PARTICLE GROUP METROPOLIS METHODS FOR TRACKING THE LEAF AREA INDEX

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

[⋆] Dept. of Signal Processing, Universidad Rey Juan Carlos, Madrid (Spain).
[†] School of Mathematics, University of Edinburgh (United Kingdom).
[+] Image Processing Laboratory, Universitat de València,València (Spain).

ABSTRACT

Monte Carlo (MC) algorithms are widely used for Bayesian inference in statistics, signal processing, and machine learning. In this work, we introduce an Markov Chain Monte Carlo (MCMC) technique driven by a particle filter. The resulting scheme is a generalization of the so-called Particle Metropolis-Hastings (PMH) method, where a suitable Markov chain of sets of weighted samples is generated. We also introduce a marginal version for the goal of jointly inferring dynamic and static variables. The proposed algorithms outperform the corresponding standard PMH schemes, as shown by numerical experiments.

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

1. INTRODUCTION

Particle filtering and Markov Chain Monte Carlo (MCMC) methods are Monte Carlo techniques widely applied in statistical models, in order to make inference about a dynamic and static parameters [1, 2, 3, 4]. The particle Metropolis-Hastings (PMH) algorithm combines the particle filtering approach with the Metropolis-Hastings (MH) technique, a well-known MCMC method [5, 6, 4]. The PMH scheme 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, has found a vast application in signal processing for estimating jointly both dynamic and static parameters [8, 9].

In this work, we introduce a novel MCMC technique driven by a particle filter (PF), called particle group Metropolis sampling (PGMS). The proposed algorithm yields a Markov chain of *sets* of weighted particles. The acceptance probabilities and the dynamics of the chain coincide exactly with those of the standard PMH scheme. Indeed, a PMH chain can be recovered by resampling the PGMS outputs. Furthermore, we also introduce the marginal version of PGMS method for the double goal of smoothing trajectories traced by a dynamic variable and estimating static parameters of the considered dynamic model. Numerical simulations show the benefits of the proposed schemes.

2. BACKGROUND

In many applications the goal is to infer a D-dimensional variable, $\mathbf{x} = x_{1:D} = [x_1 \ldots, x_D]^\top \in \mathcal{X} \subseteq \mathbb{R}^{D\times\xi}$ (where $x_d \in \mathbb{R}^\xi$), given a set of related data, $\mathbf{y} \in \mathbb{R}^{d_y}$. In a Bayesian setting, the statistical information is summarized in the posterior probability density function (pdf), i.e.,

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

where $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf and $Z \equiv Z(\mathbf{y})$ is the marginal likelihood (a.k.a., Bayesian evidence). Generally, Z is unknown, hence we only assume that we are able to evaluate $\pi(\mathbf{x}) = \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})$. Furthermore, the integrals involving $\bar{\pi}(\mathbf{x})$ are often analytically intractable. For instance, one often needs to calculate expected values as

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

where $\mathbf{h}(\mathbf{x}) : \mathbb{R}^{D\times\xi} \rightarrow \mathbb{R}^{d_h}$ is an integrable function and $\bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$. 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 by a particle filtering scheme [3, 2, 4].

Particle Filtering. Let us assume that the target density can be factorized as

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

For instance, this factorization is possible in the state-space models [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

This work has been supported by the European Research Council (ERC) through the ERC Consolidator Grant SEDAL ERC-2014-CoG 647423. V.E. acknowledges support from the *Agence Nationale de la Recherche* of France under PISCES project (ANR- 17-CE40-0031-01).

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 a resampling step is incorporated during the recursion, the method is known as sequential importance resampling (SIR) [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 [10, 11, 12]. With a SIR procedure, we obtain a particle approximation of the measure of the target pdf, i.e.,

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

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. The latter estimator is valid only if the resampled particles are properly weighted with $\widetilde{w}_d^{(n)} = \frac{1}{N}\sum\limits_{n=1}^{N} w_d^{(n)}$ [10, 11, 12]. Otherwise, an alternative estimator is $\widehat{Z} = \prod_{d=1}^D \left[\frac{1}{N}\sum_{n=1}^N \beta_d^{(n)}\right]$. See Appendix C in [11] for further details.

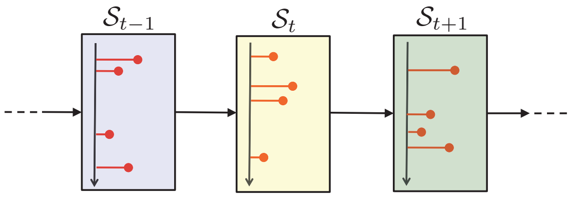


Fig. 1: Graphical representation of a Markov chain of set of weighted particles yielded by PGMS schemes.

3. PARTICLE GROUP METROPOLIS SAMPLING

In this section, we introduce a novel algorithm belonging to the class of group Metropolis sampling (GMS) methods. The GMS scheme, introduced in [11], is an MCMC technique which generalizes the independent multiple try Metropolis (I-MTM) algorithm [13]. It can be seen as a method which recycles auxiliary weighted samples in an I-MTM scheme. Moreover, the acceptance probability used in GMS is the extension of the acceptance probability of the acceptance probability employed in an independent Metropolis-Hastings (I-MH) method, considering the concept of *proper weighting* of a set of weighted samples [11, 13]. Here, we introduce a novel GMS technique specifically designed for scenarios when the target density can be factorized as in Eq. (3). The resulting algorithm, called *particle group Metropolis sampling* (PGMS) generalizes the Particle MH (PMH) sampler [8, 9]. Unlike PMH, the PGMS method produces a Markov chain of *sets* of weighted samples, as graphically represented in Figure 1. The PGMS scheme is summarized in Table 2. The PGMS method can be interpreted a way to recycle weighted trajectories (drawn in a PMH run) and include them in the final estimators. For this reason, PGMS outperforms PMH without any additional computational cost. The difference between PGMS and the PMH method is that PGMS does not use resampling steps at each iteration for selecting one sample among the N weighted samples. Indeed, PGMS stores the entire set, if accepted. Each set contains a group of weighted trajectories $\mathbf{x}^{(n)}$ obtained by a particle filter. If we are interested in estimating a unique integral $\mathbf{I}$, we can store only the estimator $\widehat{\mathbf{I}}_t = \sum_{n=1}^N \bar{w}^{(b)}\mathbf{h}(\mathbf{x}^{(n)})$ instead of all the particles.

Note that the acceptance probabilities α and the dynamics of PGMS coincides exactly with the PMH steps, so that the ergodicity of the chain is ensured [8, 9]. Indeed, on could recover a PMH chain from the PGMS outputs applying T re-

Table 1: Bootstrap particle filtering

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
$$w_d^{(n)} = \widetilde{w}_{d-1}^{(n)}\beta_d^{(n)}, \tag{5}$$
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 $\widetilde{w}_d^{(n)} = \frac{1}{N}\sum\limits_{n=1}^{N} w_d^{(n)}$, for all $n = 1, \dots, N$ (see [10]).

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

sampling steps i.e.,

$$\widetilde{\mathbf{x}}_t = \begin{cases} \mathbf{v}_t \sim \sum_{n=1}^{N} \bar{\rho}_{n,t}\delta(\mathbf{x}-\mathbf{x}_{n,t}), & \text{if } \mathcal{S}_t \neq \mathcal{S}_{t-1}, \\ \widetilde{\mathbf{x}}_{t-1} & \text{if } \mathcal{S}_t = \mathcal{S}_{t-1}, \end{cases} \tag{6}$$

for $t = 1, \dots, T$. Namely, $\{\widetilde{\mathbf{x}}_t\}_{t=1}^{T}$ is the chain obtained by one run of the PMH technique. Figure 2 graphically summarizes this procedure. Moreover, the main difference between PGMS scheme and the standard GMS method is that the samples are generated by a particle filter, i.e., in a sequential way (following the target factorization). Due to the use of resampling steps during the particle filtering stage, the resulting samples are correlated, not independent as in the standard GMS technique.

Table 2: Particle group Metropolis sampling (PGMS)

Initialization: Start with an initial set $\mathcal{S}_0 = \{\mathbf{x}_{n,0}, \rho_{n,0}\}_{n=1}^{N}$ and $\widehat{Z}_0 = \frac{1}{N}\sum_{n=1}^{N}\rho_{n,0}$.
For $t = 1, \dots, T$**:**

1. Construct a particle approximation using a particle filter (as described in Table 1),

$$\widehat{\pi}(\mathbf{x}|\mathbf{x}^{(1:N)}) = \sum_{n=1}^{N} \bar{w}^{(n)}\delta(\mathbf{x}-\mathbf{x}^{(n)}),$$

   and obtain $\widehat{Z}'$, as described in Section 2.

2. Define the set $\mathcal{S}' = \{\mathbf{x}^{(n)}, \bar{w}^{(n)}\}_{n=1}^{N}$.

3. Given $\mathcal{S}_t = \{\mathbf{x}_{n,t}, \bar{\rho}_{n,t}\}$, set $\mathcal{S}_t = \mathcal{S}'$ and $\widehat{Z}_t = \widehat{Z}'$, with probability

$$\alpha(\mathcal{S}_{t-1}, \mathcal{S}') = \min\left[1, \frac{\widehat{Z}'}{\widehat{Z}_{t-1}}\right]. \tag{7}$$

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

Return: $\{\mathcal{S}_t\}_{t=1}^{T}$ and $\{\widehat{Z}_t\}_{t=1}^{T}$.

3.1. Marginal version of PGMS

In many applications, static and dynamical parameters must be jointly estimated. More specifically, let again consider $\mathbf{x} = x_{1:D} \in \mathcal{X} \subseteq \mathbb{R}^{D\times\xi}$ and an additional static 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 [14, 15, 16, 17, 12]. In this scenario, assuming a prior pdf $g_\theta(\boldsymbol{\theta})$ over $\boldsymbol{\theta}$, the complete posterior pdf is

$$\bar{\pi}_c(\mathbf{x}, \boldsymbol{\theta}) \propto \pi_c(\mathbf{x}, \boldsymbol{\theta}) = g_\theta(\boldsymbol{\theta})\pi(\mathbf{x}|\boldsymbol{\theta}), \tag{8}$$

where

$$\pi(\mathbf{x}|\boldsymbol{\theta}) = \gamma_1(x_1|\boldsymbol{\theta})\prod_{d=2}^{D}\gamma_d(x_d|x_{1:d-1}, \boldsymbol{\theta}). \tag{9}$$

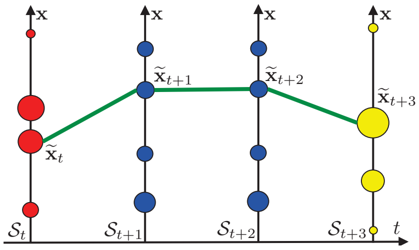

x
x
x
x
$\widetilde{\mathbf{x}}_{t+1}$
$\widetilde{\mathbf{x}}_{t+2}$
$\widetilde{\mathbf{x}}_{t+3}$
$\widetilde{\mathbf{x}}_t$
$\mathcal{S}_t$
$\mathcal{S}_{t+1}$
$\mathcal{S}_{t+2}$
$\mathcal{S}_{t+3}$
t

Fig. 2: Graphical representation of the recovery of a PMH chain by PGMS outputs. Each circle represents a possible trajectory $\widetilde{\mathbf{x}}_t = [\widetilde{x}_{1,t}, \dots, \widetilde{x}_{D,t}]$ and the size of the circle represents its weight. The green lines represent the transition of the recovered PMH chain, obtaining by resampling at each iteration.

In order to approximate $\bar{\pi}_c(\mathbf{x}, \boldsymbol{\theta})$, we can apply the particle marginal group Metropolis sampling (PM-GMS) algorithm, that is summarized in Table 3. PM-GMS draws a candidate $\boldsymbol{\theta}' \sim q_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1})$ and then run a particle filter addressing the target pdf $\bar{\pi}(\mathbf{x}|\boldsymbol{\theta}')$. PM-GMS is a generalization of the particle marginal MH (PMMH) algorithm, where a chain of set of (weighted) trajectories and a chain regarding the parameter $\boldsymbol{\theta}$ are jointly produced.

Table 3: The PM-GMS algorithm

Initialization: Start with $\boldsymbol{\theta}_0$, $\mathcal{S}_0$ and $\widehat{Z}(\boldsymbol{\theta}_0)$.
For $t = 1, \dots, T$ **:**

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

2. Run a particle filter to obtain the approximation $\widehat{\pi}(\mathbf{x}|\mathbf{v}^{(1:N)}, \boldsymbol{\theta}') = \sum_{n=1}^{N}\bar{w}^{(n)}\delta(\mathbf{x}-\mathbf{v}^{(n)})$ and the estimator $\widehat{Z}(\boldsymbol{\theta}')$.

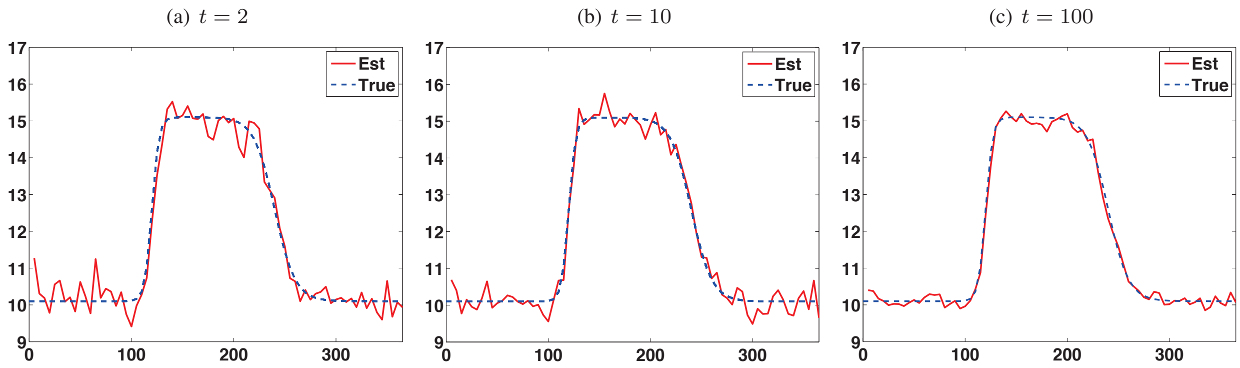
3. Define the set $\mathcal{S}' = \{\mathbf{v}^{(n)}, \bar{w}^{(n)}\}_{n=1}^{N}$.

4. Given $\mathcal{S}_t = \{\mathbf{x}_{n,t}, \bar{\rho}_{n,t}\}$, set $\mathcal{S}_t = \mathcal{S}'$, $\boldsymbol{\theta}_t = \boldsymbol{\theta}'$ and $\widehat{Z}_t = \widehat{Z}(\boldsymbol{\theta}')$, with probability

$$\alpha = \min\left[1, \frac{\widehat{Z}(\boldsymbol{\theta}')g_\theta(\boldsymbol{\theta}')q_\theta(\boldsymbol{\theta}_{t-1}|\boldsymbol{\theta}')}{\widehat{Z}(\boldsymbol{\theta}_{t-1})g_\theta(\boldsymbol{\theta}_{t-1})q_\theta(\boldsymbol{\theta}'|\boldsymbol{\theta}_{t-1})}\right].$$

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

Return: The Markov chain $\{\mathcal{S}_t, \boldsymbol{\theta}_t, \widehat{Z}_t\}_{t=1}^{T}$.


(a) $t = 2$

(b) $t = 10$

(c) $t = 100$

Fig. 3: Smoothing of the trajectory with PGMS (with $N = 5$, $\lambda = 0.1$) at different iterations **(a)** $t = 2$, **(b)** $t = 10$, and **(c)** $t = 100$, in one specific run. The true values, $\mathbf{x}^* = x^*_{1:D}$, are shown dashed lines whereas the estimated trajectory by PGMS, $\widehat{\mathbf{x}}_t = \widehat{x}_{1:D,t}$, with solid lines.

4. NUMERICAL SIMULATIONS

We consider the challenging problem of estimating biophysical parameters from remote sensing (satellite) observations. In particular, we focus on the estimation of the leaf area index (LAI) [18]. Let us denote LAI as $x_d \in \mathbb{R}^+$ (where $d \in \mathbb{N}^+$ also represents a temporal index) in a specific region at a latitude of 42° N [19]. 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

$$\begin{cases} g_d(x_d|x_{d-1}) & = \mathcal{G}\left(x_d \middle| \frac{x_{d-1}}{b}, b\right), \\ \ell_d(y_d|x_d) & = \mathcal{N}(y_d|x_d, \lambda^2), \end{cases} \qquad (10)$$

for $d = 2, \ldots, D$, with initial probability $g_1(x_1) = \mathcal{G}(x_1|1, 1)$, where $b, \lambda > 0$ and $c_d > 0$ is a normalizing constant. Note that the expected value of the Gamma pdf above is x_{d-1} and the variance is b.

First Experiment. First of all we consider that all the parameters of the model are known. The posterior pdf is

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

with $\mathbf{x} = x_{1:D} \in \mathbb{R}^D$. For generating the ground-truth (i.e., the trajectory $\mathbf{x}^* = x^*_{1:D} = [x^*_1, \ldots, x^*_D]$), 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 employed in [19]. In Figure 3, the true trajectory $x_{1:D}$ is depicted with dashed lines and the estimation (at different iterations) provided by PGMS in one specific run, in solid lines. The observations $\mathbf{y} = y_{2:D}$ are then generated each run according to the model $y_d \sim \ell_d(y_d|x_d)$. We compare the standard PMH and PGMS, setting $\lambda = 0.1$, $N = 40$ and $T = 200$ in terms of estimation of the true trajectory. We also consider different scale values $b \in \{0.01, 0.05, 0.1, 1\}$. The results, averaged over 2000 runs, are shown in Table 4. Note that PGMS outperforms the standard PMH in all cases, providing always a smaller mean square error (MSE).

Second Experiment. Now we consider that the parameter λ is also unknown, so that the complete variable of interest $[\mathbf{x}, \lambda] \in \mathbb{R}^{D+1}$. Then the posterior is $\bar{\pi}(\mathbf{x}, \lambda|\mathbf{y}) \propto \ell(\mathbf{y}|\mathbf{x}, \lambda)g(\mathbf{x}, \lambda)$ according to the model Eq. (10), where $g(\mathbf{x}, \lambda) = g(\mathbf{x})g_\lambda(\lambda)$ and $g_\lambda(\lambda)$ is a uniform pdf in $[0.01, 5]$. Then we test the standard PMMH and PM-GMS with $q_\lambda(\lambda) = g_\lambda(\lambda)$ (see 3.1), for estimating $[\mathbf{x}^*, \lambda^*]$ where $\mathbf{x}^* = x^*_{1:D}$ and $\lambda^* = 0.7$ are the true values. Table 5 compares the standard PMMH and PM-GMS for estimating λ^* (we set $N = 40$ and $T = 100$). We can observe that PM-GMS always outperforms the standard PMMH in terms of smaller MSE.

Table 4: MSE in estimating the trajectory $\mathbf{x}^* = x^*_{1:D}$, by PGMS and standard PMH with $N = 40$, $T = 200$, and $\lambda = 0.1$.

Method	Standard PMH	PGMS
	MSE	MSE
$b = 0.01$	0.0422	0.0380
$b = 0.05$	0.0130	0.0100
$b = 0.1$	0.0133	0.0102
$b = 1$	0.0178	0.0140

Table 5: Comparison among PM-GMS and the standard PMMH with $N = 40$ and $T = 100$, for estimating $\lambda^* = 0.7$.

Method	Standard PMMH	PM-GMS
	MSE	MSE
$b = 0.01$	0.0929	0.0901
$b = 0.05$	0.0186	0.0097
$b = 0.1$	0.0401	0.0288
$b = 1$	0.0223	0.0156

5. CONCLUSIONS

In this work, we present the particle group Metropolis sampling (PGMS) scheme which is an extension of the related PMH algorithm. PGMS outperforms the corresponding benchmark Monte Carlo technique without any extra computational cost, as we have shown in the numerical experiments.The PGMS method can be interpreted a suitable way of recycling particles in a PMH scheme and including them in the final estimators.

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] 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.

[11] L. Martino, V. Elvira, and G. Camps-Valls, "Group importance sampling for particle filtering and MCMC," *Digital Signal Processing*, vol. 82, pp. 133 – 151, 2018.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] J. M. Chen and T. A Black, "Defining leaf area index for non-flat leaves," *Plant, Cell and Environment*, vol. 15, pp. 421–429, 1992.

[19] 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.