In the Delayed Rejection Metropolis (DRM) algorithm [23, 24, 25], in case of rejection of the novel possible state, the authors suggest to perform an additional acceptance test considering a new candidate. If this candidate is again rejected, the procedure can be iterated until reaching a desired number of attempts. The main benefit of DRM is that the proposal pdf can be improved at each intermediate stage. However, the acceptance function progressively becomes more complex so that the implementation of DRM for a great number of attempts is not straightforward (compared to the implementation of a MTM scheme with a generic number of tries).

In the last years, other Monte Carlo methods which combine particle filtering and MCMC have become very popular in the signal processing community. For instance, this is the case of the Particle Metropolis Hastings (PMH) and the Particle Marginal Metropolis Hastings (PMMH) algorithms, which have been widely used in signal processing in order to make inference and smoothing about dynamical and static parameters in state space models [26, 27]. PMH can be interpreted as a MTM scheme where the different candidates are generated and weighted by the use of a particle filter [28, 29]. In this work, we present PMH and PMMH and discuss

---

[1]MTM includes OBMC as a special case (see Section 4.1.1).

their connections and differences with the classical MTM approach. Furthermore, we describe a suitable procedure for recycling some candidates in the final Monte Carlo estimators, called Group Metropolis Sampling (GMS) [29, 30]. The GMS scheme can be also seen as a way of generating a chain of sets of weighted samples. Finally, note that other similar and related techniques can be found within the so-called data augmentation approach [31, 32].

The remaining of the paper is organized as follows. Section 2 recalls the problem statement and some background material, introducing also the required notation. The basis of MCMC and the Metropolis-Hastings (MH) algorithm are presented in Section 3. Section 4 is the core of the work, which describes the different MCMC using multiple candidates. Section 6 provides some numerical results, applying different techniques in a hyperparameter tuning problem for a Gaussian Process regression model, and in a localization problem considering a wireless sensor network. Some conclusions are given in Section 7.

2   Problem statement and preliminaries

In many signal processing applications, the goal consists in inferring a variable of interest, $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_D] \in \mathcal{D} \subseteq \mathbb{R}^D$, given a set of observations or measurements, $\mathbf{y} \in \mathbb{R}^P$. In the Bayesian framework, the total knowledge about the parameters, after the data have been observed, is represented by the posterior probability density function (pdf) [8, 9], i.e.,

$$\begin{aligned} \bar{\pi}(\boldsymbol{\theta}) &= p(\boldsymbol{\theta}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})}{Z(\mathbf{y})}, \\ &= \frac{1}{Z}\pi(\boldsymbol{\theta}), \end{aligned} \tag{1}$$

where $\ell(\mathbf{y}|\boldsymbol{\theta})$ denotes the likelihood function (i.e., the observation model), $g(\boldsymbol{\theta})$ is the prior probability density function (pdf) and $Z = Z(\mathbf{y})$ is the marginal likelihood (a.k.a., Bayesian evidence) [33, 34] and $\pi(\boldsymbol{\theta}) = \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})$. In general Z is unknown, and it is possible to evaluate only the unnormalized target function, $\pi(\boldsymbol{\theta}) \propto \bar{\pi}(\boldsymbol{\theta})$.

The analytical study of the posterior density $\bar{\pi}(\boldsymbol{\theta})$ is often unfeasible and integrals involving $\bar{\pi}(\boldsymbol{\theta})$ are typically intractable [33, 35, 34]. For instance, one might be interested in the estimation of

$$I = E_\pi[f(\boldsymbol{\theta})] = \int_{\mathcal{D}} f(\boldsymbol{\theta})\bar{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta}, \tag{2}$$

$$= \frac{1}{Z}\int_{\mathcal{D}} f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}, \tag{3}$$

where $f(\boldsymbol{\theta})$ is a generic integrable function w.r.t. $\bar{\pi}$.

Dynamic and static parameters. In some specific application, the variable of interest $\boldsymbol{\theta}$ can be split in two disjoint parts, $\boldsymbol{\theta} = [\mathbf{x}, \boldsymbol{\lambda}]$, where one, $\mathbf{x}$, is involved into a dynamical system (for instance, $\mathbf{x}$ is the hidden state in a state-space model) and the other, $\boldsymbol{\lambda}$, is a static parameter (for instance, an unknown parameter of the model). The strategies for making inference about $\mathbf{x}$ and $\boldsymbol{\lambda}$ should take into account the different nature of the two parameters (e.g., see Section 4.2.2). The main notation and acronyms are summarized in Tables 1-2.

Table 1: Summary of the main notation.

$\boldsymbol{\theta} = [\theta_1, \ldots, \theta_D]^\top$	Variable of interest, $\boldsymbol{\theta} \in \mathcal{D} \subseteq \mathbb{R}^D$.
$\mathbf{y}$	Observed measurements/data.
$\boldsymbol{\theta} = [\mathbf{x}, \boldsymbol{\lambda}]^\top$	$\mathbf{x}$: dynamic parameters; $\boldsymbol{\lambda}$: static parameters.
$\bar{\pi}(\boldsymbol{\theta})$	Normalized posterior pdf, $\bar{\pi}(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})$.
$\pi(\boldsymbol{\theta})$	Unnormalized posterior function, $\pi(\boldsymbol{\theta}) \propto \bar{\pi}(\boldsymbol{\theta})$.
$\widehat{\pi}(\boldsymbol{\theta})$	Particle approximation of $\bar{\pi}(\boldsymbol{\theta})$.
I	Integral of interest, in Eq. (2).
$\widehat{I}, \widetilde{I}$	Estimators of I.
Z	Marginal likelihood; normalizing constant of $\pi(\boldsymbol{\theta})$.
$\widehat{Z}, \widetilde{Z}$	Estimators of the marginal likelihood Z.

Table 2: Summary of the main acronyms.

MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
I-MH	Independent Metropolis-Hastings
MTM	Multiple Try Metropolis
I-MTM	Independent Multiple Try Metropolis
I-MTM2	Independent Multiple Try Metropolis (version 2)
PMH	Particle Metropolis-Hastings
PMMH	Particle Marginal Metropolis-Hastings
GMS	Group Metropolis Sampling
EnMCMC	Ensemble MCMC
I-EnMCMC	Independent Ensemble MCMC
DRM	Delayed Rejection Metropolis
IS	Importance Sampling
SIS	Sequential Importance Sampling
SIR	Sequential Importance Resampling

2.1   Monte Carlo integration

In many practical scenarios, the integral I cannot be computed in a closed form, and numerical approximations are typically required. Many deterministic quadrature methods are available in the literature [36, 37]. However, as the dimension D of the inference problem grows ($\boldsymbol{\theta} \in \mathbb{R}^D$), the deterministic quadrature schemes become less efficient. In this case, a common approach consists of approximating the integral I in Eq. (2) by using Monte Carlo (MC) quadrature [8, 9]. Namely, considering T independent and identically distributed (i.i.d.) samples drawn from the posterior

target pdf, i.e. $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T \sim \bar{\pi}(\boldsymbol{\theta}),^2$ we can build the consistent estimator

$$\widehat{I}_T = \frac{1}{T} \sum_{t=1}^{T} f(\boldsymbol{\theta}_t). \tag{4}$$

$\widehat{I}_T$ converges in probability to I due to the weak law of large numbers. The approximation above $\widehat{I}_T$ is known as a direct (or ideal) Monte Carlo estimator if the samples $\boldsymbol{\theta}_t$ are independent and identically distributed (i.i.d.) from $\bar{\pi}$. Unfortunately, in many practical applications, direct methods for drawing independent samples from $\bar{\pi}(\boldsymbol{\theta})$ are not available. Therefore, different approaches are required, such as the Markov chain Monte Carlo (MCMC) techniques.

3 Markov chain Monte Carlo (MCMC) methods

A MCMC algorithm generates an ergodic Markov chain with invariant (a.k.a., stationary) density given by the posterior pdf $\bar{\pi}(\boldsymbol{\theta})$ [7, 9]. Specifically, given a starting state $\boldsymbol{\theta}_0$, a sequence of *correlated* samples is generated, $\boldsymbol{\theta}_0 \to \boldsymbol{\theta}_1 \to \boldsymbol{\theta}_2 \to \ldots \to \boldsymbol{\theta}_T$. Even if the samples are now correlated, the estimator

$$\widetilde{I}_T = \frac{1}{T} \sum_{t=1}^{T} f(\boldsymbol{\theta}_t) \tag{5}$$

is consistent, regardless the starting vector $\boldsymbol{\theta}^{(0)}$ [9].[3] With respect to the direct Monte Carlo approach using i.i.d. samples, the application of an MCMC algorithm entails a loss of efficiency of the estimator $\widetilde{I}_T$, since the samples are *positively* correlated, in general. In other words, to achieve a given variance obtained with the direct Monte Carlo estimator, it is necessary to generate more samples. Thus, in order to improve the performance of an MCMC technique we have to decrease the correlation among the states of the chain.[4]

3.1 The Metropolis-Hastings (MH) algorithm

One of the most popular and widely applied MCMC algorithm is the Metropolis-Hastings (MH) method [11, 7, 9]. Recall that we are able to evaluate point-wise a function proportional to the target, i.e., $\pi(\boldsymbol{\theta}) \propto \bar{\pi}(\boldsymbol{\theta})$. A proposal density (a pdf which is easy to draw from) is denoted as $q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) > 0$, with $\boldsymbol{\theta}, \boldsymbol{\theta}_{t-1} \in \mathbb{R}^D$. In Table 3, we describe the standard MH algorithm in detail.

The algorithm returns the sequence of states $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_t, \ldots, \boldsymbol{\theta}_T\}$ (or a subset of them removing the burn-in period if an estimation of its length is available). We can see that the next state $\boldsymbol{\theta}_t$ can be the proposed sample $\boldsymbol{\theta}'$ (with probability α) or the previous state $\boldsymbol{\theta}_{t-1}$ (with probability $1 - \alpha$). Under some mild regularity conditions, when t grows, the pdf of the current state $\boldsymbol{\theta}_t$ converges to the target density $\bar{\pi}(\boldsymbol{\theta})$ [9]. The MH algorithm satisfies the so-called detailed

---

[2]In this work, for simplicity, we use the same notation for denoting a random variable or one realization of a random variable.

[3]Recall we are assuming that the Markov chain is ergodic and hence the starting value is forgotten.

[4]For the sake of simplicity, we use all the generated states in the final estimators, without removing any burn-in period [9].

Table 3: **The MH algorithm**

---

1. `Initialization`: Choose an initial state $\boldsymbol{\theta}_0$.

2. `FOR` $t = 1, \ldots, T$:

   (a) Draw a sample $\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$.

   (b) Accept the new state, $\boldsymbol{\theta}_t = \boldsymbol{\theta}'$, with probability

   $$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}') = \min\left[1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}_{t-1})q(\boldsymbol{\theta}'|\boldsymbol{\theta}_{t-1})}\right], \tag{6}$$

   Otherwise, set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$.

3. `Return`: $\{\boldsymbol{\theta}_t\}_{t=1}^T$.

---

balance condition which is sufficient to guarantee that the output chain is ergodic and has $\bar{\pi}$ as stationary distribution [7, 9]. Note that the acceptance probability α can be rewritten as

$$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}') = \min\left[1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}_{t-1})q(\boldsymbol{\theta}'|\boldsymbol{\theta}_{t-1})}\right] = \min\left[1, \frac{w(\boldsymbol{\theta}'|\boldsymbol{\theta}_{t-1})}{w(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}')}\right]. \tag{7}$$

where we have denoted $w(\boldsymbol{\theta}'|\boldsymbol{\theta}_{t-1}) = \frac{\pi(\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta}_{t-1})}$ and $w(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}') = \frac{\pi(\boldsymbol{\theta}_{t-1})}{q(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}')}$ in a similar fashion of the importance sampling weights of $\boldsymbol{\theta}'$ and $\boldsymbol{\theta}_{t-1}$ [9]. If the proposal pdf is independent from the previous state, i.e., $q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = q(\boldsymbol{\theta})$, the acceptance function depends on the ratio of the importance weights $w(\boldsymbol{\theta}') = \frac{\pi(\boldsymbol{\theta}')}{q(\boldsymbol{\theta}')}$ and $w(\boldsymbol{\theta}_{t-1}) = \frac{\pi(\boldsymbol{\theta}_{t-1})}{q(\boldsymbol{\theta}_{t-1})}$, as shown in Table 4. We refer to this special MH case as the Independent MH (I-MH) algorithm. It is strictly related to other techniques described in the following (e.g., see Section 4.2.1).

4   MCMC using multiple candidates

In the standard MH technique described above, at each iteration one new sample $\boldsymbol{\theta}'$ is generated to be tested with the previous state $\boldsymbol{\theta}_{t-1}$ by the acceptance probability $\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}')$. Other generalized MH schemes generate several candidates at each iteration to be tested as new possible state. In all these schemes, an extended acceptance probability α is properly designed in order to guarantee the ergodicity of the chain. Figure 1 provides a graphical representation of the difference between MH and the techniques using several candidates.

Below, we describe the most important examples of this class of MCMC algorithms. In most of them, a single MH-type test is performed at each iteration whereas in other methods a sequence of tests is employed. Furthermore, most of these techniques use an Importance Sampling (IS) approximation of the target density [8, 9] in order to improve the proposal procedure employed

Table 4: **The Independent MH (I-MH) algorithm**

1. `Initialization:` Choose an initial state $\boldsymbol{\theta}_0$.

2. `FOR` $t = 1, \ldots, T$:

   (a) Draw a sample $\boldsymbol{\theta}' \sim q(\boldsymbol{\theta})$.

   (b) Accept the new state, $\boldsymbol{\theta}_t = \boldsymbol{\theta}'$, with probability

   $$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}') \;=\; \min\left[1, \frac{w(\boldsymbol{\theta}')}{w(\boldsymbol{\theta}_{t-1})}\right], \tag{8}$$

   Otherwise, set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$.

3. `Return:` $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$.

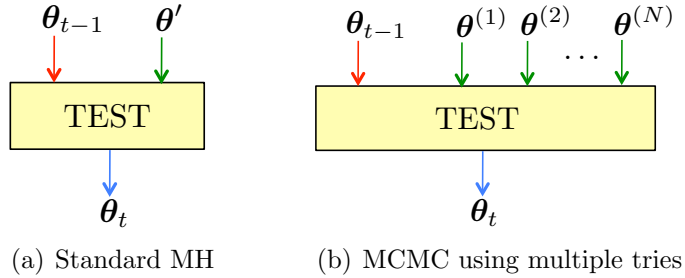(a) Standard MH   (b) MCMC using multiple tries

Figure 1: Graphical representation of the classical MH method and the MCMC schemes using different candidates at each iteration.

within a MH-type algorithm. Namely, they build an IS approximation, and then draw one sample from this approximation (resampling step). Finally, the selected sample is compared with the previous state of the chain, $\boldsymbol{\theta}_{t-1}$, according to a suitable generalized acceptance probability α. It can be proved that all the methodologies presented in this work yield a ergodic chain with the posterior $\bar{\pi}$ as invariant density.

4.1 The Multiple Try Metropolis (MTM) algorithm

The Multiple Try Metropolis (MTM) algorithms are examples of this class of methods, where N samples $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(N)}$ (called also "tries" or "candidates") are drawn from the proposal pdf $q(\boldsymbol{\theta})$, at each iteration [13, 14, 15, 16, 17, 38, 39]. Then, one of them is selected according to some suitable weights. Finally, the selected candidate is accepted or rejected as new state according to a generalized probability function α.

The MTM algorithm is given in Table 5. For the sake of simplicity, we have considered the use of the importance weights $w(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = \frac{\pi(\boldsymbol{\theta})}{q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})}$, but there is not a unique possibility, as also

shown below [13, 38]. In its general form, when the proposal depends on the previous state of the chain $q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$, the MTM requires the generation of $N-1$ auxiliary samples, $\mathbf{v}^{(i)}$, which are employed in the computation of the acceptance function α. They are needed in order to guarantee the ergodicity. Indeed, the resulting MTM kernel satisfies the detailed balance condition, so that the chain is reversible [13, 38]. Note that for $N = 1$, we have $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{(1)}$, $\mathbf{v}^{(1)} = \boldsymbol{\theta}_{t-1}$ and the acceptance probability of the MTM method becomes

$$
\begin{aligned}
\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(1)}) &= \min\left(1, \frac{w(\boldsymbol{\theta}^{(1)}|\boldsymbol{\theta}_{t-1})}{w(\mathbf{v}^{(1)}|\boldsymbol{\theta}^{(1)})}\right), \\
&= \min\left(1, \frac{w(\boldsymbol{\theta}^{(1)}|\boldsymbol{\theta}_{t-1})}{w(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}^{(1)})}\right) = \min\left(1, \frac{\pi(\boldsymbol{\theta}^{(1)})q(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}^{(1)})}{\pi(\boldsymbol{\theta}_{t-1})q(\boldsymbol{\theta}^{(1)}|\boldsymbol{\theta}_{t-1})}\right),
\end{aligned}
\tag{9}
$$

that is the acceptance probability of the classical MH technique. Several variants have been studied, for instance, with correlated tries and considering the use of different proposal pdfs [38, 40].

Remark 1. *The MTM method in Table 5 needs at step 2d the generation of $N-1$ auxiliary samples and at step 2e the computation of their weights (and, as a consequence, $N-1$ additional evaluation of the target pdf are required), that are only employed in the computation of the acceptance function α.*

4.1.1 Generic form of the weights

The importance weights are not the unique possible choice. It is possible to show that the MTM algorithm generates an ergodic chain with invariant density $\bar{\pi}$, if the weight function $w(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$ is chosen with the form

$$
w(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = \pi(\boldsymbol{\theta})q(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta})\xi(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}),
\tag{13}
$$

where

$$
\xi(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}) = \xi(\boldsymbol{\theta}, \boldsymbol{\theta}_{t-1}), \qquad \forall \boldsymbol{\theta}, \boldsymbol{\theta}_{t-1} \in \mathcal{D}.
$$

For instance, choosing $\xi(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}) = \frac{1}{q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})q(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta})}$, we obtain the importance weights $w(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = \frac{\pi(\boldsymbol{\theta})}{q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})}$ used above. If we set $\xi(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}) = 1$, we have $w(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = \pi(\boldsymbol{\theta})q(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta})$. Another interesting example can be employed if the proposal is symmetric, i.e., $q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = q(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta})$. In this case, we can choose $\xi(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}) = \frac{1}{q(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta})}$ and then $w(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = w(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})$, i.e., the weights only depend on the value of the target density at $\boldsymbol{\theta}$. Thus, MTM contains the Orientational Bias Monte Carlo (OBMC) scheme [12, Chapter 13] as a special case, when a symmetric proposal pdf is employed, and then one candidate is chosen with weights proportional to the target density, i.e., $w(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = \pi(\boldsymbol{\theta})$.

4.1.2 Independent Multiple Try Metropolis (I-MTM) schemes

The MTM method described in Table 5 requires to draw $2N - 1$ samples at each iteration (N candidates and $N - 1$ auxiliary samples) and $N - 1$ are only used in the acceptance probability function. The generation of the auxiliary points

$$
\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(j-1)}, \mathbf{v}^{(j+1)}, \ldots, \mathbf{v}^{(N)} \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(j)}),
$$

Table 5: **The MTM algorithm with importance sampling weights.**

1. `Initialization`: Choose an initial state $\boldsymbol{\theta}_0$.

2. `FOR` $t = 1, \ldots, T$:

   (a) Draw $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(N)} \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$.

   (b) Compute the importance weights

   $$w(\boldsymbol{\theta}^{(n)}|\boldsymbol{\theta}_{t-1}) = \frac{\pi(\boldsymbol{\theta}^{(n)})}{q(\boldsymbol{\theta}^{(n)}|\boldsymbol{\theta}_{t-1})}, \quad \text{with} \quad n = 1, \ldots, N. \tag{10}$$

   (c) Select one sample $\boldsymbol{\theta}^{(j)} \in \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}\}$, according to the probability mass function $\bar{w}_n = \frac{w(\boldsymbol{\theta}^{(n)}|\boldsymbol{\theta}_{t-1})}{\sum_{i=1}^{N} w(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}_{t-1})}$.

   (d) Draw $N-1$ auxiliary samples from $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(j)})$, denoted as $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(j-1)}, \mathbf{v}^{(j+1)}, \ldots, \mathbf{v}^{(N)} \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(j)})$, and set $\mathbf{v}^{(j)} = \boldsymbol{\theta}_{t-1}$.

   (e) Compute the weights of the auxiliary samples

   $$w(\mathbf{v}^{(n)}|\boldsymbol{\theta}^{(j)}) = \frac{\pi(\mathbf{v}^{(n)})}{q(\mathbf{v}^{(n)}|\boldsymbol{\theta}^{(j)})}, \quad \text{with} \quad n = 1, \ldots, N. \tag{11}$$

   (f) Set $\boldsymbol{\theta}_t = \boldsymbol{\theta}^{(j)}$ with probability

   $$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(j)}) = \min\left(1, \frac{\sum_{n=1}^{N} w(\boldsymbol{\theta}^{(n)}|\boldsymbol{\theta}_{t-1})}{\sum_{n=1}^{N} w(\mathbf{v}^{(n)}|\boldsymbol{\theta}^{(j)})}\right), \tag{12}$$

   otherwise, set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$.

3. `Return`: $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$.

can be avoided if the proposal pdf is independent from the previous state, i.e., $q(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}) = q(\boldsymbol{\theta})$. Indeed, in this case, we should draw $N-1$ samples again from $q(\boldsymbol{\theta})$ at the step 2d of Table 5. Since we have already drawn N samples from $q(\boldsymbol{\theta})$ at step 2a of Table 5, we can set

$$\mathbf{v}^{(1)} = \boldsymbol{\theta}^{(1)}, \ldots, \mathbf{v}^{(j-1)} = \boldsymbol{\theta}^{(j-1)}, \mathbf{v}^{(j)} = \boldsymbol{\theta}^{(j+1)} \ldots \mathbf{v}^{(N-1)} = \boldsymbol{\theta}^{(N)}, \tag{14}$$

without jeopardizing the ergodicity of the chain (recall that $\mathbf{v}^{(j)} = \boldsymbol{\theta}_{t-1}$). Hence, we can avoid step 2d and the acceptance function can be rewritten as

$$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(j)}) = \min\left(1, \frac{w(\boldsymbol{\theta}^{(j)}) + \sum_{n=1, n \neq j}^{N} w(\boldsymbol{\theta}^{(n)})}{w(\boldsymbol{\theta}_{t-1}) + \sum_{n=1, n \neq j}^{N} w(\boldsymbol{\theta}^{(n)})}\right). \tag{15}$$

The I-MTM algorithm is provided in Table 6.

Remark 2. *An I-MTM method requires only N new evaluations of the target pdf at each iteration, instead of $2N - 1$ new evaluations in the generic MTM scheme in Table 5.*
Note that we can also write $\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(j)})$ as

$$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(j)}) \;=\; \min\left(1, \frac{\widehat{Z}_1}{\widehat{Z}_2}\right), \tag{16}$$

where we have denoted

$$\widehat{Z}_1 = \frac{1}{N} \sum_{n=1}^{N} w(\boldsymbol{\theta}^{(n)}), \qquad \widehat{Z}_2 = \frac{1}{N}\left(w(\boldsymbol{\theta}_{t-1}) + \sum_{n=1, n \neq j}^{N} w(\boldsymbol{\theta}^{(n)})\right). \tag{17}$$

Table 6: **The Independent Multiple Try Metropolis (I-MTM) algorithm.**

---

1. `Initialization:` Choose an initial state $\boldsymbol{\theta}_0$.

2. `FOR` $t = 1, \ldots, T$:

   (a) Draw $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(N)} \sim q(\boldsymbol{\theta})$.

   (b) Compute the importance weights

   $$w(\boldsymbol{\theta}^{(n)}) = \frac{\pi(\boldsymbol{\theta}^{(n)})}{q(\boldsymbol{\theta}^{(n)})}, \quad \text{with} \quad n = 1, \ldots, N. \tag{18}$$

   (c) Select one sample $\boldsymbol{\theta}^{(j)} \in \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}\}$, according to the probability mass function $\bar{w}_n = \frac{w(\boldsymbol{\theta}^{(n)})}{\sum_{i=1}^{N} w(\boldsymbol{\theta}^{(i)})}$.

   (d) Set $\boldsymbol{\theta}_t = \boldsymbol{\theta}^{(j)}$ with probability

   $$\begin{aligned}\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(j)}) \;&=\; \min\left(1, \frac{w(\boldsymbol{\theta}^{(j)}) + \sum_{n=1, n \neq j}^{N} w(\boldsymbol{\theta}^{(n)})}{w(\boldsymbol{\theta}_{t-1}) + \sum_{n=1, n \neq j}^{N} w(\boldsymbol{\theta}^{(n)})}\right), \\ &=\; \min\left(1, \frac{\widehat{Z}_1}{\widehat{Z}_2}\right),\end{aligned} \tag{19}$$

   otherwise, set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$.

3. `Return:` $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$.

Alternative version (I-MTM2). From the IS theory, we know that $\widehat{Z}_1 = \frac{1}{N}\sum_{n=1}^{N} w(\boldsymbol{\theta}^{(n)})$ is an unbiased estimator of the normalizing constant Z of the target π (a.k.a, Bayesian evidence or marginal likelihood). It suggests to replace $\widehat{Z}_2$ with other unbiased estimators of Z (without jeopardizing the ergodicity of the chain). For instance, instead of recycling the samples generated in the same iteration as auxiliary points as in Eq. (14), we could reuse samples generated in the previous iteration $t-1$. This alternative version of I-MTM method (I-MTM2) is given in Table 7. Note that, in both cases I-MTM and I-MTM2, the selected candidate $\boldsymbol{\theta}^{(j)}$ is drawn from the following particle approximation of the target $\bar{\pi}$,

$$\widehat{\pi}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(1:N)}) = \sum_{i=1}^{N} \bar{w}(\boldsymbol{\theta}^{(i)})\delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}), \qquad \bar{w}_i = \bar{w}(\boldsymbol{\theta}^{(i)}) = \frac{w(\boldsymbol{\theta}^{(i)})}{\sum_{n=1}^{N} w(\boldsymbol{\theta}^{(n)})}, \qquad (20)$$

i.e., $\boldsymbol{\theta}^{(j)} \sim \widehat{\pi}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(1:N)})$. The acceptance probability α used in I-MTM2 can be also justified considering a proper IS weighting of a resampled particle [41] and using the expression (7) related to the standard MH method, as discussed in [29]. Figure 2 provides a graphical representation of the I-MTM schemes.

Table 7: **Alternative version of I-MTM method (I-MTM2).**

1. `Initialization`: Choose an initial state $\boldsymbol{\theta}_0$, and obtain an initial approximation $\widehat{Z}_0 \approx Z$.

2. `FOR` $t = 1, \ldots, T$:

    (a) Draw $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(N)} \sim q(\boldsymbol{\theta})$.

    (b) Compute the importance weights

    $$w(\boldsymbol{\theta}^{(n)}) = \frac{\pi(\boldsymbol{\theta}^{(n)})}{q(\boldsymbol{\theta}^{(n)})}, \quad \text{with} \quad n = 1, \ldots, N. \qquad (21)$$

    (c) Select one sample $\boldsymbol{\theta}^{(j)} \in \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}\}$, according to the probability mass function $\bar{w}_n = \frac{1}{N\widehat{Z}^*} w(\boldsymbol{\theta}^{(n)})$ where $\widehat{Z}^* = \frac{1}{N}\sum_{i=1}^{N} w(\boldsymbol{\theta}^{(i)})$.

    (d) Set $\boldsymbol{\theta}_t = \boldsymbol{\theta}^{(j)}$ and $\widehat{Z}_t = \widehat{Z}^*$ with probability

    $$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(j)}) = \min\left(1, \frac{\widehat{Z}^*}{\widehat{Z}_{t-1}}\right), \qquad (22)$$

    otherwise, set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$ and $\widehat{Z}_t = \widehat{Z}_{t-1}$.
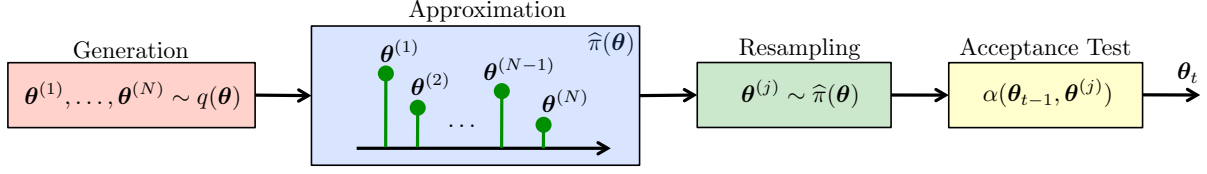
11

Figure 2: Graphical representation of the I-MTM schemes.

4.1.3 Reusing candidates in parallel I-MTM chains

Let us consider to run C independent parallel chains yielded by an I-MTM scheme. In this case, we have NC evaluations of the target function π and C resampling steps performed at each iteration (so that we have NCT total target evaluations and CT total resampling steps).

In literature, different authors have suggested to recycle the N candidates, $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(N)} \sim q(\boldsymbol{\theta})$, in order to reduce the number of evaluations of the target pdf [42] . The idea is to performs C-times the resampling procedure considering the same set of candidates, $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(N)}$ (a similar approach was proposed in [20]). Each resampled candidate is then tested as possible future state of one chain. In this scenario, The number of target evaluations per iteration is only N (hence, the total number of evaluation of π is NT). However, the resulting C parallel chains are no longer independent, and there is a lose of performance w.r.t. the independent chains. There exists also the possibility of reducing the total number of resampling steps, as suggested in the Block Independent MTM scheme [42] (but the dependence among the chains grows even more).

4.2 Particle Metropolis-Hastings (PMH) method

Assume that the variable of interest is formed by only a dynamical variable, i.e., $\boldsymbol{\theta} = \mathbf{x} = x_{1:D} = [x_1 \ldots, x_D]^\top$ (see Section 2). This is the case of inferring a hidden state in state-space model, for instance. More generally, let assume that we are able to factorize the target density as

$$\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x}) \quad = \quad \gamma_1(x_1)\gamma_2(x_2|x_1)\cdots\gamma_D(x_D|x_{1:D-1}), \tag{23}$$

$$= \quad \gamma_1(x_1)\prod_{d=2}^{D}\gamma_d(x_d|x_{d-1}). \tag{24}$$

The Particle Metropolis Hastings (PMH) method [43] is an efficient MCMC technique, proposed independently from the MTM algorithm, specifically designed for being applied in this framework. Indeed, we can take advantage of the factorization of the target pdf and consider a proposal pdf decomposed in the same fashion

$$q(\mathbf{x}) = q_1(x_1)q_2(x_2|x_1)\cdots q_D(x_D|x_{1:D-1}) = q_1(x_1)\prod_{d=2}^{D}q_d(x_d|x_{d-1}).$$

Then, as in a batch IS scheme, given an n-th sample $\mathbf{x}^{(n)} = x_{1:D}^{(n)} \sim q(\mathbf{x})$ with $x_d^{(n)} \sim q_d(x_d|x_{d-1})$, we assign the importance weight

$$w(\mathbf{x}^{(n)}) = w_D^{(n)} = \frac{\pi(\mathbf{x}^{(n)})}{q(\mathbf{x}^{(n)})} = \frac{\gamma_1(x_1^{(n)})\gamma_2(x_2^{(n)}|x_1^{(n)})\cdots\gamma_D(x_D^{(n)}|x_{1:D-1}^{(n)})}{q_1(x_1^{(n)})q_2(x_2^{(n)}|x_1^{(n)})\cdots q_D(x_D^{(n)}|x_{1:D-1}^{(n)})}. \tag{25}$$

The previous expression suggests a recursive procedure for computing the importance weights: starting with $w_1^{(n)} = \frac{\pi(x_1^{(n)})}{q(x_1^{(n)})}$ and then

$$w_d^{(n)} = w_{d-1}^{(n)}\beta_d^{(n)} = \prod_{j=1}^{d} \beta_j^{(n)}, \qquad d = 1, \ldots, D, \tag{26}$$

where we have set

$$\beta_1^{(n)} = w_1^{(n)} \quad \text{and} \quad \beta_d^{(n)} = \frac{\gamma_d(x_d^{(n)}|x_{1:d-1}^{(n)})}{q_d(x_d^{(n)}|x_{1:d-1}^{(n)})}, \tag{27}$$

for $d = 2, \ldots, D$. This method is usually referred as Sequential Importance Sampling (SIS). If resampling steps are also employed at some iteration, the method is called Sequential Importance Resampling (SIR), a.k.a., *particle filtering* (PF) (see Appendix B). PMH uses a SIR approach for providing the particle approximation $\widehat{\pi}(\mathbf{x}|\mathbf{x}^{(1:N)}) = \sum_{i=1}^{N} \bar{w}_D^{(i)}\delta(\mathbf{x}-\mathbf{x}^{(i)})$ where $\bar{w}_D^{(i)} = \frac{w_D^{(i)}}{\sum_{n=1}^{N} w_D^{(n)}}$ and $w_D^{(i)} = w(\mathbf{x}^{(i)})$, obtained using Eq. (26) (with a proper weighting of a resampled particle [41, 29]). Then, one particle is drawn from this approximation, i.e., with a probability proportional to the corresponding normalized weight.

Estimation of the marginal likelihood Z in particle filtering. SIR combines the SIS approach with the application of resampling procedures. In SIR, a consistent estimator of Z is given by

$$\widetilde{Z} = \prod_{d=1}^{D} \left[\sum_{n=1}^{N} \bar{w}_{d-1}^{(n)}\beta_d^{(n)} \right], \quad \text{where} \quad \bar{w}_{d-1}^{(i)} = \frac{w_{d-1}^{(i)}}{\sum_{n=1}^{N} w_{d-1}^{(n)}}. \tag{28}$$

Due to the application of the resampling, in SIR the standard estimator

$$\widehat{Z} = \frac{1}{N}\sum_{n=1}^{N} w_D^{(n)} = \frac{1}{N}\sum_{n=1}^{N} w(\mathbf{x}^{(n)}), \tag{29}$$

is a possible alternative *only if* a proper weighting of the resampled particles is applied [29, 41] (otherwise, it is not an estimator of Z). If a proper weighting of a resampled particle is employed, both $\widetilde{Z}$ and $\widehat{Z}$ are equivalent estimators of Z [41, 29, 28]. Without the use of resampling steps (i.e., in SIS), $\widetilde{Z}$ and $\widehat{Z}$ are always equivalent estimators [29]. See also Appendix B.

The complete description of PMH is provided in Table 8 considering the use of $\widetilde{Z}$. At each iteration, a particle filter is run in order to provide an approximation by N weighted samples of the

13

Table 8: **Particle Metropolis-Hastings (PMH) algorithm.**

---

1. `Initialization:` Choose a initial state $\mathbf{x}_0$, obtain an initial estimation $\widetilde{Z}_0 \approx Z$.

2. `For` $t = 1, \ldots, T$:

   (a) We employ a SIR approach for drawing with N particles and weighting them, $\{\mathbf{x}^{(i)}, w_D^{(i)}\}_{i=1}^N$, i.e., we obtain sequentially a particle approximation $\widehat{\pi}(\mathbf{x}) = \sum_{i=1}^N \bar{w}_D^{(i)} \delta(\mathbf{x} - \mathbf{x}^{(i)})$ where $\mathbf{x}^{(i)} = [x_1^{(i)}, \ldots, x_D^{(i)}]^\top$. Furthermore, we also obtain $\widetilde{Z}^*$ as in Eq. (28).

   (b) Draw $\mathbf{x}^* \sim \widehat{\pi}(\mathbf{x}|\mathbf{x}^{(1:N)})$, i.e., choose a particle $\mathbf{x}^* = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability $\bar{w}_D^{(i)}$, $i = 1, \ldots, N$.

   (c) Set $\mathbf{x}_t = \mathbf{x}^*$ and $\widetilde{Z}_t = \widetilde{Z}^*$ with probability

   $$\alpha = \min\left[1, \frac{\widetilde{Z}^*}{\widetilde{Z}_{t-1}}\right], \tag{30}$$

   otherwise set $\mathbf{x}_t = \mathbf{x}_{t-1}$ and $\widetilde{Z}_t = \widetilde{Z}_{t-1}$.

   (d) `Return:` $\{\mathbf{x}_t\}_{t=1}^T$ where $\mathbf{x}_t = [x_{1,t}, \ldots, x_{D,t}]^\top$.

---

measure of the target. Then, a sample among the N weighted particles is chosen by one resampling step. This selected sample is then accepted or rejected as next state of the chain according to an MH-type acceptance probability, which involves two estimators of marginal likelihood Z. PMH is also related to other popular method in molecular simulation called Configurational Bias Monte Carlo (CBMC) [44].

4.2.1 Relationship among I-MTM2, PMH and I-MH

A simple look at I-MTM2 and PMH shows that they are strictly related [28]. Indeed, the structure of the two algorithms coincides. The main difference lies that the candidates in PMH are generated sequentially, using a SIR scheme. If no resampling steps are applied, then I-MTM2 and PMH are *exactly* the same algorithm, where the candidates are drawn in a *batch* setting or *sequential* way. Hence, the application of resampling steps is the main difference between the generation procedures of PMH and I-MTM2. Owing to the use of resampling, the candidates $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ proposed by PMH are not independent (differently from I-MTM2). As an example, Figure 3 shows $N = 40$ particles (with $D = 10$) generated and weighted by SIS and SIR procedures (each path is a generated particle $\mathbf{x}^{(i)} = x_{1:10}^{(i)}$). The generation of correlated samples can be also considered

in MTM methods without jeopardizing the ergodicity of the chain, as simply shown for instance in [16], for instance. Another difference is the use of $\widetilde{Z}$ or $\widehat{Z}$. However, if a proper weighting of a resampled particle is employed, both estimators coincide [29, 41, 28]. Furthermore, both I-MTM2 and PMH can be considered as I-MH schemes where a proper importance sampling weighting of a resampled particle is employed [41]. Namely, I-MTM2 and PMH are equivalent to an I-MH technique using the following complete proposal pdf,

$$\widetilde{q}(\boldsymbol{\theta}) = \int_{\mathcal{D}^N} \widehat{\pi}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(1:N)}) \left[\prod_{i=1}^N q(\boldsymbol{\theta}^{(i)}) \right] d\boldsymbol{\theta}^{(1:N)}, \tag{31}$$

where $\widetilde{\pi}$ is given in Eq. (20), i.e., $\boldsymbol{\theta}^{(j)} \sim \widetilde{q}(\boldsymbol{\theta})$, and then considering the generalized (proper) IS weighting, $w(\boldsymbol{\theta}^{(j)}) = \widehat{Z}^*$, $w(\boldsymbol{\theta}_{t-1}) = \widehat{Z}_{t-1}$ [29, 41]. For further details see Appendix A.

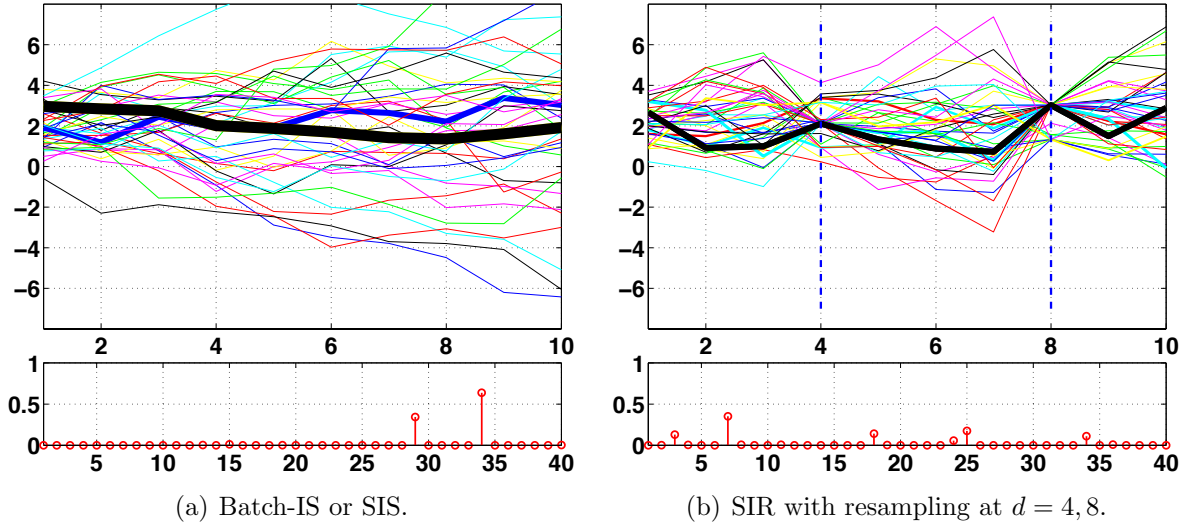(a) Batch-IS or SIS.   (b) SIR with resampling at $d = 4, 8$.

Figure 3:   Graphical representation of SIS and SIR. We consider as target density a multivariate Gaussian pdf, $\bar{\pi}(\mathbf{x}) = \prod_{d=1}^{10} \mathcal{N}(x_d|2, \frac{1}{2})$. In each figure, every component of different particles are represented, so that each particle $\mathbf{x}^{(i)} = x_{1:D}^{(i)}$ forms a *path* (with $D = 10$). We set $N = 40$. The normalized weights $\bar{w}_D^{(i)} = w(\mathbf{x}^{(i)})$ corresponding to each figure are also shown in the bottom. The line-width of each path is proportional to the corresponding weight $\bar{w}_D^{(i)}$. The particle corresponding to the greatest weight is always depicted in black. The proposal pdfs used are $q_1(x_1) = \mathcal{N}(x_1|2, 1)$ and $q(x_d|x_{d-1}) = \mathcal{N}(x_d|x_{d-1}, 1)$ for $d \geq 2$. **(a)** Batch IS or SIS. **(b)** SIR resampling steps at the iterations $d = 4, 8$.

4.2.2   Particle Marginal Metropolis-Hastings (PMMH) method

Assume now that the variable of interest if formed by both dynamical and static variables, i.e., $\boldsymbol{\theta} = [\mathbf{x}, \boldsymbol{\lambda}]^\top$. For instance, this is the case of inferring both, an hidden state $\mathbf{x}$ in state-space model, and static parameters $\boldsymbol{\lambda}$ of the model. The Particle Marginal Metropolis-Hastings (PMMH) technique is a extension of PMH which addresses this problem.

15

Let us consider $\mathbf{x} = x_{1:D} = [x_1, x_2, \ldots, x_D] \in \mathbb{R}^{d_x}$, and an additional model parameter $\boldsymbol{\lambda} \in \mathbb{R}^{d_\lambda}$ to be inferred as well ($\boldsymbol{\theta} = [\mathbf{x}, \boldsymbol{\lambda}]^\top \in \mathbb{R}^D$, with $D = d_x + d_\lambda$). Assuming a prior pdf $g_\lambda(\boldsymbol{\lambda})$ over $\boldsymbol{\lambda}$, and a factorized complete posterior pdf $\bar{\pi}(\boldsymbol{\theta}) = \bar{\pi}(\mathbf{x}, \boldsymbol{\lambda})$,

$$\bar{\pi}(\mathbf{x}, \boldsymbol{\lambda}) \propto \pi(\mathbf{x}, \boldsymbol{\lambda}) \quad = \quad g_\lambda(\boldsymbol{\lambda})\pi(\mathbf{x}|\boldsymbol{\lambda}),$$

where $\pi(\mathbf{x}|\boldsymbol{\lambda}) = \gamma_1(x_1|\boldsymbol{\lambda}) \prod_{d=2}^D \gamma_d(x_d|x_{1:d-1}, \boldsymbol{\lambda})$. For a specific value of $\boldsymbol{\lambda}$, we can use a particle filter approach, obtaining the approximation $\widehat{\pi}(\mathbf{x}|\boldsymbol{\lambda}) = \sum_{n=1}^N \bar{w}_D^{(n)} \delta(\mathbf{x} - \mathbf{x}^{(n)})$ and the estimator $\widetilde{Z}(\boldsymbol{\lambda})$, as described above. The PMMH technique is then summarized in Table 9. The pdf $q_\lambda(\boldsymbol{\lambda}|\boldsymbol{\lambda}_{t-1})$ denotes the proposal density for generating possible values of $\boldsymbol{\lambda}$. Observe that, with the specific choice $q_\lambda(\boldsymbol{\lambda}|\boldsymbol{\lambda}_{t-1}) = g_\lambda(\boldsymbol{\lambda})$, then the acceptance function becomes

$$\alpha = \min\left[1, \frac{\widetilde{Z}(\boldsymbol{\lambda}^*)}{\widetilde{Z}(\boldsymbol{\lambda}_{t-1})}\right]. \tag{32}$$

Note also that PMMH w.r.t. to $\boldsymbol{\lambda}$ can be interpreted as MH method where an the posterior cannot be evaluated point-wise. Indeed, $\widetilde{Z}(\boldsymbol{\lambda})$ approximates the marginal likelihood $p(\mathbf{y}|\boldsymbol{\lambda})$ [45].

Table 9: **Particle Marginal MH (PMMH) algorithm**

---

1. `Initialization:` Choose the initial states $\mathbf{x}_0$, $\boldsymbol{\lambda}_0$, and an initial approximation $\widetilde{Z}_0(\boldsymbol{\lambda}) \approx Z(\boldsymbol{\lambda}) \approx p(\mathbf{y}|\boldsymbol{\lambda})$.

2. `For` $t = 1, \ldots, T$:

   (a) Draw $\boldsymbol{\lambda}^* \sim q_\lambda(\boldsymbol{\lambda}|\boldsymbol{\lambda}_{t-1})$.

   (b) Given $\boldsymbol{\lambda}^*$, run a particle filter obtaining $\widehat{\pi}(\mathbf{x}|\boldsymbol{\lambda}^*) = \sum_{n=1}^N \bar{w}_D^{(n)} \delta(\mathbf{x} - \mathbf{x}^{(n)})$ and $\widetilde{Z}(\boldsymbol{\lambda}^*)$, as in Eq. (28).

   (c)

   (d) Draw $\mathbf{x}^* \sim \widehat{\pi}(\mathbf{x}|\boldsymbol{\lambda}^*, \mathbf{x}^{(1:N)})$, i.e., choose a particle $\mathbf{x}^* = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with probability $\bar{w}_D^{(i)}$, $i = 1, \ldots, N$.

   (e) Set $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}^*$, $\mathbf{x}_t = \mathbf{x}^*$, with probability

   $$\alpha = \min\left[1, \frac{\widetilde{Z}(\boldsymbol{\lambda}^*)g_\lambda(\boldsymbol{\lambda}^*)q_\lambda(\boldsymbol{\lambda}_{t-1}|\boldsymbol{\lambda}^*)}{\widetilde{Z}(\boldsymbol{\lambda}_{t-1})g_\lambda(\boldsymbol{\lambda}_{t-1})q_\lambda(\boldsymbol{\lambda}^*|\boldsymbol{\lambda}_{t-1})}\right]. \tag{33}$$

   Otherwise, set $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}^*$ and $\mathbf{x}_t = \mathbf{x}_{t-1}$.

3. `Return:` $\{\mathbf{x}_t\}_{t=1}^T$ and $\{\boldsymbol{\lambda}_t\}_{t=1}^T$.

16

4.3 Group Metropolis Sampling

The auxiliary weighted samples in the I-MTM schemes (i.e., the $N-1$ samples drawn at each iteration that are not selected to be compared with the previous state $\boldsymbol{\theta}_{t-1}$) can be recycled providing a consistent and more efficient estimators [41, 29].

The so-called Group Metropolis Sampling (GMS) method is shown in Table 10. GMS yields a sequence of sets of weighted samples $\mathcal{S}_t = \{\boldsymbol{\theta}_{n,t}, \rho_{n,t}\}_{n=1}^{N}$, for $t = 1, \ldots, T$, where we have denoted with $\rho_{n,t}$ the importance weights assigned to the samples $\boldsymbol{\theta}_{n,t}$ (see Figure 4). All the samples are then employed for a joint particle approximation of the target. Alternatively, GMS can directly provide an approximation of a specific moment of the target pdf (i.e., given a particular function f). The estimator of this specific moment provided by GMS is

$$\widetilde{I}_{NT} = \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \frac{\rho_{n,t}}{\sum_{i=1}^{N} \rho_{i,t}} f(\boldsymbol{\theta}_{n,t}) = \frac{1}{T} \sum_{t=1}^{T} \widetilde{I}_{N}^{(t)}. \tag{34}$$

Unlike in the I-MTM schemes, no resampling steps are performed in GMS. However, we can recover an I-MTM chain from the GMS output applying one resampling step when $\mathcal{S}_t \neq \mathcal{S}_{t-1}$, i.e.,

$$\boldsymbol{\theta}_t = \begin{cases} \boldsymbol{\theta}_t \sim \sum_{n=1}^{N} \frac{\rho_{n,t}}{\sum_{i=1}^{N} \rho_{i,t}} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{n,t}), & \text{if} \quad \mathcal{S}_t \neq \mathcal{S}_{t-1}, \\ \boldsymbol{\theta}_{t-1}, & \text{if} \quad \mathcal{S}_t = \mathcal{S}_{t-1}, \end{cases} \tag{38}$$

for $t = 1, \ldots, T$. More specifically, $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$ is a Markov chain obtained by one run of an I-MTM2 technique. The consistency of the GMS estimators is discussed in Appendix C. GMS can be also interpreted as an iterative IS scheme where an IS approximation of N samples is built at each iteration and compared with the previous IS approximation. This procedure is iterated T times and all the accepted IS estimators $\widetilde{I}_{N}^{(t)}$ are finally combined to provide a unique global approximation of NT samples. Note that the temporal combination of the IS estimators is obtained dynamically by the random repetitions due to the rejections in the acceptance test.

Remark 3. *The complete weighting procedure in GMS can be interpreted as the composition of two weighting schemes: (a) by an IS approach building $\{\rho_{n,t}\}_{n=1}^{N}$ and (b) by the possible random repetitions due to the rejections in the acceptance test.*

Figure 4 depicts a graphical representation of the GMS outputs as chain of sets $\mathcal{S}_t = \{\boldsymbol{\theta}_{n,t}, \rho_{n,t}\}_{n=1}^{N}$.

4.4 Ensemble MCMC algorithms

Another alternative procedure, often referred as Ensemble MCMC (EnMCMC) methods (a.k.a., called Locally weighted MCMC), involving several tries at each iteration [19, 21]. Related techniques has been proposed independently in different works [20, 22]. First, let us define the joint proposal density

$$q(\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)} | \boldsymbol{\theta}_t) : \mathcal{D}^N \to \mathbb{R}, \tag{39}$$

and, considering $N+1$ possible elements, $\mathcal{S} = \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}, \boldsymbol{\theta}^{(N+1)}\}$, we define the $D \times N$ matrix

$$\boldsymbol{\Theta}_{\neg k} = [\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(k-1)}, \boldsymbol{\theta}^{(k+1)}, \ldots, \boldsymbol{\theta}^{(N+1)}], \tag{40}$$

Table 10: **Group Metropolis Sampling**

1. `Initialization`: Choose an initial state $\boldsymbol{\theta}_0$ and an initial approximation $\widehat{Z}_0 \approx Z$.

2. `FOR` $t = 1, \ldots, T$:

   (a) Draw $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(N)} \sim q(\boldsymbol{\theta})$.

   (b) Compute the importance weights

   $$w(\boldsymbol{\theta}^{(n)}) = \frac{\pi(\boldsymbol{\theta}^{(n)})}{q(\boldsymbol{\theta}^{(n)})}, \quad \text{with} \quad n = 1, \ldots, N, \tag{35}$$

   define $\mathcal{S}^* = \{\boldsymbol{\theta}^{(n)}, w(\boldsymbol{\theta}^{(n)})\}_{n=1}^N$; and compute $\widehat{Z}^* = \frac{1}{N} \sum_{n=1}^N w(\boldsymbol{\theta}^{(n)})$.

   (c) Set $\mathcal{S}_t = \mathcal{S}^*$, i.e.,
   $$\mathcal{S}_t = \left\{ \boldsymbol{\theta}_{n,t} = \boldsymbol{\theta}^{(n)}, \rho_{n,t} = w(\boldsymbol{\theta}^{(n)}) \right\}_{n=1}^N,$$
   and $\widehat{Z}_t = \widehat{Z}^*$, with probability

   $$\alpha(\mathcal{S}_{t-1}, \mathcal{S}^*) = \min\left[1, \frac{\widehat{Z}^*}{\widehat{Z}_{t-1}} \right]. \tag{36}$$

   Otherwise, set $\mathcal{S}_t = \mathcal{S}_{t-1}$ and $\widehat{Z}_t = \widehat{Z}_{t-1}$.

3. `Return`: All the sets $\{\mathcal{S}_t\}_{t=1}^T$, or $\{\widetilde{I}_N^{(t)}\}_{t=1}^T$ where

$$\widetilde{I}_N^{(t)} = \sum_{n=1}^N \frac{\rho_{n,t}}{\sum_{i=1}^N \rho_{i,t}} g(\boldsymbol{\theta}_{n,t}), \tag{37}$$

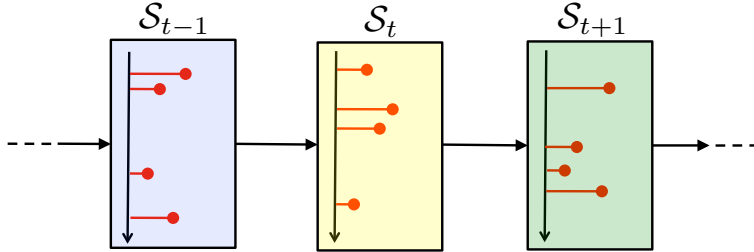and $\widetilde{I}_{NT} = \frac{1}{T} \sum_{t=1}^T \widetilde{I}_N^{(t)}$.

Figure 4: Chain of sets $\mathcal{S}_t = \{\boldsymbol{\theta}_{n,t}, \rho_{n,t}\}_{n=1}^N$ generated by the GMS method (graphical representation with $N = 4$).

with columns all the vectors in $\mathcal{S}$ with the exception of $\boldsymbol{\theta}^{(k)}$. For simplicity, in the followings we abuse of the notation writing $q(\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}|\boldsymbol{\theta}_t) = q(\boldsymbol{\Theta}_{\neg N+1}|\boldsymbol{\theta}_t)$, for instance. One simple example of joint proposal pdf is

$$q(\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}|\boldsymbol{\theta}_t) = \prod_{n=1}^{N} q(\boldsymbol{\theta}^{(n)}|\boldsymbol{\theta}_t), \tag{41}$$

i.e., considering independence among $\boldsymbol{\theta}^{(n)}$'s (and having the same marginal proposal pdf q). More sophisticated joint proposal densities can be employed. A generic EnMCMC algorithm is outlined in Table 11.

Table 11: **Generic EnMCMC algorithm**

---

1. `Initialization:` Choose an initial state $\boldsymbol{\theta}_0$.

2. `FOR` $t = 1, \ldots, T$:

    (a) Draw $\widetilde{\boldsymbol{\theta}}^{(1)}, \ldots, \widetilde{\boldsymbol{\theta}}^{(N)} \sim q(\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}|\boldsymbol{\theta}_{t-1})$.

    (b) Set $\widetilde{\boldsymbol{\theta}}^{(N+1)} = \boldsymbol{\theta}_{t-1}$.

    (c) Set $\boldsymbol{\theta}_t = \widetilde{\boldsymbol{\theta}}^{(j)}$, resampling $\widetilde{\boldsymbol{\theta}}^{(j)}$ within the set

    $$\{\widetilde{\boldsymbol{\theta}}^{(1)}, \ldots, \widetilde{\boldsymbol{\theta}}^{(N)}, \widetilde{\boldsymbol{\theta}}^{(N+1)} = \boldsymbol{\theta}_{t-1}\},$$

    formed by $N + 1$ samples, according to the probability mass function

    $$\alpha(\boldsymbol{\theta}_{t-1}, \widetilde{\boldsymbol{\theta}}^{(j)}) = \frac{\pi(\widetilde{\boldsymbol{\theta}}^{(j)})q(\boldsymbol{\Theta}_{\neg j}|\widetilde{\boldsymbol{\theta}}^{(j)})}{\sum_{\ell=1}^{N+1} \pi(\widetilde{\boldsymbol{\theta}}^{(\ell)})q(\boldsymbol{\Theta}_{\neg \ell}|\widetilde{\boldsymbol{\theta}}^{(\ell)})},$$

    for $j = 1, \ldots, N + 1$ and $\boldsymbol{\Theta}_{\neg j}$ is defined in Eq. (40).

3. `Return:` $\{\boldsymbol{\theta}_t\}_{t=1}^T$.

Remark 4. *Note that with respect to the generic MTM method, EnMCMC does not require to draw auxiliary samples and weights. Therefore, in EnMCMC a smaller number of evaluation of target is required w.r.t. a generic MTM scheme.*

4.4.1 Independent Ensemble MCMC

In this section, we present an interesting special case, which employs a single proposal pdf $q(\boldsymbol{\theta})$ independent on the previous state of the chain, i.e.,

$$q(\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}|\boldsymbol{\theta}_t) = q(\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}) = \prod_{n=1}^{N} q(\boldsymbol{\theta}^{(n)}). \tag{42}$$

In this case, the technique can be simplified as shown below. At each iteration, the algorithm described in Table 12 generates N new samples $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(N)}$ and then resample the new state $\boldsymbol{\theta}_t$ within a set of $N+1$ samples, $\{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}, \boldsymbol{\theta}_{t+1}\}$ (which includes the previous state), according to the probabilities

$$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(j)}) = \bar{w}_j = \frac{w(\boldsymbol{\theta}^{(j)})}{\sum_{i=1}^{N} w(\boldsymbol{\theta}^{(i)}) + w(\boldsymbol{\theta}_{t-1})}, \qquad j = 1, \ldots, N+1, \qquad (43)$$

where $w(\boldsymbol{\theta}) = \frac{\pi(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}$ denotes the importance sampling weight. Note that Eq. (43) for $N = 1$ becomes

$$
\begin{aligned}
\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(j)}) &= \frac{w(\boldsymbol{\theta}^{(j)})}{w(\boldsymbol{\theta}^{(i)}) + w(\boldsymbol{\theta}_{t-1})}, \\
&= \frac{\frac{\pi(\boldsymbol{\theta}^{(j)})}{q(\boldsymbol{\theta}^{(j)})}}{\frac{\pi(\boldsymbol{\theta}^{(j)})}{q(\boldsymbol{\theta}^{(j)})} + \frac{\pi(\boldsymbol{\theta}_{t-1})}{q(\boldsymbol{\theta}_{t-1})}}, \\
&= \frac{\pi(\boldsymbol{\theta}^{(j)})q(\boldsymbol{\theta}_{t-1})}{\pi(\boldsymbol{\theta}^{(j)})q(\boldsymbol{\theta}_{t-1}) + \pi(\boldsymbol{\theta}_{t-1})q(\boldsymbol{\theta}^{(j)})}, \qquad (44)
\end{aligned}
$$

that is the Barker's acceptance function (see [46, 47]).

Table 12: **EnMCMC with an independent proposal pdf (I-EnMCMC).**

---

1. `Initialization`: Choose an initial state $\boldsymbol{\theta}_0$.

2. `FOR` $t = 1, \ldots, T$:

   (a) Draw $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(N)} \sim q(\boldsymbol{\theta})$, and set $\boldsymbol{\theta}^{(N+1)} = \boldsymbol{\theta}_{t-1}$.

   (b) Compute the importance weights

   $$w(\boldsymbol{\theta}^{(n)}) = \frac{\pi(\boldsymbol{\theta}^{(n)})}{q(\boldsymbol{\theta}^{(n)})}, \quad \text{with} \quad n = 1, \ldots, N+1. \qquad (45)$$

   (c) Set $\boldsymbol{\theta}_t = \boldsymbol{\theta}^{(j)}$, resampling $\boldsymbol{\theta}^{(j)}$ within the set $\{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}, \boldsymbol{\theta}_{t+1}\}$ formed by $N+1$ samples, according to the probability mass function

   $$\alpha(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(j)}) = \bar{w}_j = \frac{w(\boldsymbol{\theta}^{(j)})}{\sum_{i=1}^{N} w(\boldsymbol{\theta}^{(i)}) + w(\boldsymbol{\theta}_{t-1})}.$$

3. `Return`: $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$.

As discussed in [42, Appendix B], [48, Appendix C], [41], the density of a resampled candidate becomes closer and closer to $\bar{\pi}$ as N grows, i.e., $N \to \infty$. Hence, the performance of I-EnMCMC

clearly improves with $N \to \infty$ (see Appendix A). The I-EnMCMC algorithm produces an ergodic chain with invariant density $\bar{\pi}$, by resampling $N+1$ samples at each iteration (N new samples $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}$ from q and setting $\boldsymbol{\theta}^{(N+1)} = \boldsymbol{\theta}_{t-1}$). Figure 5 summarizes the steps of I-EnMCMC.
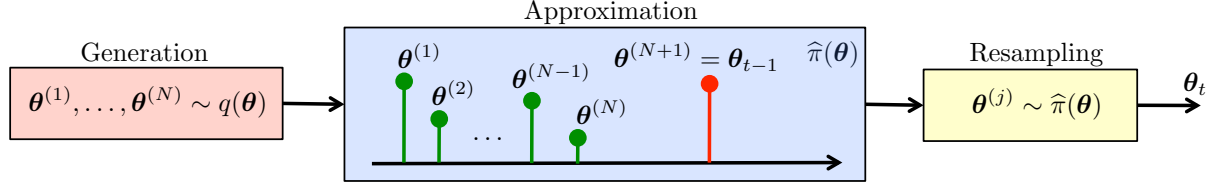
Figure 5: Graphical representation of the steps of the I-EnMCMC scheme.

4.5 Delayed Rejection Metropolis (DRM) Sampling

An alternative use of different candidates in one iteration of a Metropolis-type method is given in [23, 24, 25]. The idea behind the proposed algorithm, called Delayed Rejection Metropolis (DRM) algorithm, is the following. As in a standard MH method, at each iteration, one sample is proposed $\boldsymbol{\theta}^{(1)} \sim q_1(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$ and accepted with probability

$$\alpha_1(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(1)}) = \min\left(1, \frac{\pi(\boldsymbol{\theta}^{(1)})q_1(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}^{(1)})}{\pi(\boldsymbol{\theta}_{t-1})q_1(\boldsymbol{\theta}^{(1)}|\boldsymbol{\theta}_{t-1})}\right).$$

If $\boldsymbol{\theta}^{(1)}$ is accepted then $\boldsymbol{\theta}_t = \boldsymbol{\theta}^{(1)}$ and the chain is moved forward. If $\boldsymbol{\theta}^{(1)}$ is rejected, the DRM method suggests of drawing another samples $\boldsymbol{\theta}^{(2)} \sim q_2(\boldsymbol{\theta}|\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}_{t-1})$ (considering a different proposal pdf q_2 taking into account possibly the previous candidate $\boldsymbol{\theta}^{(1)}$) and accepted with a suitable acceptance probability

$$\alpha_2(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(2)}) =$$
$$= \min\left(1, \frac{\pi(\boldsymbol{\theta}^{(2)})q_1(\boldsymbol{\theta}^{(1)}|\boldsymbol{\theta}^{(2)})q_2(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)})(1 - \alpha_1(\boldsymbol{\theta}^{(2)}, \boldsymbol{\theta}^{(1)}))}{\pi(\boldsymbol{\theta}_{t-1})q_1(\boldsymbol{\theta}^{(1)}|\boldsymbol{\theta}_{t-1})q_2(\boldsymbol{\theta}^{(2)}|\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}_{t-1})(1 - \alpha_1(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(1)}))}\right).$$

The acceptance function $\alpha_2(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(2)})$ is designed in order to ensure the ergodicity of the chain. If $\boldsymbol{\theta}^{(2)}$ is rejected we can set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$ and perform another iteration of the algorithm, or continue with this iterative strategy drawing $\boldsymbol{\theta}^{(3)} \sim q_3(\boldsymbol{\theta}|\boldsymbol{\theta}^{(2)}, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}_{t-1})$ and test it with a proper probability $\alpha_3(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(3)})$. The DRM algorithm with only 2 acceptance stages is outlined in Table 13 and summarized in Figure 6.

Remark 5. *Note that the proposal pdf can be improved at each intermediate stage ($\boldsymbol{\theta}^{(1)} \sim q_1(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$, $\boldsymbol{\theta}^{(2)} \sim q_2(\boldsymbol{\theta}|\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}_{t-1})$ etc.), using the information provided by the previous generated samples and the corresponding target evaluations.*

The idea behind DRM of creating a path of intermediate points, then improving the proposal pdf, and hence fostering larger jumps have been also considered in other works [14, 49].

Table 13: **Delayed Rejection Metropolis algorithm with 2 acceptance steps.**

1. `Initialization`: Choose an initial state $\boldsymbol{\theta}_0$.

2. `FOR` $t = 1, \ldots, T$:

   (a) Draw $\boldsymbol{\theta}^{(1)} \sim q_1(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$ and $u_1 \sim \mathcal{U}([0,1])$.

   (b) Define the probability

   $$\alpha_1(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(1)}) = \min\left(1, \frac{\pi(\boldsymbol{\theta}^{(1)})q_1(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}^{(1)})}{\pi(\boldsymbol{\theta}_{t-1})q_1(\boldsymbol{\theta}^{(1)}|\boldsymbol{\theta}_{t-1})}\right), \tag{46}$$

   (c) If $u_1 \leq \alpha_1(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(1)})$, set $\boldsymbol{\theta}_t = \boldsymbol{\theta}^{(1)}$.

   (d) Otherwise, if $u_1 > \alpha_1(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(1)})$, do:
   (d1) Draw $\boldsymbol{\theta}^{(2)} \sim q_2(\boldsymbol{\theta}|\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}_{t-1})$ and $u_2 \sim \mathcal{U}([0,1])$.
   (d2) Given the function

   $$\psi(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(2)}|\boldsymbol{\theta}^{(1)}) = \pi(\boldsymbol{\theta}_{t-1})q_1(\boldsymbol{\theta}^{(1)}|\boldsymbol{\theta}_{t-1}) \times$$
   $$q_2(\boldsymbol{\theta}^{(2)}|\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}_{t-1})(1 - \alpha_1(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(1)})),$$

   and the probability

   $$\alpha_2(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(2)}) = \min\left(1, \frac{\psi(\boldsymbol{\theta}^{(2)}, \boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}^{(1)})}{\psi(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(2)}|\boldsymbol{\theta}^{(1)})}\right). \tag{47}$$

   (d3) If $u_2 \leq \alpha_2(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(2)})$, set $\boldsymbol{\theta}_t = \boldsymbol{\theta}^{(2)}$.
   (d4) Otherwise, if $u_2 > \alpha_1(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}^{(2)})$, set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$.
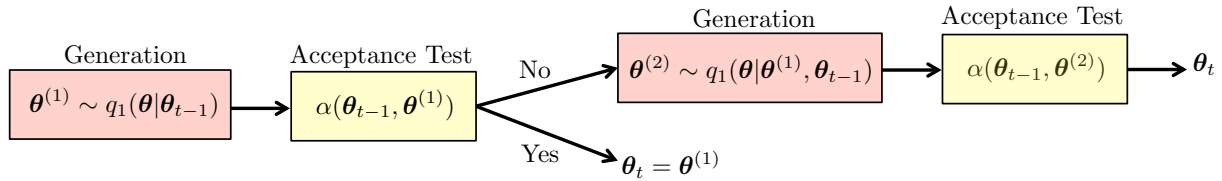
3. `Return`: $\{\boldsymbol{\theta}_t\}_{t=1}^T$.

Figure 6: Graphical representation of the DRM scheme with two acceptance tests.

5 Summary: computational cost, differences and connections

The performance of the algorithms described above improves as N grows, in general: the correlation among samples vanishes to zero, and the acceptance rate of new state approaches one (see Section 6.1).[5] As N increases, they become similar and similar to an exact sampler drawing independent samples directly from the target density (for MTM, PMH and EnMCMC schemes the explanation is given in Appendix A). However, this occurs at the expense of an additional computational cost.

In Table 14, we summarize the total number of target evaluations, E, and the total number of samples used in the final estimators, Q (without considering to remove any burn-in period). The generic MTM algorithm has the greatest number of target evaluations. However, a random-walk proposal pdf can be used in a generic MTM algorithm and, in general, it fosters the exploration of the state space. In this sense, the generic EnMCMC seems to be preferable w.r.t. MTM, since $E = NT$ and the random-walk proposal can be applied. A disadvantage of the EnMCMC schemes is that their acceptance function seems worse in terms of Peskun's ordering [50] (see numerical results in Section 6.1). Namely, fixing the number of N tries, the target π the proposal q pdfs, the MTM schemes seem to provide greater acceptance rates than the corresponding EnMCMC techniques. This is theoretically proved for $N = 1$ [50], and the difference vanishes to zero as N grows. The GMS technique, like other strategies [20, 42], has been proposed to recycle samples or re-use target evaluations, in order to increase Q (see also Section 4.1.3).

Table 14: **Total number of target evaluations, E, and total number of samples, Q, used in the final estimators.**

Algorithm	MTM	I-MTM	I-MTM2	PMH
E	$2NT - 1$	NT	NT	NT
Q	T	T	T	T

Algorithm	GMS	EnMCMC	I-EnMCMC	DMR
E	NT	NT	NT	NT
Q	NT	T	T	T

In PMH, the components of the different tries are drawn sequentially and they are correlated due to the application of the resampling steps. In DMR, each candidate is drawn in a batch way (all the components jointly) but the different candidates are drawn in a sequential manner (see Figure 6), $\boldsymbol{\theta}^{(1)}$ then $\boldsymbol{\theta}^{(2)}$ etc. The benefit of this strategy is that the proposal pdf can be improved considering the previous generated tries. Hence, if the proposal takes into account the previous samples, DMR generates correlated candidates as well. The main disadvantage of DRM is that

---

[5]Generally, an acceptance rate close to 1 is not an evidence of good performance for an MCMC algorithm. However, for the techniques tackled in this work, the situation is different: as N grows, the procedure used for proposing a novel possible state (involving N tries, resampling steps etc.) becomes better and better, yielding a better approximation of the target pdf. See Appendix A for further details.

the implementation for a generic $N > 2$ is not straightforward.

I-MTM and I-MTM2 differs for the acceptance function employed. Furthermore, The main difference between the I-MTM2 and PMH schemes is the use of resampling steps during the generation the different tries. For this reason, the candidates of PMH are correlated (unlike in I-MTM2). I-MTM2 and PMH can be interpreted as I-MH methods using a sophisticated proposal density $\widehat{q}(\boldsymbol{\theta})$ in Eq. (31), and an extended IS weighting procedure is employed. Note that, indeed, $\widehat{q}$ cannot be evaluated pointwise, hence a standard IS weighting strategy cannot be employed.

6 Numerical Experiments

We test different MCMC using multiple candidates in different numerical experiments. In the first example, an exhaustive comparison among several techniques with an independent proposal is given. We have considered different number of tries, length of the chain, parameters of the proposal pdfs and also different dimension of the inference problem. In the second numerical simulation, we compare different particle methods. The third one regards the hyperparameter selection for a Gaussian Process (GP) regression model. The last two examples are localization problems in a wireless sensor network (WSN): in the fourth one some parameters of the WSN are also tuned, whereas in last example a real data analysis is performed.

6.1 A first comparison of efficiency

In order to compare the performance of different techniques, in this section we consider a multi-modal, multidimensional Gaussian target density. More specifically, we have

$$\bar{\pi}(\boldsymbol{\theta}) = \frac{1}{3} \sum_{i=1}^{3} \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \qquad \boldsymbol{\theta} \in \mathbb{R}^D, \tag{48}$$

where $\boldsymbol{\mu}_1 = [\mu_{1,1}, \ldots, \mu_{1,D}]^\top$, $\boldsymbol{\mu}_2 = [\mu_{2,1}, \ldots, \mu_{2,D}]^\top$, $\boldsymbol{\mu}_3 = [\mu_{3,1}, \ldots, \mu_{3,D}]^\top$, with $\mu_{1,d} = -3$, $\mu_{2,d} = 0$, $\mu_{3,d} = 2$ for all $d = 1, \ldots, D$. Moreover, the covariance matrices are diagonal, $\boldsymbol{\Sigma}_i = \delta_i \mathbf{I}_D$ (where $\mathbf{I}_D$ is the $D \times D$ identity matrix), with $\delta_i = 0.5$ for $i = 1, 2, 3$. Hence, given a random variable $\boldsymbol{\Theta} \sim \bar{\pi}(\boldsymbol{\theta})$, we know analytically that $E[\boldsymbol{\Theta}] = [\bar{\theta}_1, \ldots, \bar{\theta}_D]^\top$ with $\bar{\theta}_d = -\frac{1}{3}$ for all d, and $\text{diag}\{\text{Cov}[\boldsymbol{\Theta}]\} = [\xi_1, \ldots, \xi_D]^\top$ with $\xi_d = \frac{85}{18}$ for all $d = 1, \ldots, D$.

We apply I-MTM, IMTM2 and I-EnMCMC in order to estimate all the expected values and all the variances of the marginal target pdfs. Namely, for a given dimension D, we have to estimate all $\{\bar{\theta}_d\}_{d=1}^{D}$ and $\{\xi_d\}_{d=1}^{D}$, hence $2D$ values. The results are averaged over 3000 independent runs. At each run, we compute an averaged square error obtained in the estimation of the $2D$ values and then calculate the Mean Square Error (MSE) averaged over the 3000 runs. For all the techniques, we consider a Gaussian proposal density $q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \sigma^2 \mathbf{I}_D)$ with $\boldsymbol{\mu} = [\mu_1 = 0, \ldots, \mu_D = 0]^\top$ (independent from the previous state) and different values of σ are considered.

6.1.1 Description of the experiments

We perform several experiments varying the number of tries, N, the length of the generated chain, T, the dimension of the inference problem, D, and the scale parameter of the proposal pdf, σ. In

Figures 7(a)-(b)-(c)-(d)-(e), we show the MSE (obtained by different techniques) as function of N, T, D and σ, respectively. In Figure 7(d), we only consider I-MTM with $N \in \{1, 5, 100, 1000\}$ in order to show the effect of using different tries N in different dimensions D. Note that I-MTM with $N = 1$ coincides with I-MH.

Let us denote as $\phi(\tau)$ the auto-correlation function of the states of the generated chain. Figures 8(a)-(b)-(c) depicts the normalized auto-correlation function $\bar{\phi}(\tau) = \frac{\phi(\tau)}{\phi(0)}$ (recall that $\phi(0) \geq \phi(\tau)$ for $\tau \geq 0$) at different lags $\tau = 1, 2, 3$, respectively. Furthermore, given the definition of the Effective Sample Size (ESS) [51, Chapter 4],

$$ESS = \frac{T}{1 + 2 \sum_{\tau=1}^{\infty} \bar{\phi}(\tau)}, \tag{49}$$

in Figure 8(d), we show of the ratio $\frac{ESS}{T}$ (approximated; cutting off the series in the denominator at lag $\tau = 10$), as function of N.[6] Finally, in Figures 9(a)-(b), we provide the Acceptance Rate (AR) of a new state (i.e., the expected number of accepted jumps to a novel state), as function of N and D, respectively.

6.1.2 Comment on the results

Figures 7(a)-(b), 8 and 9(a), clearly show that the performance improves as N grows, for all the algorithms. The MSE values and the correlation decrease, and the ESS and the AR grow. I-MTM seems to provide the best performance. Recall that for $N = 1$, I-EnMCMC becomes an I-MH with Baker's acceptance function and I-MTM becomes the I-MH in Table 4 [47]. For $N = 1$, the results confirm the Peskun's ordering about the acceptance function for a MH method [50, 47]. Observing the results, the Peskun's ordering appears valid also for the multiple try case, $N > 1$. I-MTM2 seems to have worse performance than I-MTM for all N. With respect to I-EnMCMC, I-MTM2 performs better for smaller N. The difference among the MSE values obtained by the samplers becomes smaller as N grows, as shown in Figure 7(a)-(b) (note that in the first one $D = 1$, in the other $D = 10$, and the range of N is different). The comparison among I-MTM, I-MTM2 and I-EnMCMC seems not to be affected by changing T and σ, as depicted in Figures 7(c)-(e). Namely, the MSE values change but the ordering of the methods (e.g., best and worst) seems to depend mainly on N. Obviously, for greater D, more tries are required in order to obtain good performance (see Figures 7(d) and 9(b), for instance). Note that for $N \to \infty$, I-MTM, I-MTM2 and I-EnMCMC perform similarly to an exact sampler drawing T independent samples from $\bar{\bar{\pi}}(\boldsymbol{\theta})$: the correlation $\phi(\tau)$ among the samples approaches zero (for all τ), ESS approaches T and AR approaches 1.

---

[6]Since the MCMC algorithms yield *positive* correlated sequences of states, we have $ESS < T$ in general.

(a) MSE versus N ($T = 2000$, $\sigma^2 = 2$, (b) MSE versus N ($T = 500$, $\sigma^2 = 2$, (c) MSE versus T ($N = 5$, $\sigma^2 = 2$, $D = 1$)
$D = 1$) $D = 10$)

(d) MSE versus D ($T = 500$, $\sigma^2 = 2$)   (e) MSE versus σ ($N = 5$, $T = 100$, $D = 1$)
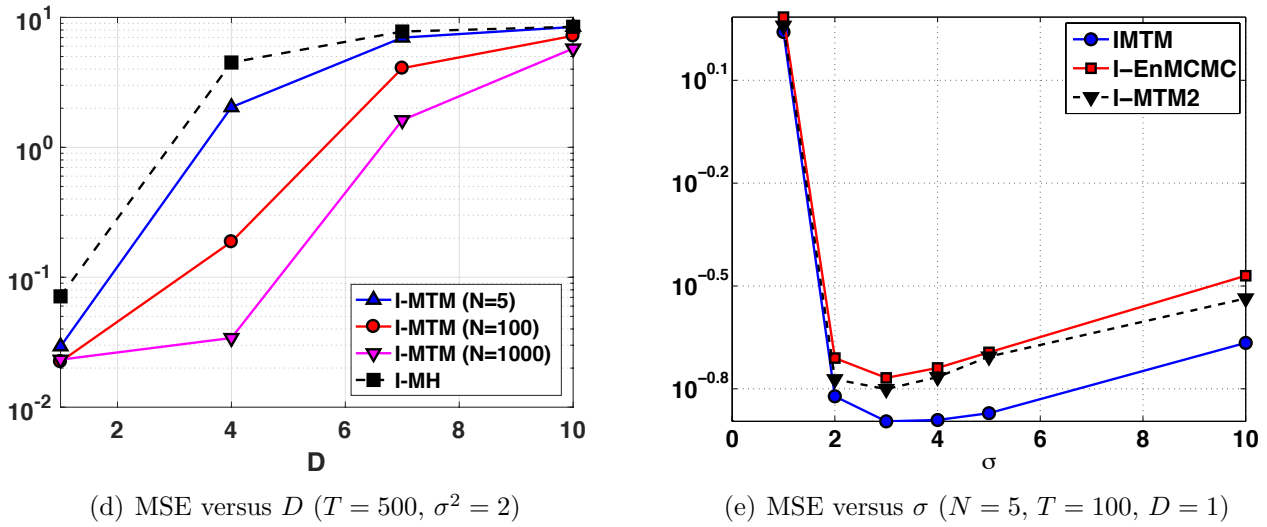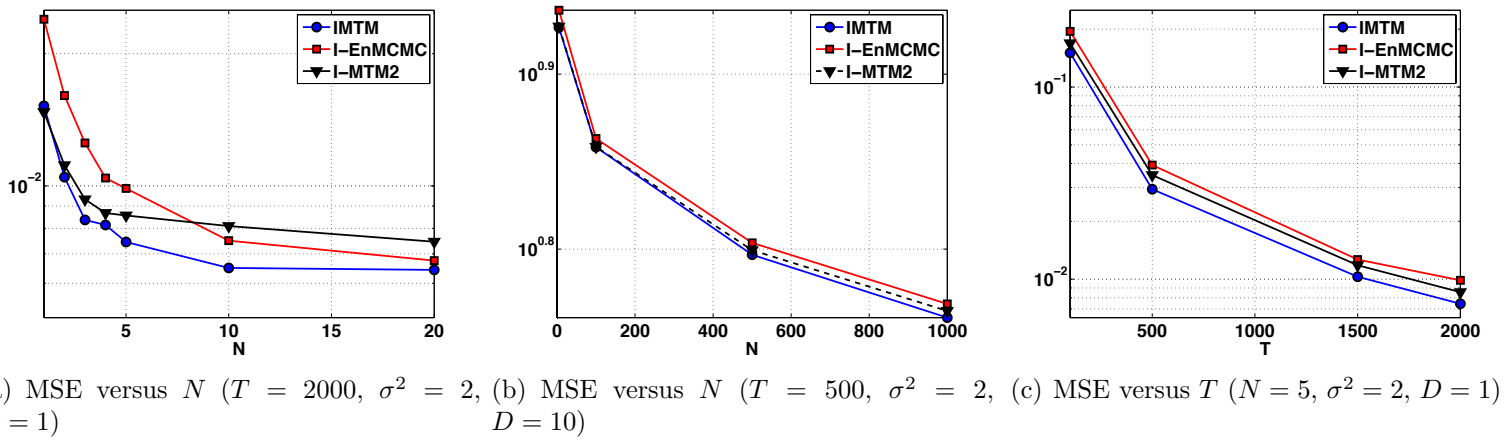
Figure 7: MSE as functions of different parameters (log-scale in the vertical axis). **(a)-(b)** MSE versus N with $T \in \{2000, 500\}$, $\sigma^2 = 2$, $D \in \{1, 10\}$. **(c)** MSE versus T with $N = 5$, $\sigma^2 = 2$, $D = 1$. **(d)** MSE versus D with $N \in \{1, 5, 10^2, 10^3\}$, $T = 2000$, $\sigma^2 = 2$ testing only for I-MTM (for $N = 1$, it coincides with I-MH). **(e)** MSE versus σ with $N = 5$, $T = 100$, $D = 1$.

6.2 Numerical experiment comparing particle schemes

In this section, in order to clarify the differences between batch and particle schemes, we consider again a multidimensional Gaussian target density, that can be express as

$$\bar{\pi}(\boldsymbol{\theta}) = \bar{\pi}(\theta_1, \ldots, \theta_D) = \prod_{d=1}^{D} \mathcal{N}(\theta_d | \mu_d, \sigma^2), \tag{50}$$
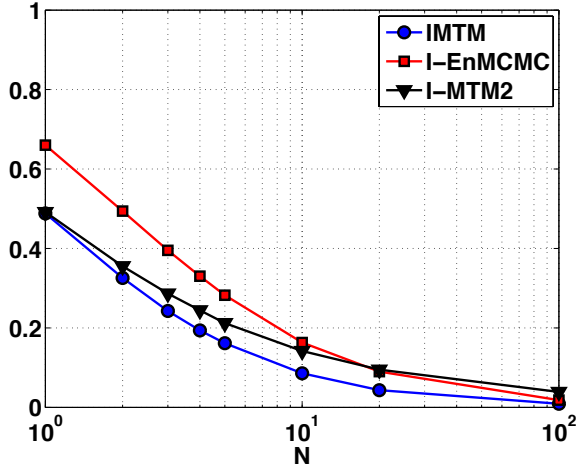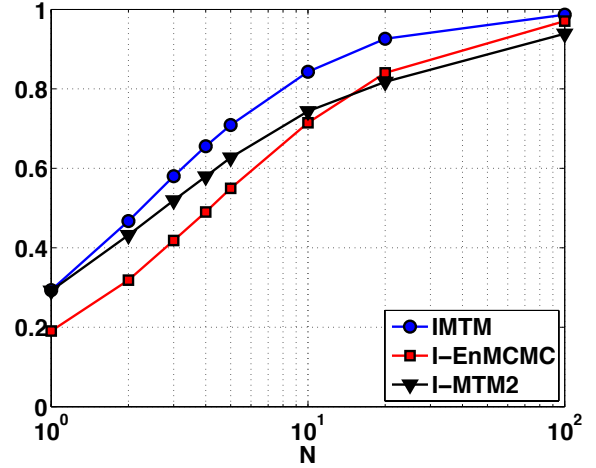
(a) $\bar{\phi}(1)$ versus N ($T = 2000$, $\sigma = 2$, $D = 1$).
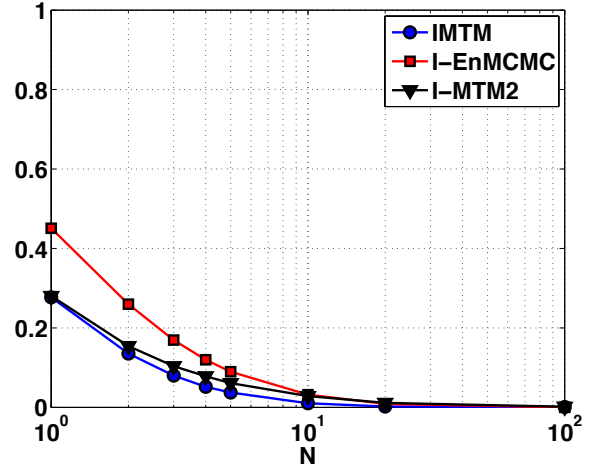
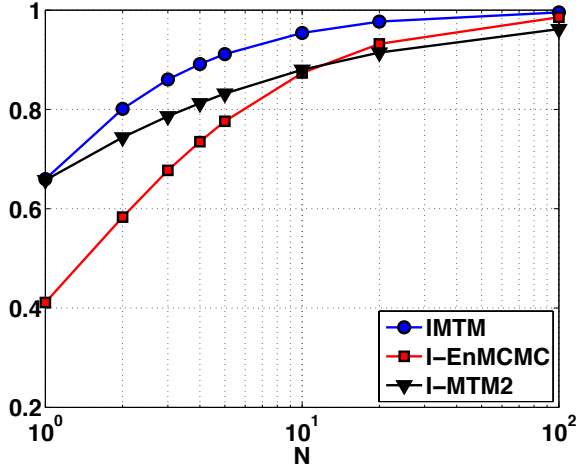(b) $\bar{\phi}(2)$ versus N ($T = 2000$, $\sigma = 2$, $D = 1$).
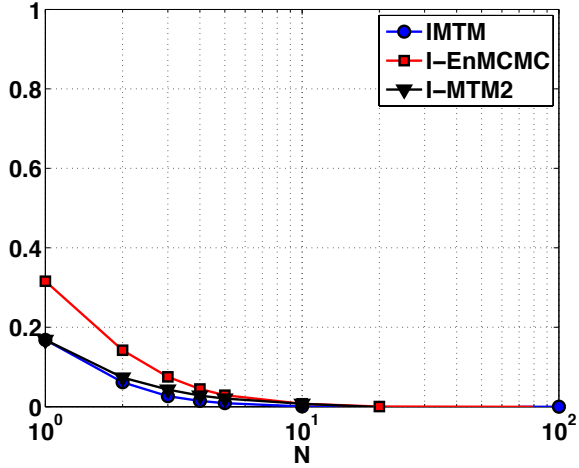
(c) $\bar{\phi}(3)$ versus N ($T = 2000$, $\sigma = 2$, $D = 1$).

(d) $\frac{ESS}{T}$ versus N ($T = 2000$, $\sigma = 2$, $D = 1$).

Figure 8: (log-scale in the horizontal axis) (**a**)-(**b**)-(**c**) Auto-correlation function $\bar{\phi}(\tau)$ of the states of generated chain as function of N, at different lags $\tau = 1, 2, 3$, respectively ($T = 2000$, $\sigma = 2$, $D = 1$). (**d**) The Effective Sample Size rate $\frac{ESS}{T}$ as function of N, keeping fixed $T = 2000$, $\sigma = 2$, $D = 1$.

with $\boldsymbol{\theta} = \theta_{1:D} \in \mathbb{R}^D$, $D = 10$, with $\mu_{1:3} = 2$, $\mu_{4:7} = 4$, $\mu_{8:10} = -1$ (shown dashed line in Figure 10(d)), and $\sigma = \frac{1}{2}$. The target pdf $\bar{\pi}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \sigma^2 \mathbf{I}_D)$ (with $\boldsymbol{\mu} = \mu_{1:10}$) is formed by independent Gaussian pieces, and can factored as in Eq. (50). The factorization allows the use of a sequential proposal procedure and hence particle schemes can be employed, even if there is not correlation

(a) AR versus N ($T = 2000$, $\sigma^2 = 2$, $D = 1$).
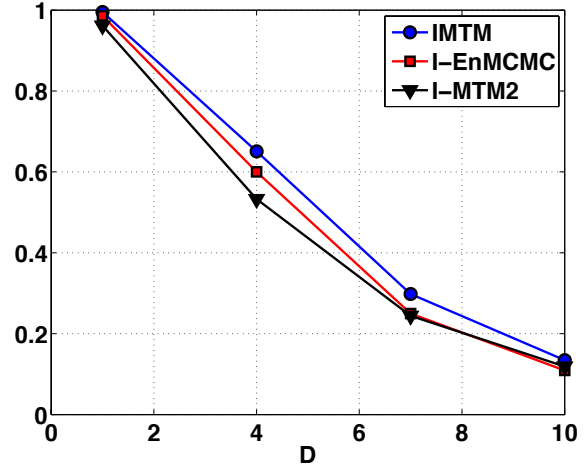
(b) AR versus D ($N = 100$, $T = 2000$, $\sigma^2 = 2$).

Figure 9: **(a)** Acceptance Rate (AR) as function of N, with $T = 2000$, $\sigma^2 = 2$, $D = 1$ (log-scale in the horizontal axis). **(b)** AR as function of N, with $N = 100$, $T = 2000$, $\sigma^2 = 2$.

between θ_d and θ_{d-1}.[7]

We apply I-MTM, I-MTM2, PMH, and a variant of PMH, denote as var-PMH which uses the corresponding acceptance probability of I-MTM in Eq. (19) instead of the acceptance function of the classical PMH in Eq. (30). The goal is to estimate the vector $\boldsymbol{\mu}$. We compute the MSE in estimating the vector $\boldsymbol{\mu} = \mu_{1:10}$, averaging the results over 500 independent simulations. The components of $\boldsymbol{\mu}$ are shown in Figure 10(d) with a dashed line.

For all the techniques, we employ a sequential construction of the N candidates (using the chain rule, see below): in PMH and var-PMH the resampling is applied at each iteration whereas in I-MTM and I-MTM2 no resampling is applied. More specifically, the proposal density for all the methods is

$$q(\boldsymbol{\theta}) = q_1(\theta_1) \prod_{d=2}^{D} q_d(\theta_d | \theta_{d-1}), \tag{51}$$

where $q_1(\theta_1) = \mathcal{N}(\theta_1 | -2, 4)$ and $q_d(\theta_d | \theta_{d-1}) = \mathcal{N}(\theta_d | \theta_{d-1}, \sigma_p^2)$, but PMH and var-PMH employ resampling steps so that the generated tries are correlated (whereas in I-MTM and I-MTM2 the generated candidates are independent).

We test all the techniques considering different value of number of tries N and number of iterations of the chain T. Figures 10(a)-(b) show the MSE as function of number of iterations T, keeping fixed the number of tries $N = 3$. Figure 10(a) reports the results of the MTM schemes whereas Figure 10(b) reports the results of the PMH schemes. Figure 10(c) depicts the MSE as function of N (with $T = 2000$), for the PMH methods. Note that the use of only $N = 3$

---

[7]For the particle schemes, we consider $\boldsymbol{\theta} = \mathbf{x}$ as a dynamic parameter (see Section 2).

particles and the application of the resampling at each iteration is clearly a disadvantage for the PMH schemes. If the resampling is applied very often (as in this case), a greater number of N is advisable (such as $N = 100$ or $N = 1000$). Hence, the results confirm that applying a resampling step at each iteration is not optimal and that a smaller rate of resampling steps could improve the performance [6, 52]. The results also confirm that the use of an acceptance probability of type in Eq. (19) provides smaller MSE, i.e., I-MTM and var-PMH perform better than I-MTM2 and PMH, respectively. This is more evident for small number of candidates N. When N grows, the performance of PMH and var-PMH methods becomes similar, since the acceptance probability approaches 1, in both cases. Figure 10(d) depicts 35 different states $\boldsymbol{\theta}_t = \theta_{1:10,t}$ at different iteration indices t, obtained with var-PMH ($N = 1000$ and $T = 1000$) and the values $\mu_{1:10}$ are given in dashed line.

## 6.3 Hyperparameter tuning for Gaussian Process (GP) regression models

Let us assume observed data pairs $\{y_j, \mathbf{z}_j\}_{j=1}^P$, with $y_j \in \mathbb{R}$ and $\mathbf{z}_j \in \mathbb{R}^L$. We also denote the corresponding $P \times 1$ output vector as $\mathbf{y} = [y_1, \ldots, y_P]^\top$ and the $L \times P$ input matrix as $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_P]$. We address the regression problem of inferring the unknown function f which links the variable y and $\mathbf{z}$. Thus, the assumed model is $y = f(\mathbf{z}) + e$, where $e \sim N(e; 0, \sigma^2)$, and that $f(\mathbf{z})$ is a realization of a Gaussian Process (GP) [53, 54]. Hence $f(\mathbf{z}) \sim \mathcal{GP}(\mu(\mathbf{z}), \kappa(\mathbf{z}, \mathbf{r}))$ where $\mu(\mathbf{z}) = 0$, $\mathbf{z}, \mathbf{r} \in \mathbb{R}^L$, and we consider the kernel function

$$\kappa(\mathbf{z}, \mathbf{r}) = \exp\left(-\sum_{\ell=1}^L \frac{(z_\ell - r_\ell)^2}{2\delta^2}\right). \tag{52}$$

Given these assumptions, the vector $\mathbf{f} = [f(\mathbf{z}_1), \ldots, f(\mathbf{z}_P)]^\top$ is distributed as $p(\mathbf{f}|\mathbf{Z}, \delta, \kappa) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{K})$, where $\mathbf{0}$ is a $P \times 1$ null vector, and $\mathbf{K}_{ij} := \kappa(\mathbf{z}_i, \mathbf{z}_j)$, for all $i, j = 1, \ldots, P$, is a $P \times P$ matrix. The vector containing all the hyperparameters of the model is $\boldsymbol{\theta} = [\delta, \sigma]$, i.e., all the parameters of the kernel function in Eq. (52) and standard deviation σ of the observation noise. In this experiment, we focus on the marginal posterior density of the hyperparameters, $\bar{\pi}(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa) \propto \pi(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa) = p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{Z}, \kappa)p(\boldsymbol{\theta})$, which can be evaluated analytically, but we cannot compute integrals involving it [54]. Considering a uniform prior within $[0, 20]^2$, $p(\mathbf{x})$ and since $p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{Z}, \kappa) = \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I})$, we have

$$\log\left[\pi(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)\right] = -\frac{1}{2}\mathbf{y}^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log\left[\det\left(\mathbf{K} + \sigma^2\mathbf{I}\right)\right] + C,$$

where $C > 0$, and clearly $\mathbf{K}$ depends on δ [54]. The moments of this marginal posterior cannot be computed analytically. Then, in order to compute the Minimum Mean Square Error (MMSE) estimator $\widehat{\boldsymbol{\theta}} = [\widehat{\delta}, \widehat{\sigma}]$, i.e., the expected value $E[\boldsymbol{\Theta}]$ with $\boldsymbol{\Theta} \sim \bar{\pi}(\boldsymbol{\theta}|\mathbf{y}, \mathbf{Z}, \kappa)$, we approximate $E[\boldsymbol{\Theta}]$ via Monte Carlo quadrature. More specifically, we apply I-MTM2, GMS, a MH scheme with a longer chain and a static IS method. For all these methodologies, we consider the same number of target evaluations, denoted as E, in order to provide a fair comparison.
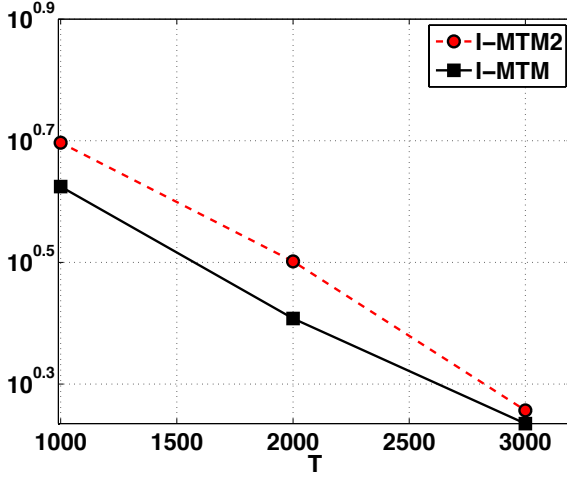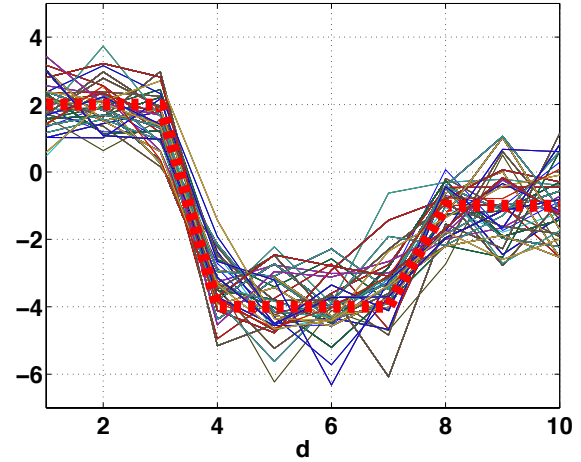
(a) MSE as function of T.
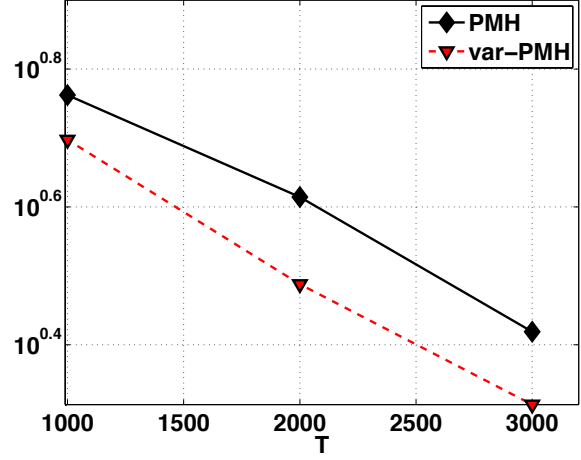
(b) MSE as function of T.
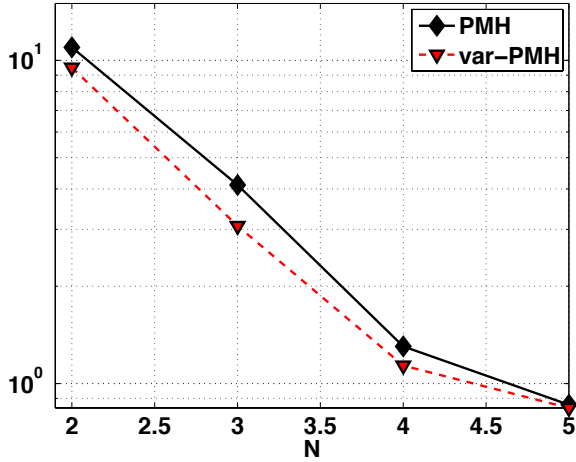
(c) MSE as function of N.

(d) Different states of the var-PMH chain.

Figure 10: **(a)**-**(b)** MSE (semi-log scale) as function of T ($N = 3$). **(c)** MSE (semi-log scale) as function of N ($T = 2000$). **(d)** Different states $\boldsymbol{\theta}_t = \theta_{1:10,t}$ at different iterations t, obtained with var-PMH ($N = 1000$ and $T = 1000$). The values $\boldsymbol{\mu} = \mu_{1:10}$ are shown in dashed line ($\mu_{1:3} = 2$, $\mu_{4:7} = 4$ and $\mu_{8:10} = -1$).

We generated $P = 200$ pairs of data, $\{y_j, \mathbf{z}_j\}_{j=1}^{P}$, according to the GP model above setting $\delta^* = 3$, $\sigma^* = 10$, $L = 1$, and drawing $z_j \sim \mathcal{U}([0, 10])$. We keep fixed these data over the different runs. We computed the ground-truth $\boldsymbol{\theta} = [\delta = 3.5200, \sigma = 9.2811]$ using an exhaustive and costly grid approximation, in order to compare the different techniques. For I-MTM2, GMS, and MH schemes, we consider the same adaptive Gaussian proposal pdf $q_t(\boldsymbol{\theta}|\boldsymbol{\mu}_t, \lambda^2 \mathbf{I}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_t, \lambda^2 \mathbf{I})$, with $\lambda = 5$ and $\boldsymbol{\mu}_t$ is adapted considering the arithmetic mean of the outputs after a training

30

period, $t \geq 0.2T$, in the same fashion of [55, 56] ($\boldsymbol{\mu}_0 = [1,1]^\top$). First, we test both techniques fixing $T = 20$ and varying the number of tries N. Then, we set $N = 100$ and vary the number of iterations T. Figure 11 (log-log plot) shows the Mean Square Error (MSE) in the approximation of $\widehat{\boldsymbol{\theta}}$ averaged over 10^3 independent runs. Observe that GMS always outperforms the corresponding I-MTM2 scheme. These results confirm the advantage of recycling the auxiliary samples drawn at each iteration during an I-MTM2 run. In Figure 12, we show the MSE obtained by GMS keeping invariant the number of target evaluations $E = NT = 10^3$ and varying $N \in \{1, 2, 10, 20, 50, 100, 250, 10^3\}$. As a consequence, we have $T \in \{10^3, 500, 100, 50, 20, 10, 4, 1\}$. Note that the case $N = 1$, $T = 10^3$, corresponds to an adaptive MH (A-MH) method with a longer chain, whereas the case $N = 10^3$, $T = 1$, corresponds to a static IS scheme (both with the same posterior evaluations $E = NT = 10^3$). We observe that the GMS always provides smaller MSE than the static IS approach. Moreover, GMS outperforms A-MH with the exception of two cases where $T \in \{1, 4\}$.
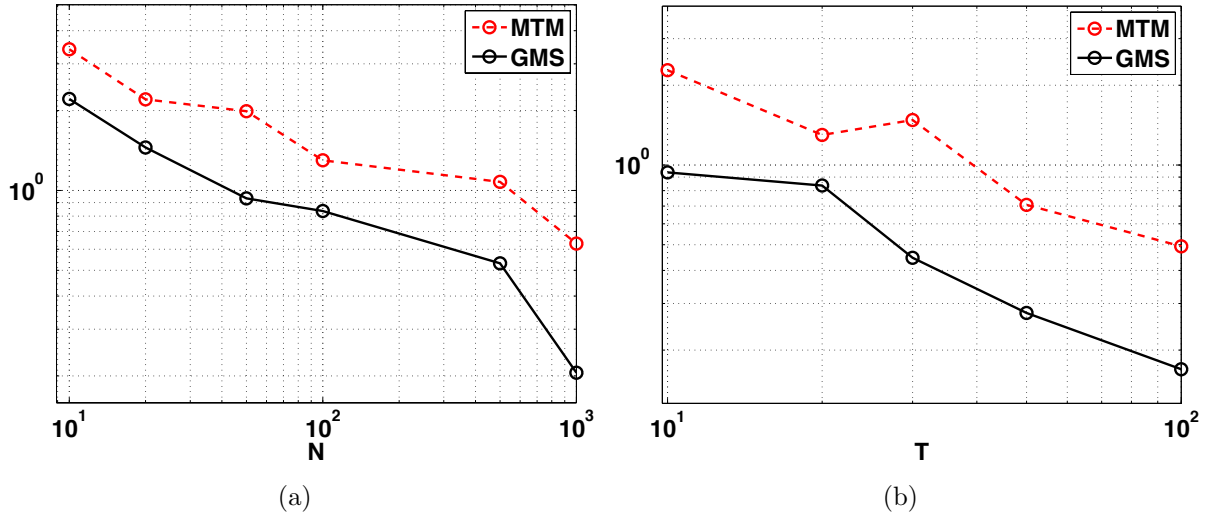

(a)   (b)

Figure 11: MSE (loglog-scale; averaged over 10^3 independent runs) obtained with the I-MTM2 and GMS algorithms (using the same proposal pdf and the same values of N and T) **(a)** as function of N with $T = 20$ and **(b)** as function of T with $N = 100$.

6.4 Localization of a target in a wireless sensor network

We consider the problem of positioning a target in $\mathbb{R}^2$ using a range measurements in a wireless sensor network (WSN) [57, 58]. We also assume that the measurements are contaminated by noise with different unknown power, one per each sensor. This situation is common in several practical scenarios. Indeed, even if the sensors have the same construction features, the noise perturbation of each the sensor can vary with the time and depends on the location of the sensor. This occurs owing to different causes: manufacturing defects, obstacles in the reception, different physical
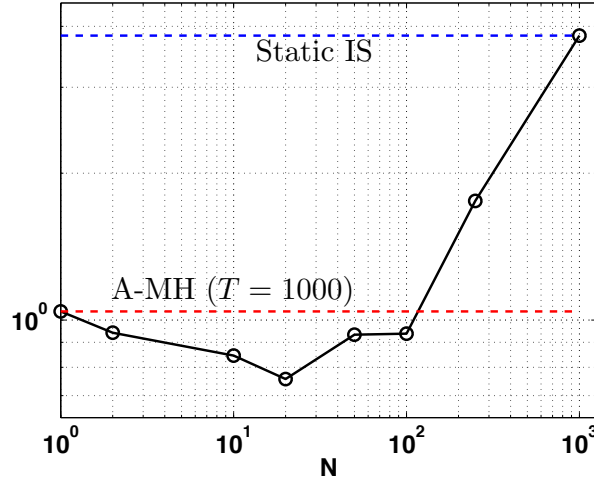
Figure 12: MSE (loglog-scale; averaged over 10^3 independent runs) of GMS (circles) versus the number of candidates $N \in \{1, 2, 10, 20, 50, 100, 250, 10^3\}$, but keeping fixed the total number of posterior evaluations $E = NT = 1000$, so that $T \in \{1000, 500, 100, 50, 20, 10, 4, 1\}$. The MSE values the extreme cases $N = 1$, $T = 1000$, and $N = 1000$, $T = 1$, are depicted with dashed lines. In first case, GMS coincides with an adaptive MH scheme (due the adaptation of the proposal, in this example) with a longer chain. The second one represents a static IS scheme (clearly, using the sample proposal than GMS). We can observe the benefit of the dynamic combination of IS estimators obtained by GMS.

environmental conditions (such as humidity and temperature) etc. Moreover, in general, these conditions change along time, hence it is necessary that the central node of the network is able to re-estimate the noise powers jointly with position of the target (and other parameters of the models if required) whenever a new block of observations is processed. More specifically, let us denote the target position with the random vector $\mathbf{Z} = [Z_1, Z_2]^\top$. The position of the target is then a specific realization $\mathbf{Z} = \mathbf{z}$. The range measurements are obtained from $N_S = 6$ sensors located at $\mathbf{h}_1 = [3, -8]^\top$, $\mathbf{h}_2 = [8, 10]^\top$, $\mathbf{h}_3 = [-4, -6]^\top$, $\mathbf{h}_4 = [-8, 1]^\top$, $\mathbf{h}_5 = [10, 0]^\top$ and $\mathbf{h}_6 = [0, 10]^\top$ as shown in Figure 13(a). The observation models are given by

$$Y_j = 20 \log \left(||\mathbf{z} - \mathbf{h}_j|| \right) + B_j, \quad j = 1, \ldots, N_S, \tag{53}$$

where B_j are independent Gaussian random variables with pdfs, $\mathcal{N}(b_j; 0, \zeta_j^2)$, $j = 1, \ldots, N_S$. We denote $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_{N_S}]$ the vector of standard deviations. Given the position of the target $\mathbf{z}^* = [z_1^* = 2.5, z_2^* = 2.5]^\top$ and setting $\boldsymbol{\zeta}^* = [\zeta_1^* = 1, \zeta_2^* = 2, \zeta_3^* = 1, \zeta_4^* = 0.5, \zeta_5^* = 3, \zeta_6^* = 0.2]$ (since $N_S = 6$ and $D = N_S + 2 = 8$), we generate $N_O = 20$ observations from each sensor according to the model in Eq. (53). Then, we finally obtain a measurement matrix $\mathbf{Y} = [y_{k,1}, \ldots, y_{k,N_S}] \in \mathbb{R}^{d_Y}$, where $d_Y = N_O N_S = 120$, $k = 1, \ldots, N_O$. We consider uniform prior $\mathcal{U}(\mathcal{R}_z)$ over the position $[z_1, z_2]^\top$ with $\mathcal{R}_z = [-30 \times 30]^2$, and a uniform prior over ζ_j, so that $\boldsymbol{\zeta}$ has prior $\mathcal{U}(\mathcal{R}_\zeta)$ with

32

$\mathcal{R}_\zeta = [0, 20]^{N_S}$. Thus, the posterior pdf is

$$\bar{\pi}(\boldsymbol{\theta}|\mathbf{Y}) = \bar{\pi}(\mathbf{z}, \boldsymbol{\zeta}|\mathbf{Y}) = \ell(\mathbf{y}|z_1, z_2, \zeta_1, \ldots, \zeta_{N_S}) \prod_{i=1}^{2} p(z_i) \prod_{j=1}^{N_S} p(\zeta_j),$$

$$= \left[\prod_{k=1}^{N_O} \prod_{j=1}^{N_S} \frac{1}{\sqrt{2\pi\zeta_j^2}} \exp\left(-\frac{1}{2\zeta_j^2} (y_{k,j} + 10\log(||\mathbf{z} - \mathbf{h}_j||)^2 \right) \right] \mathbb{I}_z(\mathcal{R}_z) \mathbf{I}_\zeta(\mathcal{R}_\zeta),$$

where $\boldsymbol{\theta} = [\mathbf{z}, \boldsymbol{\zeta}]^\top$ is a vector of parameters of dimension $D = N_S + 2 = 8$ that we desire to infer, and $\mathbb{I}_c(\mathcal{R})$ is an indicator variable that is 1 if $c \in \mathcal{R}$, otherwise is 0.

Our goal is to compute the Minimum Mean Square Error (MMSE) estimator, i.e., the expected value of the posterior $\bar{\pi}(\boldsymbol{\theta}|\mathbf{Y}) = \bar{\pi}(\mathbf{z}, \boldsymbol{\zeta}|\mathbf{Y})$ (recall that $D = 8$). Since the MMSE estimator cannot be computed analytically, we apply Monte Carlo methods for approximating it. We compare GMS, the corresponding MTM scheme, the Adaptive Multiple Importance Sampling (AMIS) technique [59], and N parallel MH chains with a random walk proposal pdf. For all of them we consider Gaussian proposal densities. For GMS and MTM, we set $q_t(\boldsymbol{\theta}|\boldsymbol{\mu}_{n,t}, \sigma^2\mathbf{I}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_t, \sigma^2\mathbf{I})$ which is adapted considering the empirical mean of the generated samples after a training period, $t \geq 0.2T$ [55, 56], $\boldsymbol{\mu}_0 \sim \mathcal{U}([1, 5]^D)$ and $\sigma = 1$. For AMIS, we have $q_t(\boldsymbol{\theta}|\boldsymbol{\mu}_t, \mathbf{C}_t) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_t, \mathbf{C}_t)$, where $\boldsymbol{\mu}_t$ is as previously described (with $\boldsymbol{\mu}_0 \sim \mathcal{U}([1, 5]^D)$) and $\mathbf{C}_t$ is also adapted using the empirical covariance matrix, starting $\mathbf{C}_0 = 4\mathbf{I}$. We also test the use of N parallel Metropolis-Hastings (MH) chains (we also consider the case of $N = 1$, i.e., a single chain), with a Gaussian random-walk proposal pdf, $q_n(\boldsymbol{\mu}_{n,t}|\boldsymbol{\mu}_{n,t-1}, \sigma^2\mathbf{I}) = \mathcal{N}(\boldsymbol{\mu}_{n,t}|\boldsymbol{\mu}_{n,t-1}, \sigma^2\mathbf{I})$ with $\boldsymbol{\mu}_{n,0} \sim \mathcal{U}([1, 5]^D)$ for all n and $\sigma = 1$.

We fix the total number of evaluations of the posterior density as $E = NT = 10^4$. Note that, generally, the evaluation of the posterior is the most costly step in MC algorithms (however, AMIS has the additional cost of re-weighting all the samples at each iteration according to the deterministic mixture procedure [60, 59]). We recall that T denotes the total number of iterations and N the number of samples drawn from each proposal at each iteration. We consider $\boldsymbol{\theta}^* = [\mathbf{z}^*, \boldsymbol{\zeta}^*]^\top$ as the ground-truth and compute the Mean Square Error (MSE) in the estimation obtained with the different algorithms. The results are averaged over 500 independent runs and they are provided in Tables 15, 16, and 17 and Figure 13(b). Note that GMS outperforms AMIS for each a pair $\{N, T\}$ (keeping fixed $E = NT = 10^4$), and GMS also provides smaller MSE values than N parallel MH chains (the case $N = 1$ corresponds to a unique longer chain). Figure 13(b) shows the MSE versus N maintaining $E = NT = 10^4$ for GMS and the corresponding MTM method. This figure again confirms the advantage of recycling the samples in a MTM scheme.

6.5 Localization with real data

In this section, we describe a numerical experiment involving real data. More specifically, we consider a localization problem [57]. We have carried out an experiment with a network consisting of four nodes. Three of them are placed at fixed positions and play the role of sensors that measure the strength of the radio signals transmitted by the target. The other node plays the role of the target to be localized. All nodes are bluetooth devices (Conceptronic CBT200U2A) with a nominal maximum range of 200 m. We consider a square monitored area of 4×4 m and place the sensors

Table 15: **Results GMS.**

MSE	1.30	1.24	1.22	1.21	1.22	**1.19**	1.31	**1.44**
N	10	20	50	100	200	500	1000	2000
T	1000	500	200	100	50	20	10	5
E	$NT = 10^4$							
MSE range	Min MSE= **1.19**			———		Max MSE= **1.44**		

Table 16: **Results AMIS [59].**

MSE	1.58	1.57	1.53	1.48	1.42	**1.29**	1.48	**1.71**
N	10	20	50	100	200	500	1000	2000
T	1000	500	200	100	50	20	10	5
E	$NT = 10^4$							
MSE range	Min MSE= **1.29**			———		Max MSE= **1.71**		

Table 17: **Results N parallel MH chains with random-walk proposal pdf.**

MSE	1.42	**1.31**	1.44	2.32	2.73	**3.21**	3.18	3.15
N	1	5	10	50	100	500	1000	2000
T	10^4	2000	1000	200	100	20	10	5
E	$NT = 10^4$							
MSE range	Min MSE= **1.31**			———		Max MSE=**3.21**		

at fixed positions $\mathbf{h}_1 = [0.5, 1]$, $\mathbf{h}_2 = [3.5, 1]$ and $\mathbf{h}_3 = [2, 3]$, with all coordinates in meters. The target is located at $\mathbf{z} = [z_1 = 2.5, z_2 = 2]$. The measurement provided by the i-th sensor is denoted as a random variable Y_i, considering the following model

$$Y_i = \kappa - 20 \log [||\mathbf{z} - \mathbf{h}_i||] + B_i, \tag{54}$$

where B_i are again independent Gaussian random variables with pdfs $\mathcal{N}(b_i; 0, \zeta^2)$, for all $i = 1, 2, 3$. Differently from the previous section, we estimate in advance the following parameters of the model, $\hat{\kappa} \approx -26.58$ and $\hat{\zeta} \approx 4.73$, using a least square fitting. We obtain $N_O = 5$ measurements from each sensor ($d_Y = 3N_O = 15$), and we consider a uniform prior on the 4×4 m area. Given these measurements, we approximate the expected value $E[\mathbf{Z}]$ of the corresponding posterior $\bar{\pi}(\mathbf{z})$ (here $\boldsymbol{\theta} = \mathbf{z}$) using a thin deterministic bivariate grid, obtaining the ground truth $\approx [3.17, 2.62]^\top$. We test an MH method and MTM scheme using with a random walk Gaussian proposal pdf, $q(\mathbf{z}|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}|\mathbf{z}_{t-1}, \sigma^2 \mathbf{I}_2)$, with $\sigma = 1$, $T \in \{1000, 5000\}$, and $N \in \{10, 100, 1000\}$ tries for MTM (clearly, $N = 1$ for MH). We also test a Metropolis-adjusted Langevin algorithm (MALA), where the proposal is Gaussian random walk density with mean $\mathbf{z}_{t-1} + \beta \nabla[\log \pi(\mathbf{z})]$ and $\nabla[\log \pi(\mathbf{z})]$ denotes the gradient of $\log \pi(\mathbf{z})$ [61]. The covariance matrix of MALA Gaussian proposal is $\sigma^2 \mathbf{I}_2$ (as the other techniques) and the drift parameter $\beta = \sigma^2/2$. We compute the MSE in estimating
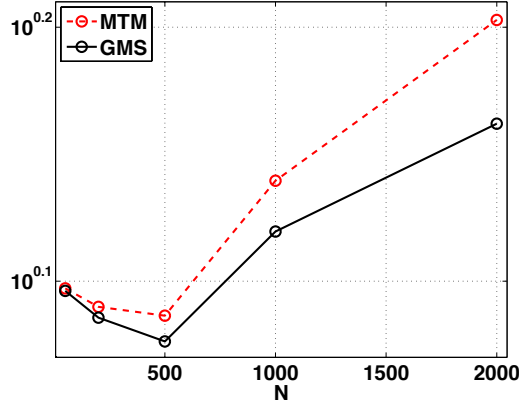
(a)

Figure 13: MSE (log-scale) versus the number of candidates $N \in \{50, 200, 500, 1000, 2000\}$ obtained by GMS and the corresponding I-MTM algorithm (without using all the samples of the accepted sets, but only a resampled one), keeping fixed the total number of evaluations $E = NT = 10^4$ of the posterior pdf, so that $T \in \{200, 50, 20, 10, 5\}$.

$E[\mathbf{Z}] \approx [3.17, 2.62]^{\top}$ and averaged the results over 2000 independent runs (at each run, we take the mean of the square error values of each component). The results are shown in Table 18. Recall that MALA uses the additional information of the gradient. Note that, a MALA-type proposal pdf can also be used in a MTM scheme. The use of multiple tries improves the mixing of the Markov chain and speeds up the convergence.

Table 18: **MSE obtained in the localization problem with real data.**

Algorithm	MH	MALA	MALA	MTM	MTM	MTM
N	1	1	1	10	100	1000
T	1000	1000	5000	1000	1000	1000
MSE	0.2511	0.2081	0.1359	0.0944	0.0469	0.0030

7   Conclusions

We have provided a thorough review of MCMC methods using multiple candidates in order to select the next state of the chain. We have presented and compared different Multiple Try Metropolis, Ensemble MCMC and Delayed Rejection Metropolis schemes. We have also described the Group Metropolis Sampling technique which generates a chain of set of weighted samples, so that some candidates are properly reused in the final estimators. Furthermore, we have shown how the Particle Metropolis-Hastings algorithm can be interpreted as an MTM scheme using a particle filter for generating the different weighted candidates. Several connections and differences have

35

been pointed out. Finally, we have tested several techniques in different numerical experiments: two toy examples in order to provide an exhaustive comparison among the methods, a numerical example regarding the hyperparameter selection for a Gaussian Process (GP) regression model, and two localization problems, one of them involving a real data analysis.

Acknowledgements

References

[1] A. Doucet, X. Wang, Monte Carlo methods for signal processing, IEEE Signal Processing Magazine 22 (6) (2005) 152–170.

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

[3] L. Martino, J. Míguez, A novel rejection sampling scheme for posterior probability distributions, Proc. of the 34th IEEE ICASSP (2009) 1–5.

[4] L. Martino, H. Yang, D. Luengo, J. Kanniainen, J. Corander, A fast universal self-tuned sampler within Gibbs sampling, Digital Signal Processing 47 (2015) 68 – 83.

[5] M. F. Bugallo, S. Xu, P. M. Djurić, Performance comparison of EKF and particle filtering methods for maneuvering targets, Digital Signal Processing 17 (2007) 774–786.

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

[7] F. Liang, C. Liu, R. Caroll, Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples, Wiley Series in Computational Statistics, England, 2010.

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

[9] C. P. Robert, G. Casella, Monte Carlo Statistical Methods, Springer, 2004.

[10] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1) (1970) 97–109.

[11] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equations of state calculations by fast computing machines, Journal of Chemical Physics 21 (1953) 1087–1091.

[12] D. Frenkel, B. Smit, Understanding molecular simulation: from algorithms to applications, Academic Press, San Diego.

[13] J. S. Liu, F. Liang, W. H. Wong, The multiple-try method and local optimization in metropolis sampling, Journal of the American Statistical Association 95 (449) (2000) 121–134.

[14] L. Martino, V. P. D. Olmo, J. Read, A multi-point Metropolis scheme with generic weight functions, Statistics & Probability Letters 82 (7) (2012) 1445–1453.

[15] M. Bédard, R. Douc, E. Mouline, Scaling analysis of multiple-try MCMC methods, Stochastic Processes and their Applications 122 (2012) 758–786.

[16] R. V. Craiu, C. Lemieux, Acceleration of the Multiple Try Metropolis algorithm using antithetic and stratified sampling, Statistics and Computing 17 (2) (2007) 109–120.

[17] R. Casarin, R. V. Craiu, F. Leisen, Interacting multiple try algorithms with different proposal distributions, Statistics and Computing 23 (2) (2013) 185–200.

[18] S. Pandolfi, F. Bartolucci, N.Friel, A generalization of the Multiple-try Metropolis algorithm for Bayesian estimation and model selection, Journal of Machine Learning Research (Workshop and Conference Proceedings Volume 9: AISTATS 2010) 9 (2010) 581–588.

[19] R. Neal, MCMC using ensembles of states for problems with fast and slow variables such as Gaussian process regression, arXiv:1101.0387, 2011.

[20] B. Calderhead, A general construction for parallelizing Metropolis-Hastings algorithms, Proceedings of the National Academy of Sciences of the United States of America (PNAS) 111 (49) (2014) 17408–17413.

[21] E. Bernton, S. Yang, Y. Chen, N. Shephard, J. S. Liu, Locally weighted Markov Chain Monte Carlo, arXiv:1506.08852 (2015) 1–14.

[22] H. Austad, Parallel multiple proposal mcmc algorithms, Master thesis, Norwegian University (2007) 1–44.

[23] H. Haario, M. Laine, A. Mira, E. Saksman, DRAM: efficient adaptive MCMC, Statistics and Computing 16 (4) (2006) 339–354.

[24] A. Mira, On Metropolis-Hastings algorithms with delayed rejection, Metron, Vol. LIX (3-4) (2001) 231–241.

[25] L. Tierney, A. Mira, Some adaptive Monte Carlo methods for Bayesian inference, Statistics in Medicine 18 (1999) 2507–2515.

[26] C. Andrieu, A. Doucet, R. Holenstein, Particle Markov chain Monte Carlo methods, J. R. Statist. Soc. B 72 (3) (2010) 269–342.

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

[28] L. Martino, F. Leisen, J. Corander, On multiple try schemes and the Particle Metropolis-Hastings algorithm, viXra:1409.0051 (2014) 1–21.

[29] L. Martino, V. Elvira, G. Camps-Valls, Group Importance Sampling for particle filtering and MCMC, arXiv:1704.02771 (2017) 1–39.

[30] L. Martino, V. Elvira, G. Camps-Valls, Group Metropolis Sampling, European Signal Processing Conference (EUSIPCO) (2017) 1–5.

[31] S. C. Leman, Y. Chen, M. Lavine, The multiset sampler, Journal of the American Statistical Association 104 (487) (2009) 1029–1041.

[32] G. Storvik, On the flexibility of Metropolis-Hastings acceptance probabilities in auxiliary variable proposal generation, Scandinavian Journal of Statistics 38 (2) (2011) 342–358.

[33] J. M. Bernardo, A. F. M. Smith, Bayesian Theory, Wiley & sons, 1994.

[34] C. P. Robert, The Bayesian Choice, Springer, 2007.

[35] G. E. P. Box, G. C. Tiao, Bayesian Inference in Statistical Analysis, Wiley & sons, 1973.

[36] R. L. Burden, J. D. Faires, Numerical Analysis, Brooks Cole, 2000.

[37] P. K. Kythe, M. R. Schaferkotter, Handbook of Computational Methods for Integration, Chapman and Hall/CRC, 2004.

[38] L. Martino, J. Read, On the flexibility of the design of multiple try Metropolis schemes, Computational Statistics 28 (6) (2013) 2797–2823.

[39] L. Martino, F. Louzada, Issues in the Multiple Try Metropolis mixing, Computational Statistics 32 (1) (2017) 239–252.

[40] R. Casarin, R. V. Craiu, F. Leisen, Interacting multiple try algorithms with different proposal distributions., Statistics and Computing 23(2) (2013) 185–200.

[41] L. Martino, V. Elvira, F. Louzada, Weighting a resampled particle in Sequential Monte Carlo, IEEE Statistical Signal Processing Workshop, (SSP) 122 (2016) 1–5.

[42] L. Martino, V. Elvira, D. Luengo, J. Corander, F. Louzada, Orthogonal parallel MCMC methods for sampling and optimization, Digital Signal Processing 58 (2016) 64–84.

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

[44] J. I. Siepmann, D. Frenkel, Configurational bias Monte Carlo: a new sampling scheme for flexible chains, Molecular Physics 75 (1) (1992) 59–70.

[45] C. Andrieu, G. O. Roberts, The pseudo-marginal approach for efficient monte carlo computations, The Annals of Statistics 37 (2) (2009) 697–725.

[46] A. A. Barker, Monte Carlo calculation of the radial distribution functions for a protonelectron plasma, Austral. J Phys. 18 (1973) 119–133.

[47] L. Martino, V. Elvira, Metropolis Sampling, Wiley StatsRef: Statistics Reference Online.

[48] L. Martino, V. Elvira, D. Luengo, J. Corander, Layered adaptive importance sampling, Statistics and Computing 27 (3) (2017) 599–623.

[49] H. Tjelmeland, B. K. Hegstad, Mode jumping proposals in MCMC, Scandinavian Journal of Statistics 28 (1) (2001) 205–223.

[50] P. H. Peskun, Optimum Monte-Carlo sampling using Markov chains, Biometrika 60 (3) (1973) 607–612.

[51] D. Gamerman, Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Chapman and Hall/CRC, 1997.

[52] A. Doucet, A. M. Johansen, A tutorial on particle filtering and smoothing: fifteen years later, technical report.

[53] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[54] C. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.

[55] D. Luengo, L. Martino, Fully adaptive Gaussian mixture Metropolis-Hastings algorithm, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[56] H. Haario, E. Saksman, J. Tamminen, An adaptive Metropolis algorithm, Bernoulli 7 (2) (2001) 223–242.

[57] A. M. Ali, K. Yao, T. C. Collier, E. Taylor, D. Blumstein, L. Girod, An empirical study of collaborative acoustic source localization, Proc. Information Processing in Sensor Networks (IPSN07), Boston.

[58] A. T. Ihler, J. W. Fisher, R. L. Moses, A. S. Willsky, Nonparametric belief propagation for self-localization of sensor networks, IEEE Transactions on Selected Areas in Communications 23 (4) (2005) 809–819.

[59] J. M. Cornuet, J. M. Marin, A. Mira, C. P. Robert, Adaptive multiple importance sampling, Scandinavian Journal of Statistics 39 (4) (2012) 798–812.

[60] M. F. Bugallo, L. Martino, J. Corander, Adaptive importance sampling in signal processing, Digital Signal Processing 47 (2015) 36–49.

[61] G. O. Roberts, J. S. Rosenthal, Optimal scaling of discrete approximations to Langevin diffusions, Journal of the Royal Statistical Society. Series B (Statistical Methodology) 60 (1) (1998) 255–268.

[62] L. Martino, V. Elvira, F. Louzada, Effective Sample Size for importance sampling based on discrepancy measures, Signal Processing 131 (2017) 386–401.

[63] L. Martino, J. Read, V. Elvira, F. Louzada, Cooperative parallel particle filters for on-line model selection and applications to urban mobility, Digital Signal Processing 60 (2017) 172–185.

A Distribution after resampling

Let us also denote as $\boldsymbol{\theta} \in \{\boldsymbol{\theta}^{(1)} \ldots, \boldsymbol{\theta}^{(N)}\}$, a generic sample after applying one multinomial resampling step according to the normalized IS weights $\bar{w}_n$, $n = 1, \ldots, N$. The density of $\boldsymbol{\theta}$ is given by

$$\widetilde{q}(\boldsymbol{\theta}) = \int_{\mathcal{D}^N} \widehat{\pi}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(1:N)}) \left[\prod_{i=1}^N q(\boldsymbol{\theta}^{(i)}) \right] d\boldsymbol{\theta}^{(1:N)}, \tag{55}$$

where

$$\widehat{\pi}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(1:N)}) = \sum_{j=1}^N \bar{w}_j \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(j)}). \tag{56}$$

We also define also the matrix $\mathbf{m}_{\neg n} = [\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(n)}, \boldsymbol{\theta}^{(n+1)}, \ldots, \boldsymbol{\theta}^{(N)}]$, containing all the samples except for the n-th. After some straightforward rearrangements, Eq. (55) can be rewritten as

$$\widetilde{q}(\boldsymbol{\theta}) = \sum_{j=1}^N \left(\int_{\mathcal{D}^{N-1}} \frac{\pi(\boldsymbol{\theta})}{\sum_{n=1}^N \frac{\pi(\boldsymbol{\theta}^{(n)})}{q(\boldsymbol{\theta}^{(n)})}} \left[\prod_{\substack{n=1 \\ n \neq j}}^N q(\boldsymbol{\theta}^{(n)}) \right] d\mathbf{m}_{\neg j} \right). \tag{57}$$

Finally, we can write

$$\widetilde{q}(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}) \sum_{j=1}^N \int_{\mathcal{D}^{N-1}} \frac{1}{N\widehat{Z}} \left[\prod_{\substack{n=1 \\ n \neq j}}^N q(\boldsymbol{\theta}^{(n)}) \right] d\mathbf{m}_{\neg j}, \tag{58}$$

where $\widehat{Z} = \frac{1}{N} \sum_{n=1}^N \frac{\pi(\boldsymbol{\theta}^{(n)})}{q(\boldsymbol{\theta}^{(n)})}$ that is that IS estimator of Z. The equation above represents the density of a resampled particle $\boldsymbol{\theta} \in \{\boldsymbol{\theta}^{(1)} \ldots, \boldsymbol{\theta}^{(N)}\}$. Note that if $\widehat{Z} = Z$ then $\widetilde{q}(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})$. Clearly, for a finite value of N, there exists a discrepancy between $\widetilde{q}(\boldsymbol{\theta})$ and $\bar{\pi}(\boldsymbol{\theta})$, but this discrepancy decreases as N grows.

B Particle Filtering

In this appendix, we recall some basic concepts and more recent results about the Sequential Importance Sampling (SIS) and Sequential Importance Resampling (SIR) methods, important for

the description of the techniques above.

SIS procedure. We assume that the variable of interest is formed by only a dynamical variable, i.e., $\boldsymbol{\theta} = \mathbf{x} = x_{1:D} = [x_1 \ldots, x_D]^\top$ (for simplicity, consider $x_d \in \mathbb{R}$) and the target can be factorized as

$$\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x}) \;=\; \gamma_1(x_1) \prod_{d=2}^{D} \gamma_d(x_d|x_{d-1}). \tag{59}$$

Given a proposal of type $q(\mathbf{x}) = q_1(x_1) \prod_{d=2}^{D} q_d(x_d|x_{d-1})$, and a sample $\mathbf{x}^{(n)} = x_{1:D}^{(n)} \sim q(\mathbf{x})$ with $x_d^{(n)} \sim q_d(x_d|x_{d-1})$, we assign the importance weight

$$w(\mathbf{x}^{(n)}) = w_D^{(n)} = \frac{\pi(\mathbf{x}^{(n)})}{q(\mathbf{x}^{(n)})} = \frac{\gamma_1(x_1^{(n)})\gamma_2(x_2^{(n)}|x_1^{(n)})\cdots\gamma_D(x_D^{(n)}|x_{1:D-1}^{(n)})}{q_1(x_1^{(n)})q_2(x_2^{(n)}|x_1^{(n)})\cdots q_D(x_D^{(n)}|x_{1:D-1}^{(n)})}. \tag{60}$$

The weight above can be compute with a recursive procedure for computing the importance weights: starting with $w_1^{(n)} = \frac{\pi(x_1^{(n)})}{q(x_1^{(n)})}$ and then

$$w_d^{(n)} = w_{d-1}^{(n)}\beta_d^{(n)} = \prod_{j=1}^{d} \beta_j^{(n)}, \qquad d = 1, \ldots, D, \tag{61}$$

where we have set

$$\beta_1^{(n)} = w_1^{(n)} \quad \text{and} \quad \beta_d^{(n)} = \frac{\gamma_d(x_d^{(n)}|x_{1:d-1}^{(n)})}{q_d(x_d^{(n)}|x_{1:d-1}^{(n)})}, \tag{62}$$

for $d = 2, \ldots, D$. Let also define the partial target pdfs

$$\pi_d(x_{1:d}) = \gamma_1(x_1) \prod_{i=2}^{d} \gamma_i(x_i|x_{i-1}) \tag{63}$$

SIR procedure. In SIR, a.k.a., standard particle filtering, resampling steps are incorporated during the recursion as shown of Table 19 [6, 52]. In general, the resampling steps are applied only in certain iterations in order to avoid the path degeneration, taking into account an approximation $\widehat{ESS}$ of the Effective Sampling Size (ESS) [62]. If $\widehat{ESS}$ is smaller than a pre-established threshold, the particles are resampled. Two examples of ESS approximation are $\widehat{ESS} = \frac{1}{\sum_{n=1}^{N}(\bar{w}_d^{(n)})^2}$ and $\widehat{ESS} = \frac{1}{\max \bar{w}_d^{(n)}}$ where $\bar{w}_d^{(n)} = \frac{w_d^{(n)}}{\sum_{i=1}^{N} w_d^{(i)}}$ (note that $1 \leq \widehat{ESS} \leq N$). Hence, the condition for the adaptive resampling can be expressed as $\widehat{ESS} < \eta N$ where $\eta \in [0,1]$. SIS is given when $\eta = 0$ and SIR for $\eta \in (0,1]$. When $\eta = 1$, the resampling is applied at each iteration and in this case SIR is often called *bootstrap particle filter* [6, 52]. If $\eta = 0$, no resampling steps are applied, and we have the SIS method described above.

Remark 6. *Note that in Table 19, we have employed a proper weighting for resampling particles [41],*

$$w_d^{(1)} = w_d^{(2)} = \ldots = w_d^{(n)} = \widehat{Z}_d. \tag{64}$$

Generally, it is remarked that $w_d^{(1)} = w_d^{(2)} = \ldots = w_d^{(n)}$ but a specific value is not given. If a different value $c \neq \widehat{Z}_d$ is employed, i.e., $w_d^{(1)} = \ldots = w_d^{(n)} = c$, the algorithm is still valid but the weight recursion loses part of the statistical meaning. This is the reason why the marginal likelihood estimator $\widehat{Z} = \widehat{Z}_D = \frac{1}{N} \sum_{n=1}^{N} w_D^{(n)}$ is consistent only, if a proper weighting after resampling is used [41, 29, 63].

Table 19: **The SIR method with proper weighting after resampling.**

---

1. Choose N the number of particles, the initial particles $x_0^{(n)}$, $n = 1, \ldots, N$, an ESS approximation $\widehat{ESS}$ [62] and a constant value $\eta \in [0, 1]$.

2. For $d = 1, \ldots, D$:

   (a) **Propagation:** Draw $x_d^{(n)} \sim q_d(x_d | x_{d-1}^{(n)})$, for $n = 1, \ldots, N$.

   (b) **Weighting:** Compute the weights

   $$w_d^{(n)} = w_{d-1}^{(n)} \beta_d^{(n)} = \prod_{j=1}^{d} \beta_j^{(n)}, \qquad n = 1, \ldots, N, \tag{65}$$

   where $\beta_d^{(n)} = \frac{\gamma_d(x_d^{(n)} | x_{d-1}^{(n)})}{q_d(x_d^{(n)} | x_{d-1}^{(n)})}$.

   (c) if $\widehat{ESS} < \eta N$ then:

      i. **Resampling:** Resample N times within the set $\{x_{d-1}^{(n)}\}_{n=1}^{N}$ according to the probabilities $\bar{w}_d^{(n)} = \frac{w_d^{(n)}}{\sum_{j=1}^{N} w_d^{(j)}}$, obtaining N resampled particles $\{\bar{x}_d^{(n)}\}_{n=1}^{N}$. Then, set $x_d^{(n)} = \bar{x}_d^{(n)}$, for $n = 1, \ldots, N$.

      ii. **Proper weighting:** Compute $\widehat{Z}_d = \frac{1}{N} \sum_{n=1}^{N} w_d^{(n)}$ and set $w_d^{(n)} = \widehat{Z}_d$ for all $n = 1, \ldots, N$.

3. Return $\{\mathbf{x}_n = x_{1:D}^{(n)}, w_n = w_D^{(n)}\}_{n=1}^{N}$.

B.1 Marginal likelihood estimators in SIS

Remark 7. *In SIS, there are two possible formulations of the estimator of the marginal likelihoods* $Z_d = \int_{\mathbb{R}^d} \pi_d(x_{1:d}) dx_{1:d}$,

$$\widehat{Z}_d = \frac{1}{N} \sum_{n=1}^{N} w_d^{(n)} = \frac{1}{N} \sum_{n=1}^{N} w_{d-1}^{(n)} \beta_d^{(n)}, \tag{66}$$

$$\overline{Z}_d = \prod_{j=1}^{d} \left[\sum_{n=1}^{N} \bar{w}_{j-1}^{(n)} \beta_j^{(n)} \right]. \tag{67}$$

In SIS, both estimators are equivalent $\overline{Z}_d \equiv \widehat{Z}_d$.

Indeed, the classical IS estimator of the normalizing constant Z_d at the d-th iteration is

$$\widehat{Z}_d = \frac{1}{N} \sum_{n=1}^{N} w_d^{(n)} = \frac{1}{N} \sum_{n=1}^{N} w_{d-1}^{(n)} \beta_d^{(n)}, \tag{68}$$

$$= \frac{1}{N} \sum_{n=1}^{N} \left[\prod_{j=1}^{d} \beta_j^{(n)} \right]. \tag{69}$$

An alternative formulation, denoted as $\overline{Z}_d$, is often used

$$\overline{Z}_d = \prod_{j=1}^{d} \left[\sum_{n=1}^{N} \bar{w}_{j-1}^{(n)} \beta_j^{(n)} \right] \tag{70}$$

$$= \prod_{j=1}^{d} \left[\frac{\sum_{n=1}^{N} w_j^{(n)}}{\sum_{n=1}^{N} w_{j-1}^{(n)}} \right] = \widehat{Z}_1 \prod_{j=2}^{d} \left[\frac{\widehat{Z}_j}{\widehat{Z}_{j-1}} \right] = \widehat{Z}_d. \tag{71}$$

where we have employed $\bar{w}_{j-1}^{(n)} = \frac{w_{j-1}^{(n)}}{\sum_{i=1}^{N} w_{j-1}^{(i)}}$ and $w_j^{(n)} = w_{j-1}^{(n)} \beta_j^{(n)}$ [52].

Furthermore, note that $\overline{Z}_d$ can be written in a recursive form as

$$\overline{Z}_d = \overline{Z}_{d-1} \left[\sum_{n=1}^{N} \bar{w}_{d-1}^{(n)} \beta_d^{(n)} \right]. \tag{72}$$

B.2 Marginal likelihood estimators in SIR

Remark 8. *If a proper weighting after resampling is applied in SIR, both formulations* $\widehat{Z}_d$ *and* $\overline{Z}_d$ *in Eqs.* (66)-(67) *provide consistent estimator of* Z_d *and they are equivalent,* $\widehat{Z}_d \equiv \overline{Z}_d$ *(as in SIS).*

If a proper weighting is not applied, only

$$\overline{Z}_d = \prod_{j=1}^{d} \left[\sum_{n=1}^{N} \bar{w}_{j-1}^{(n)} \beta_j^{(n)} \right].$$

is a consistent estimator of Z_d, in SIR. In this case, $\widehat{Z}_d = \frac{1}{N}\sum_{n=1}^{N} w_d^{(n)}$ is not a possible alternative (without using a proper weighting after resampling). However, considering the proper weighting of the resampled particles, then $\widehat{Z}_d$ is also a consistent estimator of Z_d and it is equivalent to $\overline{Z}_d$. Below, we analyze three cases:

- **No Resampling ($\eta = 0$):** this scenario corresponds to SIS where $\widehat{Z}_d, \overline{Z}_d$ are equivalent as shown in Eq. (71).

- **Resampling at each iteration ($\eta = 1$):** using the proper weighting, $w_{d-1}^{(n)} = \widehat{Z}_{d-1}$ for all n and for all d, and replacing in Eq. (68) we have

$$\widehat{Z}_d = \widehat{Z}_{d-1}\left[\frac{1}{N}\sum_{n=1}^{N}\beta_d^{(n)}\right], \tag{73}$$

$$= \frac{1}{N}\prod_{j=1}^{d}\left[\sum_{n=1}^{N}\beta_j^{(n)}\right]. \tag{74}$$

Since after resampling all particles have the same weight, we have $\bar{w}_{d-1}^{(n)} = \frac{1}{N}$ for all n. Replacing it in the expression of $\overline{Z}_d$ in (72), we obtain

$$\overline{Z}_d = \frac{1}{N}\prod_{j=1}^{d}\left[\sum_{n=1}^{N}\beta_j^{(n)}\right], \tag{75}$$

that coincides with $\widehat{Z}_d$ in Eq. (74).

- **Adaptive resampling ($0 < \eta < 1$):** for the sake of simplicity, let us start considering a unique resampling step applied at the k-th iteation with $k < d$. We check if both estimators are equal at d-th iteration of the recursion. Due to Eq. (71), we have $\overline{Z}_k \equiv \widehat{Z}_k$,[8] since before the k-th iteration no resampling has been applied. With the proper weighting $w_k^{(n)} = \widehat{Z}_k$ for all n, at the next iteration we have

$$\widehat{Z}_{k+1} = \frac{1}{N}\sum_{n=1}^{N}w_k^{(n)}\beta_{k+1}^{(n)} = \widehat{Z}_k\left[\frac{1}{N}\sum_{n=1}^{N}\beta_{k+1}^{(n)}\right],$$

and using Eq. (72), we obtain

$$\overline{Z}_{k+1} = \overline{Z}_k\left[\sum_{n=1}^{N}\frac{1}{N}\beta_{k+1}^{(n)}\right] = \widehat{Z}_k\left[\frac{1}{N}\sum_{n=1}^{N}\beta_{k+1}^{(n)}\right],$$

so that the estimators are equivalent also at the $(k+1)$-th iteration, $\overline{Z}_{k+1} \equiv \widehat{Z}_{k+1}$. Since we are assuming no resampling steps after the k-th iteration and until the d-th iteration, we have that $\overline{Z}_i \equiv \widehat{Z}_i$ for $i = k+2,\ldots,d$ due to we are in a SIS scenario for $i > k$ (see Eq. (71)). This reasoning can be easily extended for different number of resampling steps.

---

[8]We consider to compute the estimators before the resampling.

C Consistency of GMS estimators

Dynamic of GMS. We have already seen that we can recover an I-MTM chain from the GMS outputs applying one resampling step for each t when $\mathcal{S}_t \neq \mathcal{S}_{t-1}$, i.e.,

$$\boldsymbol{\theta}_t = \begin{cases} \boldsymbol{\theta}_t \sim \sum_{n=1}^{N} \frac{\rho_{n,t}}{\sum_{i=1}^{N} \rho_{i,t}} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{n,t}), & \text{if} \quad \mathcal{S}_t \neq \mathcal{S}_{t-1}, \\ \boldsymbol{\theta}_{t-1}, & \text{if} \quad \mathcal{S}_t = \mathcal{S}_{t-1}, \end{cases} \tag{76}$$

for $t = 1, \ldots, T$. The sequence $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$ is a chain obtained by one run of an I-MTM2 technique. Note that (a) the sample generation, (b) the acceptance probability function and hence (c) the dynamics of GMS exactly coincide with the corresponding steps of I-MTM2 (or PMH; depending on candidate generation procedure). Hence, the ergodicity of the recovered chain is ensured.

Parallel chains from GMS outputs. As described in Section 4.1.3, we can extend the consideration above for generation C parallel I-MTM2 chains. Indeed, we resample C times instead of only one, i.e.,

$$\boldsymbol{\theta}_t^{(c)} = \begin{cases} \boldsymbol{\theta}_t \sim \sum_{n=1}^{N} \frac{\rho_{n,t}}{\sum_{i=1}^{N} \rho_{i,t}} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{n,t}), & \text{if} \quad \mathcal{S}_t \neq \mathcal{S}_{t-1}, \\ \boldsymbol{\theta}_{t-1}, & \text{if} \quad \mathcal{S}_t = \mathcal{S}_{t-1}, \end{cases}$$

for $c = 1, \ldots, C$, where the super-index denotes the c-th chain (similar procedures have been suggested in [20, 42]). Clearly, the resulting C parallel chains are not independent, and there is an evident loss of performance w.r.t. the case of independent chains. However, at each iteration, the number of target evaluations per iteration is only N instead of NC. Note that that each chain in ergodic, so that each estimator $\widetilde{I}_T^{(c)} = \frac{1}{T} \sum_{t=1}^{T} g(\boldsymbol{\theta}_t^{(c)})$ is consistent (i.e., convergence to the true value for $T \to \infty$). As a consequence, the arithmetic mean of consistent estimators,

$$\widetilde{I}_{C,T} = \frac{1}{C} \sum_{c=1}^{C} \widetilde{I}_T^{(c)} = \frac{1}{CT} \sum_{t=1}^{T} \sum_{c=1}^{C} g(\boldsymbol{\theta}_t^{(c)}), \tag{77}$$

is also consistent, for all values of $C \geq 1$.

GMS as limit case. Let us consider the case $\mathcal{S}_t \neq \mathcal{S}_{t-1}$ (the other one is trivial), at some iteration t. In this scenario, the samples of the C parallel I-MTM2 chains, $\boldsymbol{\theta}_t^{(1)}, \boldsymbol{\theta}_t^{(2)}, \ldots, \boldsymbol{\theta}_t^{(C)}$, are obtained by resampled independently C samples from the set $\{\boldsymbol{\theta}_{1,t}, \ldots, \boldsymbol{\theta}_{N,t}\}$ according to the normalized weights $\bar{\rho}_{n,t} = \frac{\rho_{n,t}}{\sum_{i=1}^{N} \rho_{i,t}}$, for $n = 1, \ldots, N$. Recall that the samples $\boldsymbol{\theta}_t^{(1)}, \boldsymbol{\theta}_t^{(2)}, \ldots, \boldsymbol{\theta}_t^{(C)}$, will be used in the final estimator $\widetilde{I}_{C,T}$ in Eq. (77).

Let us denote as $\#j$ the number of times that a specific candidate $\boldsymbol{\theta}_{j,t}$ (contained in the set $\{\boldsymbol{\theta}_{n,t}\}_{n=1}^{N}$) has been selected as state of one of C chains, at the t iteration. As $C \to \infty$, The fraction $\frac{\#j}{C}$ approaches exactly the corresponding weights $\bar{\rho}_{j,t}$. Then, for $C \to \infty$, we have that the estimator in Eq. (77) approaches the GMS estimator, i.e.,

$$\lim_{C \to \infty} \widetilde{I}_{C,T} = \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \bar{\rho}_{n,t} g(\boldsymbol{\theta}_{n,t}). \tag{78}$$

45

670   Since $\widetilde{I}_{C,T}$ as $T \to \infty$ is consistent for all values of C, then the GMS estimator is also consistent (and it can be obtained as $C \to \infty$).