DISTRIBUTED PARTICLE METROPOLIS-HASTINGS SCHEMES

Luca Martino[⋆]*, Víctor Elvira*[†]*, Gustau Camps-Valls*[⋆]

[⋆] Image Processing Laboratory, Universitat de València (Spain).
[†] IMT Lille Douai CRISTAL (UMR 9189), Villeneuve d'Ascq (France).

ABSTRACT

We introduce a Particle Metropolis-Hastings algorithm driven by several parallel particle filters. The communication with the central node requires the transmission of only a set of weighted samples, one per filter. Furthermore, the marginal version of the previous scheme, called Distributed Particle Marginal Metropolis-Hastings (DPMMH) method, is also presented. DPMMH can be used for making inference on both a dynamical and static variable of interest. The ergodicity is guaranteed, and numerical simulations show the advantages of the novel schemes.

Index Terms— Particle MCMC, Particle Filtering, Monte Carlo, Bayesian inference, state-space models

1. INTRODUCTION

Particle filtering and Markov Chain Monte Carlo (MCMC) are broadly used Monte Carlo techniques in order to make inference about a variable of interest [1, 2, 3, 4]. The Particle Metropolis-Hastings (PMH) algorithm combines the particle filtering approach with a well-known MCMC method, the Metropolis-Hastings (MH) technique [5, 6, 4]. It has been particularly designed for making inference and smoothing about a hidden state in state-space models [7, 8]. In PMH, two trajectories obtained by different runs of a particle filter are compared according to suitable MH-type acceptance probability. Its marginal version, the so-called Particle Marginal MH (PMMH) method, is widely applied in signal processing for making inference jointly about both dynamic and static parameters: typically, the hidden state and a static unknown parameter of a state-space model [8, 9].

In this work, we show how several parallel particle filters (PFs) can drive a PMH-type technique. The PFs can use a different proposal pdf or, more generally, can be different kind of algorithms, for instance, some of them can be bootstrap PFs and other auxiliary PFs [2, 10, 7, 11, 12]. In the novel Distributed PMH scheme, the use of M parallel processor speeds up the final resulting technique. The communication to the central node requires the transfer of only M weighted particles, one per filter. Furthermore, we also introduce the Distributed Particle Marginal MH method, i.e., the marginal version of the novel scheme. Numerical simulations show the benefits of the proposed schemes.

2. BACKGROUND

In many applications the goal is to infer a variable of interest, $\mathbf{x} = x_{1:D} = [x_1 \dots, x_D]^\top \in \mathcal{X} \subseteq \mathbb{R}^{D\times\xi}$ (where $x_d \in \mathbb{R}^\xi$), given a set of related observations or measurements, $\mathbf{y} \in \mathbb{R}^{d_y}$. The statistical information is summarized in the posterior probability density function (pdf) given by

$$\bar{\pi}(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})}, \quad (1)$$

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf and $Z(\mathbf{y})$ is the marginal likelihood (a.k.a., Bayesian evidence). Generally, Z is unknown and often impossible to be computed in closed form, hence we only assume to be able to evaluate the unnormalized target function, $\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})$.[1] Furthermore, the computation of integrals involving $\bar{\pi}(\mathbf{x})$ are often intractable. We consider the problem of approximating via Monte Carlo a complicated integral involving the target $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$ and an integrable function $\mathbf{h}(\mathbf{x}) : \mathbb{R}^{D\times\xi} \to \mathbb{R}^{d_h}$, i.e.,

$$\mathbf{I} = E_{\bar{\pi}}[\mathbf{h}(\mathbf{X})] = \frac{1}{Z}\int_{\mathcal{X}} \mathbf{h}(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \qquad \mathbf{X} \sim \bar{\pi}(\mathbf{x}). \quad (2)$$

In this work, we compute an estimator $\widehat{\mathbf{I}}$ of $\mathbf{I}$ using an MCMC technique driven by a particle approximation of measure of $\bar{\pi}(\mathbf{x})$ obtained via Sequential Importance Resampling (SIR) [3, 2, 4].

2.1. Particle Filtering

Let us assume that the target density can be factorized as

$$\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x}) \quad = \quad \gamma_1(x_1)\prod_{d=2}^{D}\gamma_d(x_d|x_{d-1}). \quad (3)$$

This work has been supported by the European Research Council (ERC) through the ERC Consolidator Grant SEDAL ERC-2014-CoG 647423.

[1] We intentionally drop the dependence on $\mathbf{y}$ of the marginal likelihood, i.e. $Z \equiv Z(\mathbf{y})$ for the ease of notation.

For instance, this factorization is possible in the state-space model framework [2, 7]. Given a proposal pdf factorized in the same way, i.e., $q(\mathbf{x}) = q_1(x_1)\prod_{d=2}^{D} q_d(x_d|x_{d-1})$, we can draw N samples from the proposal, $\mathbf{x}^{(n)} = x_{1:D}^{(n)} = [x_1^{(n)} \dots, x_D^{(n)}]^\top \sim q(\mathbf{x})$, where $x_d^{(n)} \sim q_d(x_d|x_{d-1})$, and we assign the importance weight $w^{(n)} = \frac{\pi(\mathbf{x}^{(n)})}{q(\mathbf{x}^{(n)})}$. The weight above can be computed recursively and, in this case, the resulting technique is called Sequential Importance Sampling (SIS). If resampling steps are incorporated during the recursion, the method is known as Sequential Importance Resampling (SIR) or alternatively particle filtering [2, 7]. Table 1 shows a SIR scheme where a resampling step is performed at each iteration (a.k.a., *bootstrap particle filter*) and a proper weighting of a resampled particle is applied [13, 14, 15]. With a SIR procedure, we obtain a particle approximation of the measure of the target pdf, i.e.,

$$\begin{aligned}\widehat{\pi}(\mathbf{x}|\mathbf{x}^{(1:N)}) &= \frac{1}{N\widehat{Z}}\sum_{n=1}^{N} w^{(n)}\delta(\mathbf{x}-\mathbf{x}^{(n)}),\\ &= \sum_{n=1}^{N} \bar{w}^{(n)}\delta(\mathbf{x}-\mathbf{x}^{(n)}), \qquad (4)\end{aligned}$$

where $\bar{w}^{(n)} = \frac{w^{(n)}}{\sum_{j=1}^{N} w^{(j)}}$ and $\widehat{Z} = \frac{1}{N}\sum_{j=1}^{N} w^{(j)}$ is an unbiased estimator of the marginal likelihood.

Table 1: Sequential Importance Resampling (SIR)

- **Initialization:** Choose $x_0^{(n)}$ and set $\widetilde{w}_0^{(n)} = \frac{1}{N}$ for $n = 1, \dots, N$.
- **For** $d = 1, \dots, D$:

  1. *Propagation:* Draw $x_d^{(n)} \sim q_d(x_d|x_{d-1}^{(n)})$, for $n = 1, \dots, N$.
  2. *Weighting:* Compute the weights
  $$w_d^{(n)} = \widetilde{w}_{d-1}^{(n)}\beta_d^{(n)}, \qquad (5)$$
  where $\beta_d^{(n)} = \frac{\gamma_d(x_d^{(n)}|x_{d-1}^{(n)})}{q_d(x_d^{(n)}|x_{d-1}^{(n)})}$, for $n = 1, \dots, N$.
  3. *Resampling:*
     (a) Resample N particles from the current approximation, $\widetilde{x}_d^{(n)} \sim \sum_{i=1}^{N} \bar{w}_d^{(i)}\delta(x - x_d^{(i)})$, where $\bar{w}_d^{(i)} = \frac{w_d^{(i)}}{\sum_{j=1}^{N} w_d^{(j)}}$ and $n = 1, \dots, N$.
     (b) Set $x_d^{(n)} = \widetilde{x}_d^{(n)}$ and $\widetilde{w}_d^{(n)} = \frac{1}{N}\sum_{n=1}^{N} w_d^{(n)}$, for all $n = 1, \dots, N$ (see [13]).

- **Return:** Set $\{\mathbf{x}^{(n)} = x_{1:D}^{(n)}, w^{(n)} = w_D^{(n)}\}_{n=1}^{N}$, so that
$$\widehat{\pi}(\mathbf{x}|\mathbf{x}^{(1:N)}) = \frac{1}{N\widehat{Z}}\sum_{n=1}^{N} w^{(n)}\delta(\mathbf{x}-\mathbf{x}^{(n)}).$$

3. DISTRIBUTED PARTICLE METROPOLIS-HASTINGS ALGORITHM

The Particle Metropolis-Hastings (PMH) algorithm combines the SIR method with the MH technique. It has been particularly designed for making inference about a dynamic variable when the posterior can be factorized as in Eq. (3) [8]. In PMH, two different trajectories obtained by different runs of a particle filter (as in Table 1) are compared according to suitable MH-type acceptance probability. In this work, we show how several parallel particle filters, each one consider a different proposal pdf, can drive a PMH-type technique. Let us consider the problem of making inference about the variable of interest $\mathbf{x} = x_{1:D} = [x_1, \dots, x_D]^\top$ according to the posterior $\bar{\pi}$ factorized as in Eq. (3). The standard PMH method uses a single proposal pdf $q(\mathbf{x}) = q_1(x_1)\prod_{d=2}^{D} q_d(x_d|x_{1:d-1})$, employed in a SIR method in order to generate new candidates before of applying the MH-type test [4]. Here, we assume that M independent processing units are available jointly with a central node. We use M parallel particle filters, each one with a different proposal pdf, $q_m(\mathbf{x}) = q_{m,1}(x_1)\prod_{d=2}^{D} q_{m,d}(x_d|x_{1:d-1})$, one per each processor. Then, after one run of the parallel particle filters[2], we obtain M particle approximations, $\widehat{\pi}(\mathbf{x}|\mathbf{v}_m^{(1:N)})$, of the target pdf, where we have denoted with $\{\mathbf{v}_m^{(n)}\}_{n=1}^{N}$ the particles generated by the m-th filter. Since we aim to reduce the communication cost to the central node, we consider that each machine only transmits the pair $\{\widetilde{\mathbf{x}}_m, \widehat{Z}_m\}$, where $\widetilde{\mathbf{x}}_m \sim \widehat{\pi}(\mathbf{x}|\mathbf{v}_m^{(1:N)})$. The *Distributed Particle Metropolis-Hastings* (DPMH) technique is summarized in Table 2.

In step 1 of Table 2, different kinds of particle filtering algorithms can be also employed (not only using different proposal pdfs). In step 2, M resampling steps are performed, one per processor. An additional resampling step is performed in the central node (step 3). The resampled particle $\widetilde{\mathbf{x}}$ is accepted as a new state of the chain with probability α in Eq. (6). Otherwise, the chain remains in the previous state (i.e., $\mathbf{x}_t = \mathbf{x}_{t-1}$) as in a classical MH algorithm. The standard PMH method is a special case of the DPMH of Table 2 for $M = 1$. Note also that the method in Table 2 has the structure of a Independent Multiple Try Metropolis (I-MTM) algorithm using different proposal pdfs [16, 17, 18, 19] considering the first two steps as a sophisticated proposal procedure for generating M different tries.

Ergodicity. The ergodicity of chain generated by DPMH is ensured since it can be interpreted as a standard PMH method

[2] We consider the same number of particles N for all the filters, for simplicity.

Table 2: Distributed Particle MH algorithm

- **Initialization:** Choose $\mathbf{x}_0$ and $\widehat{Z}_{m,0}$ for $m = 1, \ldots, M$.
- For $t = 1, \ldots, T$:

1. *(Parallel Processors)* Construct M particle approximations $\widehat{\pi}(\mathbf{x}|\mathbf{v}_m^{(1:N)}) = \frac{1}{N\widehat{Z}_m}\sum_{n=1}^{N} w_m^{(n)}\delta(\mathbf{x} - \mathbf{v}_m^{(n)})$, where $\widehat{Z}_m = \frac{1}{N}\sum_{n=1}^{N} w_m^{(n)}$, with $m = 1, \ldots, M$, using M parallel SIR methods given in Table 1, with a different proposal $q_m(\mathbf{x})$.
2. *(Parallel Processors)* Draw $\widetilde{\mathbf{x}}_m \sim \widehat{\pi}(\mathbf{x}|\mathbf{v}_m^{(1:N)})$, for $m = 1, \ldots, M$, and transmit $\{\widetilde{\mathbf{x}}_m, \widehat{Z}_m\}_{m=1}^{M}$.
3. *(Central Node)* Draw $\widetilde{\mathbf{x}} \sim \widehat{\pi}(\mathbf{x}|\widetilde{\mathbf{x}}_{1:M}) = \sum_{m=1}^{M} \frac{\widehat{Z}_m}{\sum_{j=1}^{M}\widehat{Z}_j}\delta(\mathbf{x} - \widetilde{\mathbf{x}}_m)$.
4. *(Central Node)* Set $\mathbf{x}_t = \widetilde{\mathbf{x}}$ and $\widehat{Z}_{m,t} = \widehat{Z}_m$, for $m = 1, \ldots, M$, with probability
$$\alpha = \min\left[1, \frac{\sum_{m=1}^{M}\widehat{Z}_m}{\sum_{m=1}^{M}\widehat{Z}_{m,t-1}}\right]. \qquad (6)$$
Otherwise, with prob. $1 - \alpha$, set $\mathbf{x}_t = \mathbf{x}_{t-1}$ and $\widehat{Z}_{m,t} = \widehat{Z}_{m,t-1}$, for all m.

- **Return:** The Markov chain $\{\mathbf{x}_t\}_{t=1}^{T}$.

considering a single particle approximation

$$\widehat{\pi}(\mathbf{x}|\mathbf{v}_{1:M}^{(1:N)}) = \sum_{m=1}^{M} \frac{\widehat{Z}_m}{\sum_{j=1}^{M}\widehat{Z}_j}\widehat{\pi}(\mathbf{x}|\mathbf{v}_m^{(1:N)}), \qquad (7)$$

This particle approximation can be interpreted as being obtained by a single particle filter splitting the particles in M disjoint sets and then applying the partial resampling approach [13, 14], i.e., performing resampling steps within these sets. Hence, we resample once, i.e., draw $\widetilde{\mathbf{x}} \sim \widehat{\pi}(\mathbf{x}|\mathbf{v}_{1:M}^{(1:N)})$. This procedure is equivalent to steps 1-2-3 of Table 2. The proper weight of this resampled particle is $\widehat{Z} = \frac{1}{M}\sum_{m=1}^{M}\widehat{Z}_m$, so that the acceptance function of the equivalent classical PMH method is $\alpha(\mathbf{x}_{t-1}, \widetilde{\mathbf{x}}) = \min\left[1, \frac{\widehat{Z}}{\widehat{Z}_{t-1}}\right] = \min\left[1, \frac{\frac{1}{M}\sum_{m=1}^{M}\widehat{Z}_m}{\frac{1}{M}\sum_{m=1}^{M}\widehat{Z}_{m,t-1}}\right]$, where $\widehat{Z}_{t-1} = \frac{1}{M}\sum_{m=1}^{M}\widehat{Z}_{m,t-1}$ [8, 13].

Benefits. An advantage of the DPMH scheme is that the generation of samples can be parallelized (i.e., fixing the computational cost, DPMH allows the use of M processors in parallel) and the communication to the central node requires the transfer of only M particles, $\widetilde{\mathbf{x}}'_m$, and M weights, $\widehat{Z}'_m$, instead of NM particles and NM weights. Another important benefit of DPMH is that different types of particle filters can be jointly employed, for instance, different proposal pdfs can be used. Its marginal version is described below.

3.1. Distributed Particle Marginal MH

In several applications, a static and a dynamical variable should be estimated. Let us consider $\mathbf{x} = x_{1:D} \in \mathcal{X} \subseteq \mathbb{R}^{D\times\xi}$ and an additional model parameter $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$. For instance, in the state-space models, $x_d \in \mathbb{R}^\xi$ represents the hidden state (hence, $\mathbf{x} = x_{1:D}$ is the hidden trajectory to be estimated) and $\boldsymbol{\theta}$ a static unknown parameter of the model [20, 21, 22, 23, 15]. Assuming a prior pdf $g_\theta(\boldsymbol{\theta})$ over $\boldsymbol{\theta}$, and a factorized complete posterior pdf, we have

$$\bar{\pi}_c(\mathbf{x}, \boldsymbol{\theta}) \propto \pi_c(\mathbf{x}, \boldsymbol{\theta}) = g_\theta(\boldsymbol{\theta})\pi(\mathbf{x}|\boldsymbol{\theta}),$$

where $\pi(\mathbf{x}|\boldsymbol{\theta}) = \gamma_1(x_1|\boldsymbol{\theta})\prod_{d=2}^{D}\gamma_d(x_d|x_{1:d-1}, \boldsymbol{\theta})$. The Distributed Marginal PMH (DPMMH) technique is then summarized in Table 3. We can easily design a marginal version of DPMH in Section 3, drawing $\boldsymbol{\theta}' \sim q_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$ and run M particle filters addressing the target pdf $\bar{\pi}(\mathbf{x}|\boldsymbol{\theta}')$.

Table 3: Distributed Particle Marginal MH (DPMMH)

- **Initialization:** Choose $\mathbf{x}_0$, $\boldsymbol{\theta}_0$, and $\widehat{Z}(\boldsymbol{\theta}_0)$.
- **For** $t = 1, \ldots, T$:

1. Draw $\boldsymbol{\theta}' \sim q_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$ and $\widetilde{\mathbf{x}}_m \sim \widehat{\pi}(\mathbf{x}|\mathbf{v}_m^{(1:N)}, \boldsymbol{\theta}') = \frac{1}{N\widehat{Z}_m(\boldsymbol{\theta}')}\sum_{n=1}^{N} w_m^{(n)}(\boldsymbol{\theta}')\delta(\mathbf{x} - \mathbf{v}_m^{(n)})$, for $m = 1, \ldots, M$, where each approximation $\widehat{\pi}$ is obtained with one run of a particle filter. Transmit to the central node $\{\boldsymbol{\theta}', \widetilde{\mathbf{x}}_m, \widehat{Z}_m\}_{m=1}^{M}$.
2. Draw $\widetilde{\mathbf{x}} \sim \sum_{m=1}^{M}\frac{\widehat{Z}_m}{\sum_{j=1}^{M}\widehat{Z}_j}\delta(\mathbf{x} - \widetilde{\mathbf{x}}_m)$.
3. Set $\boldsymbol{\theta}_t = \boldsymbol{\theta}'$, $\mathbf{x}_t = \widetilde{\mathbf{x}}$, with probability
$$\alpha = \min\left[1, \frac{\left[\sum_{m=1}^{M}\widehat{Z}_m(\boldsymbol{\theta}')\right] g_\theta(\boldsymbol{\theta}')q_\theta(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}')}{\left[\sum_{m=1}^{M}\widehat{Z}_m(\boldsymbol{\theta}_{t-1})\right] g_\theta(\boldsymbol{\theta}_{t-1})q_\theta(\boldsymbol{\theta}'|\boldsymbol{\theta}_{t-1})}\right].$$
Otherwise, set $\boldsymbol{\theta}_t = \boldsymbol{\theta}'$ and $\mathbf{x}_t = \mathbf{x}_{t-1}$.

- **Return:** The Markov chain $\{\boldsymbol{\theta}_t, \mathbf{x}_t\}_{t=1}^{T}$.

4. NUMERICAL SIMULATIONS

We consider the challenging problem of estimating the Leaf Area Index (LAI) from remote sensing (satellite) observations. Let us denote LAI as $x_d \in \mathbb{R}^+$ (where $d \in \mathbb{N}^+$ also represents a temporal index) [24]. Since $x_t > 0$, we consider Gamma prior pdfs over the evolutions of LAI and Gaussian perturbations for the "in-situ" received measurements, y_t. More specifically, we assume the following state-space model,

$$\begin{cases} g_d(x_d|x_{d-1}) = \mathcal{G}\left(x_d\middle|\frac{x_{d-1}}{b}, b\right), \\ \ell_d(y_d|x_d) = \mathcal{N}(y_d|x_d, \lambda^2) \end{cases} \qquad (8)$$

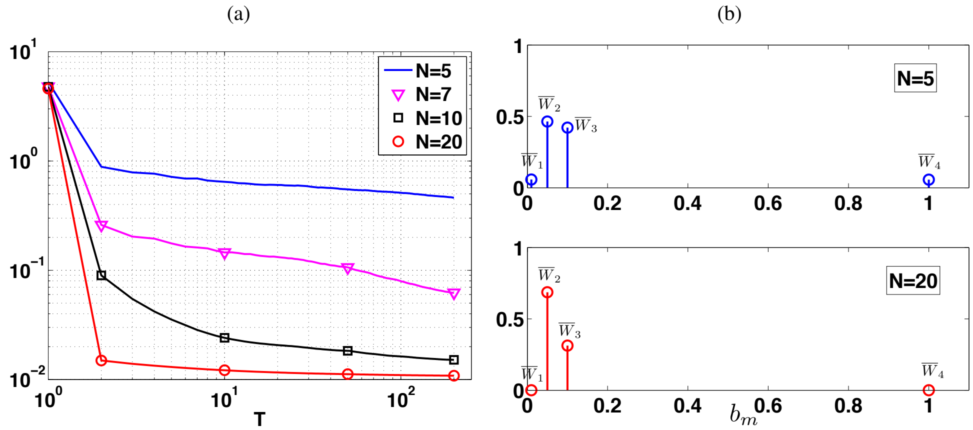

Fig. 1: **(a)** MSE in estimation of the trajectory (averaged over 2000 runs) obtained by DPMH as function T and different values of $N \in \{5, 7, 10, 20\}$. **(b)** Averaged values of the normalized weights $\overline{W}_m = \frac{\widehat{Z}_m}{\sum_{j=1}^{M} \widehat{Z}_j}$ (with $N = 5$ and $N = 10$) associated to each filter. DPMH is able to detect the best variances (b_2 and b_3) of the proposal pdfs among the values $b_1 = 0.01$, $b_2 = 0.05$, $b_3 = 0.1$ and $b_4 = 1$ (as confirmed by Table 4).

for $d = 2, \dots, D$, with initial probability $g_1(x_1) = \mathcal{G}(x_1|1, 1)$, where $b, \lambda > 0$. Note that the expected value of the Gamma pdf above is x_{d-1} and the variance is b. Considering that all the parameters of the model are known, the posterior pdf is

$$\bar{\pi}(\mathbf{x}|\mathbf{y}) \propto \left(\prod_{d=2}^{D} \ell_d(y_d|x_d) g_d(x_d|x_{d-1}) \right) g_1(x_1),$$

with $\mathbf{x} = x_{1:D} \in \mathbb{R}^D$. For generating the ground-truth (i.e., the trajectory $\mathbf{x}^* = x^*_{1:D} = [x^*_1, \dots, x^*_D]$), we simulate the temporal evolution of LAI in one year (i.e., $1 \leq d \leq D = 365$) by using a double logistic function (as suggested in the literature [24]), i.e.,

$$x_d = a_1 + a_2 \left(\frac{1}{1 + \exp(a_3(d - a_4))} + \frac{1}{1 + \exp(a_5(d - a_6))} + 1 \right), \tag{9}$$

with $a_1 = 0.1$, $a_2 = 5$, $a_3 = -0.29$, $a_4 = 120$, $a_5 = 0.1$ and $a_6 = 240$ as employed in [24]. The observations $\mathbf{y} = y_{2:D}$ are then generated (each run) according to $y_d \sim \ell_d(y_d|x_d) = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left(-\frac{1}{2\lambda^2}(y_d - x_d)^2\right)$. First of all, we test the standard PMH and DPMH (fixing $\lambda = 0.1$). For DPMH, we use $M = 4$ parallel filters with different scale parameters $\mathbf{b} = [b_1 = 0.01, b_2 = 0.05, b_3 = 0.1, b_4 = 1]^\top$. Figure 1(a) depicts the evolution of the MSE obtained by DPMH as a function of T and considering different values of $N \in \{5, 7, 10, 20\}$. The performance of DPMH improves as T and N grow, as expected. DPMH detects the best parameters among the four values in $\mathbf{b}$, following the weights $\overline{W}_m$ (see Figure 1(b)) and DPMH takes advantage of this ability. Indeed, we compare DPMH with $N = 10$, $T = 200$, $M = 4$ using the variances in the vector $\mathbf{b}$, with $M = 4$ different standard PMH algorithms with $N = 40$ and $T = 200$ (clearly, $M = 1$ for PMH) in order to keep the total number of evaluation of the posterior fixed, $E = NMT = 8 \cdot 10^3$, each one using a parameter b_m, $m = 1, \dots, M$. The results, averaged over 2000 runs, are shown in Table 4. In terms of MSE, DPMH always outperforms the 4 possible standard PMH methods. Moreover, due to the parallelization, in this case DPMH can save $\approx 15\%$ of the spent computational time.

Table 4: Comparison among PMH and DPMH with $E = NMT = 8 \cdot 10^3$ and $T = 200$ ($\lambda = 0.1$), estimating the trajectory $\mathbf{x}^* = x^*_{1:D}$.

Proposal Var	Standard PMH	DPMH
	$N = 40$ ($M = 1$)	$N = 10$ $M = 4$
	MSE	**MSE**
$b_1 = 0.01$	0.0422	0.0108
$b_2 = 0.05$	0.0130	
$b_3 = 0.1$	0.0133	
$b_4 = 1$	0.0178	
Average	0.0216	0.0108
Norm. Time	1	0.83

5. CONCLUSIONS

We have presented Distributed Particle Metropolis-Hastings schemes driven by M parallel particle filters and where the communication with the central node require the transmission only M weighted samples. Each particle filter uses a different proposal density. Numerical simulations show the benefits of the novel schemes and ability of the method in automatically detecting the better proposal pdf. Based on this fact and the ideas in [1, 15, 11, 12], as future line, we plan to design adaptive DPMH schemes. Furthermore, we plan to tackle more challenging applications in Earth science and communications.

6. REFERENCES

[1] M. Bolić, P. M. Djurić, and S. Hong, “Resampling algorithms and architectures for distributed particle filters,” *IEEE Transactions Signal Processing*, vol. 53, no. 7, pp. 2442–2450, 2005.

[2] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, “Particle filtering,” *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, September 2003.

[3] W. J. Fitzgerald, “Markov chain Monte Carlo methods with applications to signal processing,” *Signal Processing*, vol. 81, no. 1, pp. 3–18, January 2001.

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

[5] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić, “Particle methods for change detection, system identification and control,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 423–438, March 2004.

[6] T. Bertozzi, D. Le Ruyet, C. Panazio, and H. Vu-Thien, “Channel tracking using particle filtering in unresolvable multipath environments,” *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 15, pp. 2328–2338, November 2004.

[7] A. Doucet and A. M. Johansen, “A tutorial on particle filtering and smoothing: fifteen years later,” *technical report*, 2008.

[8] C. Andrieu, A. Doucet, and R. Holenstein, “Particle Markov chain Monte Carlo methods,” *J. R. Statist. Soc. B*, vol. 72, no. 3, pp. 269–342, 2010.

[9] L. Martino, F. Leisen, and J. Corander, “On multiple try schemes and the Particle Metropolis-Hastings algorithm,” *viXra:1409.0051*, 2014.

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

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

[12] J. Míguez and M. A. Vázquez, “A proof of uniform convergence over time for a distributed particle filter,” *Signal Processing*, vol. 122, pp. 152–163, 2016.

[13] L. Martino, V. Elvira, and F. Louzada, “Weighting a resampled particle in Sequential Monte Carlo,” *IEEE Statistical Signal Processing Workshop, (SSP)*, vol. 122, pp. 1–5, 2016.

[14] L. Martino, V. Elvira, and G. Camps-Valls, “Group importance sampling for Particle Filtering and MCMC,” *arXiv:1704.02771*, pp. 1–39, 2017.

[15] L. Martino, J. Read, V. Elvira, and F. Louzada, “Cooperative parallel particle filters for on-line model selection and applications to urban mobility,” *Digital Signal Processing*, vol. 60, pp. 172–185, 2017.

[16] L Martino, “A review of Multiple Try MCMC algorithms for signal processing,” *Digital Signal Processing*, vol. 75, pp. 134–152, 2018.

[17] L. Martino and J. Read, “On the flexibility of the design of multiple try Metropolis schemes,” *Computational Statistics*, vol. 28, no. 6, pp. 2797–2823, 2013.

[18] L. Martino, V. P. Del Olmo, and J. Read, “A multipoint Metropolis scheme with generic weight functions,” *Statistics & Probability Letters*, vol. 82, no. 7, pp. 1445–1453, 2012.

[19] L. Martino and F. Louzada, “Issues in the Multiple Try Metropolis mixing,” *Computational Statistics*, vol. 32, no. 1, pp. 239–252, 2017.

[20] P. M. Djurić and J. Míguez, “Sequential particle filtering in the presence of additive Gaussian noise with unknown parameters,” in *Proceedings of ICASSP*, May 2002.

[21] J. Míguez, M. F. Bugallo, and P. M. Djurić, “A new class of particle filters for random dynamical systems with unknown statistics,” *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 15, pp. 2278–2294, November 2004.

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

[23] P. M. Djurić, “Sequential estimation of signals under model uncertainty,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., chapter 18, pp. 381–400. Springer, 2001.

[24] J. L. Gomez-Dans, P. E. Lewis, and M. Disney, “Efficient emulation of radiative transfer codes using Gaussian Processes and application to land surface parameter inferences,” *Remote Sensing*, vol. 8, no. 2, 2016.