PARTICLE FILTERS WITH INDEPENDENT RESAMPLING

Roland Lamberti[1], *Yohan Petetin*[1], *François Septier*[2], *François Desbouvries*[1]

(1) Samovar, Telecom Sudparis, CNRS, Université Paris-Saclay, 9 rue Charles Fourier, 91011 Evry, France
(2) Institut Mines-Télécom / Télécom Lille / CRIStAL UMR CNRS 9189, Villeneuve d'Ascq, France

ABSTRACT

In many signal processing applications we aim to track a state of interest given available observations. Among existing techniques, sequential Monte Carlo filters are importance sampling-based algorithms meant to propagate in time a set of weighted particles which represent the a posteriori density of interest. As is well known weights tend to degenerate over time, and resampling is a commonly used rescue for discarding particles with low weight. Unfortunately conditionally independent resampling produces a set of dependent samples and the technique suffers from sample impoverishment. In this paper we modify the resampling step of particle filtering techniques in order to produce independent samples per iteration. We validate our technique via simulations.

Index Terms— Sequential Monte Carlo, Particle Filters, resampling procedures

1. INTRODUCTION

Let $\mathbf{X}_k \in \mathbb{R}^p$ and $\mathbf{Y}_k \in \mathbb{R}^q$ be respectively a hidden and an observed process. Let $p(\mathbf{x}_k|\mathbf{y}_{0:k})$ (or in short $p_{k|k}$) denote the a posteriori filtering pdf of $\mathbf{x}_k$ given $\mathbf{y}_{0:k} = \{\mathbf{y}_i\}_{i=0}^k$. We assume that $\{\mathbf{X}_k, \mathbf{Y}_k\}_{k\in\mathbb{N}}$ is a Hidden Markov Chain (HMC):

$$p(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}) = p(\mathbf{x}_0) \prod_{i=1}^{k} f_{i|i-1}(\mathbf{x}_i|\mathbf{x}_{i-1}) \prod_{i=0}^{k} g_k(\mathbf{y}_i|\mathbf{x}_i).$$

We address the Bayesian filtering problem, which consists of computing $p(\mathbf{x}_k|\mathbf{y}_{0:k})$. Recursive solutions are of particular interest, and indeed $p_{k|k}$ can be computed from $p_{k-1|k-1}$ by the well known recursion (see e.g. [1] [2]):

$$p(\mathbf{x}_k|\mathbf{y}_{0:k}) = \frac{g_k(\mathbf{y}_k|\mathbf{x}_k) \int f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{0:k-1}) d\mathbf{x}_{k-1}}{p(\mathbf{y}_k|\mathbf{y}_{0:k-1})} \quad (1)$$

where $p(\mathbf{y}_k|\mathbf{y}_{0:k-1}) = \int \mathcal{N} d\mathbf{x}_k$ (here $\mathcal{N}$ stands for the numerator of Eq. (1)).

Many efforts have been devoted to the computation of Eq. (1). Generally one needs to resort to approximations. Among them sequential Monte Carlo (SMC) methods (see e.g. [3] [2]) which are based on importance sampling (IS) propagate a set of weighted particles $\{\mathbf{x}_{0:k-1}^i, w_{k-1}^i\}_{i=1}^N$, leading to a discrete approximation $\hat{p}(\mathbf{x}_k|\mathbf{y}_{0:k})$ of $p(\mathbf{x}_k|\mathbf{y}_{0:k})$. Resampling is widely used as a means to prevent weight degeneracy; however the particles produced in one time loop are not independent, and resampling also leads to sample impoverishment, since a single particle can be resampled several times, while particles with low weights are eliminated. In this paper we propose a modification of the resampling step of SMC algorithms, which consists of producing (conditionally) independent particles. We discuss the benefits and drawbacks of our algorithm and validate our results via simulations. The rest of this paper is organized as follows. Classical SMC algorithms, which involve dependent resampling, are recalled in section 2. Our method is described in section 3. Section 4 is devoted to simulations, and we end our paper with a conclusion.

2. PARTICLE FILTER WITH DEPENDENT RESAMPLING

Let us first recall the rationale of particle filtering (PF), which is based on the sequential application of IS techniques. At time $k-1$, we assume that we have a discrete approximation of $p(\mathbf{x}_{0:k-1}|\mathbf{y}_{0:k-1})$ given by a set of weighted samples $\{\mathbf{x}_{0:k-1}^i,\ w_{k-1}^i\}_{i=1}^N$, in which $\mathbf{x}_{0:k-1}^i \sim q(\mathbf{x}_{0:k-1}|\mathbf{y}_{0:k-1})$, $w_{k-1}^i \propto p(\mathbf{x}_{0:k-1}^i, \mathbf{y}_{0:k-1})/q(\mathbf{x}_{0:k-1}^i|\mathbf{y}_{0:k-1})$, $\sum_{i=1}^N w_{k-1}^i = 1$. At time k, the i^{th} trajectory is first extended by a particle sampled from an importance distribution $q(\mathbf{x}_k|\mathbf{x}_{0:k-1}^i, \mathbf{y}_{0:k})$ (from now on we will take $q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{0:k}) = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{0:k})$) and next weighted by a weight w_k^i proportional to $w_{k-1}^i f_{k|k-1}(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)\ g_k(\mathbf{y}_k|\mathbf{x}_k^i)/\ q(\mathbf{x}_k^i|\ \mathbf{x}_{k-1}^i, \mathbf{y}_{0:k})$.

However, it is well known (see e.g. [4]) that the algorithm degenerates when it is applied sequentially. More precisely, after some iterations only few particles have a significant non null weight. To address this weight degeneracy problem, a common rescue is to use after the weighting step a multinomial resampling step, which aims at discarding the particles with low weights and duplicating those with high weights (variations of the multinomial resampling techniques have also been proposed, see e.g. [5–7]). The resampling step is optional and is generally performed when the so-called effective sample size (ESS) criterion, $\text{ESS} = 1/\sum_{i=1}^N (w_k^i)^2$ falls under a given threshold [8] [9]. Note that the resampling

step is locally harmful (a single particle can be chosen several times, and some particles are eliminated, which results in support shrinkage); however it recreates diversity for the subsequent iterations.

In summary, we obtain the sampling importance resampling (SIR) algorithm, which is recalled below (the resampling step is optional, hence notation *(R)*). Starting from $\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^N$:

S. for all i, $1 \leq i \leq N$, sample $\tilde{\mathbf{x}}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{y}_{0:k})$;

W. for all i, $1 \leq i \leq N$, set $w_k^i \propto w_{k-1}^i \frac{f_{k|k-1}(\tilde{\mathbf{x}}_k^i|\mathbf{x}_{k-1}^i) g_k(\mathbf{y}_k|\tilde{\mathbf{x}}_k^i)}{q(\tilde{\mathbf{x}}_k^i|\mathbf{x}_{k-1}^i, \mathbf{y}_{0:k})}$, $\sum_{i=1}^N w_k^i = 1$;

(R.) for all i, $1 \leq i \leq N$, sample $\mathbf{x}_k^i \sim \sum_{j=1}^N w_k^j \delta_{\tilde{\mathbf{x}}_k^j}$, set $w_k^i = 1/N$.

Let us now focus on the combination of the three steps which composes one loop $k-1 \to k$ of the SMC algorithm when resampling is indeed performed. In that case, one can show easily that given $\{\mathbf{x}_{k-1}^i\}_{i=1}^N$, each $\mathbf{x}_k^i$ is drawn from a common pdf $\tilde{q}(\mathbf{x}_k|\{\mathbf{x}_{k-1}^i\}_{i=1}^N)$ (or $\tilde{q}(\mathbf{x}_k)$ in shorthand notation) where

$$\tilde{q}(\mathbf{x}) = \sum_{i=1}^N \int \frac{p_i(\mathbf{x})}{\frac{p_i(\mathbf{x})}{q_i(\mathbf{x})} + \sum_{j \neq i} \frac{p_j(\mathbf{x}^j)}{q_j(\mathbf{x}^j)}_{j \neq i}} \prod q_j(\mathbf{x}^j) \mathrm{d}\mathbf{x}^1 \cdots \mathbf{x}^j, \quad (2)$$

$$q_i(\mathbf{x}) = q(\mathbf{x}|\mathbf{x}_{k-1}^i, \mathbf{y}_{0:k}), \quad (3)$$

$$p_i(\mathbf{x}) = f_{k|k-1}(\mathbf{x}|\mathbf{x}_{k-1}^i) g_k(\mathbf{y}_k|\mathbf{x}) \quad (4)$$

In other words, given $\{\mathbf{x}_{k-1}^i\}_{i=1}^N$, all the particles $\mathbf{x}_k^i$, $1 \leq i \leq N$ obtained using the PF algorithm are identically distributed; however, it is obvious that the obtained particles are conditionally dependent (by construction they are sampled independently given $\{\mathbf{x}_{k-1}^i\}_{i=1}^N$ *and* $\{\tilde{\mathbf{x}}_{k-1}^i\}_{i=1}^N$, but are not independent given $\{\mathbf{x}_{k-1}^i\}_{i=1}^N$ only). For instance, when $\mathbf{x}_k^1$ has been sampled, there is a non-null probability that $\mathbf{x}_k^2 = \mathbf{x}_k^1$.

3. PF WITH INDEPENDENT RESAMPLING

3.1. Independent resampling

From the discussion of section 2, our objective is to replace the set $\{\mathbf{x}_k^i\}_{i=1}^N$ (obtained via a classical resampling step) by a set of particles $\{\overline{\mathbf{x}}_k^i\}_{i=1}^N$ which, given $\{\mathbf{x}_{k-1}^i\}_{i=1}^N$, are independent and identically distributed according to $\tilde{q}$ in (2).

The reason why is as follows. In order to compare sets $\{\mathbf{x}_k^i\}_{i=1}^N$ and $\{\overline{\mathbf{x}}_k^i\}_{i=1}^N$, assume that one wants to compute some moment

$$\Theta = \int f(\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{0:k}) \mathrm{d}\mathbf{x}_k, \quad (5)$$

and let $\widehat{\Theta}_k$ and $\widetilde{\Theta}_k$ be the Monte Carlo estimates of Θ built respectively from the set $\{\mathbf{x}_k^i\}_{i=1}^N$ produced by the particle filter with multinomial resampling described in Section 2, and that which would be deduced from samples $\{\overline{\mathbf{x}}_k^i\}_{i=1}^N$ drawn (conditionally) independently from $\tilde{q}(\mathbf{x})$:

$$\widehat{\Theta}_k = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_k^i), \quad (6)$$

$$\widetilde{\Theta}_k = \frac{1}{N} \sum_{i=1}^N f(\overline{\mathbf{x}}_k^i) \quad (7)$$

Then given $\{\mathbf{x}_{0:k-1}^i, w_{k-1}^i\}_{i=1}^N$, one can show easily that

$$\mathrm{E}(\widehat{\Theta}_k) = \mathrm{E}(\widetilde{\Theta}_k), \quad (8)$$

$$\mathrm{var}(\widehat{\Theta}_k) = \mathrm{var}(\widetilde{\Theta}_k) + \frac{N-1}{N} \mathrm{var}\left(\sum_{i=1}^N w_k^i f(\tilde{\mathbf{x}}_k^i) \right) \quad (9)$$

In other words, the estimator $\widetilde{\Theta}_k$ deduced from independent samples outperforms $\widehat{\Theta}_k$ whatever $f(.)$ in (5) (of course in practice the estimate of (5) is obtained before resampling, but (8) enables us to appreciate the interest of $\{\overline{\mathbf{x}}_k^i\}_{i=1}^N$ over $\{\mathbf{x}_k^i\}_{i=1}^N$).

The idea of obtaining independent particles can be simply illustrated in the particular case of the SIR algorithm with optimal importance distribution. In this case, the importance distribution $q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{y}_{0:k})$ coincides with $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{y}_k) \propto f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}^i) g_k(\mathbf{y}_k|\mathbf{x}_k)$, and the weights w_k^i are proportional to $w_{k-1}^i p(\mathbf{y}_k|\mathbf{x}_{k-1}^i)$ where $p(\mathbf{y}_k|\mathbf{x}_{k-1}^i) = \int f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}^i) g_k(\mathbf{y}_k|\mathbf{x}_k) \mathrm{d}\mathbf{x}_k$, and thus no longer depend on $\{\mathbf{x}_k^i\}_{i=1}^N$. For this setting, $\tilde{q}(\mathbf{x})$ in (2) reduces to the mixture pdf

$$\tilde{q}(\mathbf{x}) = \sum_{i=1}^N w_k^i p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{y}_k). \quad (10)$$

Consequently, one can draw conditionally independent samples $\{\overline{\mathbf{x}}_k^i\}_{i=1}^N$ according to $\tilde{q}(\mathbf{x})$ by a simple (and fast) two-step procedure: for each i, $1 \leq i \leq N$, select an index $j^i \sim \mathrm{Pr}(j^i = p) = w_k^p$, and then sample $\mathbf{x}_k^i \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^{j^i}, \mathbf{y}_k)$. This algorithm coincides with the fully adapted PF (FA-APF) algorithm which is known to outperform the SIR algorithm with optimal importance distribution [10] [11].

Unfortunately, in the general case we do not have a closed-form expression of $\tilde{q}(\mathbf{x})$ in (2). However, it remains possible to obtain samples $\overline{\mathbf{x}}_k^i$ drawn independently according to $\tilde{q}(\mathbf{x})$. Starting from $\{\mathbf{x}_{0:k-1}^i, w_{k-1}^i\}_{i=1}^M$ (the reason why we use notation $M \neq N$ will be explained soon) let us consider the following procedure:

S. for all i, j, $1 \leq i, j \leq M$, sample $\tilde{\mathbf{x}}_k^{i,j} \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{y}_{0:k})$

W. for all i, j, $1 \leq i, j \leq M$, set $w_k^{i,j} \propto w_{k-1}^i \times \frac{f_{k|k-1}(\tilde{\mathbf{x}}_k^{i,j}|\mathbf{x}_{k-1}^i) g_k(\mathbf{y}_k|\tilde{\mathbf{x}}_k^{i,j})}{q(\tilde{\mathbf{x}}_k^{i,j}|\mathbf{x}_{k-1}^i, \mathbf{y}_{0:k})}$, $\sum_{i=1}^N w_k^{i,j} = 1$;

(R.) for all j, $1 \leq j \leq M$, sample $\overline{\mathbf{x}}_k^j \sim \sum_{l=1}^N w_k^{l,j} \delta_{\tilde{\mathbf{x}}_k^{l,j}}$; set $\mathbf{x}_k^j = \overline{\mathbf{x}}_k^j$ and $w_k^j = 1/N$.

3.2. Comments and remarks

Let us comment this algorithm. By construction all particles $\{\mathbf{x}_k^i\}_{i=1}^M$ are different, so the algorithm no longer suffers from support shrinkage.

On the other hand, even if we start with an approximation of size M, the algorithm relies on the sampling and weighting of M^2 new particles, so involves an extra computational cost if we compare it with the particle filter of section 2 when $M = N$. However, note that the resampling step is not necessary performed at each iteration; consequently the above algorithm can be performed only when resampling is necessary. In addition, and in the spirit of Island particle filter [12] the independent resampling mechanism can easily be parallelized (note that we do *not* run several particle filters in parallel; see comments below).

Finally, our resampling technique raises one question which will be addressed in our simulations. Is it possible to select $M \leq N$ such that the computational cost of the two resampling procedures (dependent vs. independent) is of the same order? Ideally, we would like to improve the performances of classical PF when $M = \sqrt{N}$ and indeed we will see that it is actually possible in some cases.

3.3. Connection with existing works

Let us now discuss how our proposed independent resampling scheme differs from recent filtering methodologies based on "(Sequential or not) MC within SMC". The Island particle filter [12] provides a means of parallelizing SMC methods. Instead of considering a unique set of N particles, this method consists of dividing the particles population into N_1 sets (denoted islands) of N_2 particles each (with $N = N_1 N_2$). Each island of particles evolves according to the standard SMC framework, thus allowing for a complete parallelization of the islands. However, as discussed in [12], the bias which occurs from the division of the population (i.e. small islands of size N_2) can be reduced by performing an additional resampling step at the island level. This method is clearly different from our proposition since it can still suffer from the sample impoverishment of the intra (and inter) island resampling stages.

Finally, in [13] it is proposed to use an *empirical* distribution obtained from a local (S)MC algorithm as proposal distribution of a standard or fully-adapted APF algorithm. A quite similar algorithm was previously proposed in [14] in order to solve a complex filtering problem. In this so-called Nested SMC (NSMC) framework, a local (S)MC algorithm is used for each particle path-space $\mathbf{x}_{0:k-1}^i$ in order to obtain an empirical approximation of the optimal importance distribution and an approximation of the importance weights. Our proposition differs from such contributions in the sense that our aim is to ensure a set of (conditionally) independent resampled particles at the end of each iteration. With the NSMC, such an independence property is clearly not ensured since the sampling stage consists of generating N particles from an empirical measure (i.e. a discrete approximation based on a finite number of local particles from the nested algorithm) that could be identical for several draws. As a consequence, there is a non-null probability to select exactly the same local particle for different global particles. Finally, as discussed previously, our strategy can be used only when this is necessary, thus leading to an efficient and easily implementable adaptive framework.

4. SIMULATIONS

In this section we compare our approach with the classical particle filter in the framework of target tracking with range-bearing measurements. The state-vector (position and velocity in Cartesian coordinates) is defined as $\mathbf{x}_k = [p_{x,k}, \dot{p}_{x,k}, p_{y,k}, \dot{p}_{y,k}]^T$ and estimated from range-bearing measurements $\mathbf{y}_k = [\rho_k, \theta_k]^T$. The target state evolution and observation equations read

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{u}_k \\ \mathbf{y}_k &= \begin{pmatrix} \sqrt{p_{x,k}^2 + p_{y,k}^2} \\ \arctan \frac{p_{y,k}}{p_{x,k}} \end{pmatrix} + \mathbf{v}_k \end{aligned}$$

where $\mathbf{F}_k = \begin{pmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{pmatrix}$, $\mathbf{x}_0$, $\mathbf{u}_1$, $\cdots$, $\mathbf{u}_k$, $\mathbf{v}_0$, $\cdots$, $\mathbf{v}_k$ are independent with $\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}_4, \mathbf{Q})$, $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}_2, \mathbf{R})$,

$$\mathbf{Q} = \sigma_Q^2 \begin{pmatrix} \frac{\tau^3}{3} & \frac{\tau^2}{2} & 0 & 0 \\ \frac{\tau^2}{2} & \tau & 0 & 0 \\ 0 & 0 & \frac{\tau^3}{3} & \frac{\tau^2}{2} \\ 0 & 0 & \frac{\tau^2}{2} & \tau \end{pmatrix}, \mathbf{R} = \begin{pmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_\theta^2 \end{pmatrix},$$

$\tau = 1$. We compare the performances of estimates of interest $\hat{\mathbf{x}}_k$ by computing their averaged root mean square error (RMSE) w.r.t. $\mathbf{x}_k$ on $N_{\mathrm{MC}} = 100$ simulations,

$$\mathrm{RMSE}_k = \sqrt{\frac{1}{N_{\mathrm{MC}}} \sum_{n=1}^{N_{\mathrm{MC}}} \|\hat{\mathbf{x}}_k - \mathbf{x}_k\|^2} \qquad (11)$$

4.1. Independent resampling with $M = N$

We first intend to confirm that independent resampling gives better performance than classical resampling for a given number of particles N. We use $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{0:k}) = f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$ and we resample at each time step.

For $M = N = 500$ particles over $T = 100$ time steps, $\tau = 1$, $\sigma_Q = \sqrt{10}$, $\sigma_\rho = 1$ and $\sigma_\theta = \frac{\pi}{180}$, $N_{MC} = 100$, the averaged RMSEs are displayed in Fig. 1. As expected this result shows that the performance of independent resampling is significantly better than dependent resampling; moreover the performance improvement increases over time which is due in

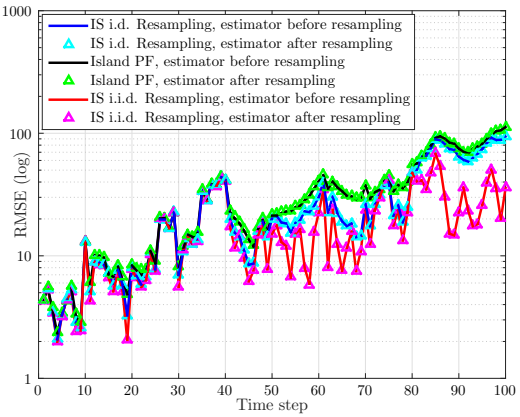

Fig. 1. Example of RMSE performance between PF with dependent and independent resampling with same number of particles $M = N = 500$. Estimates before and after resampling coincide. For comparison, the RMSE of a basic island particle filter (Algorithm (1) from [12]) with 5 islands containing 100 particles each is shown as well.

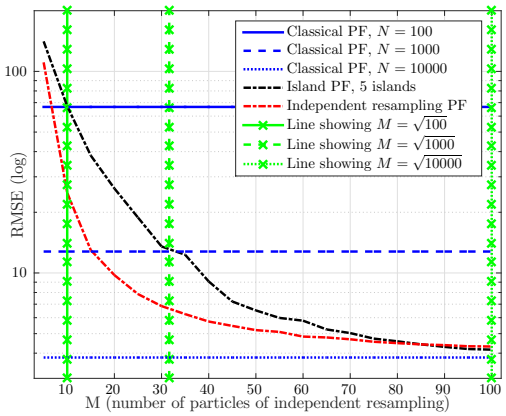

Fig. 2. Example of RMSE performance between an independent resampling particle filter with varying number of particles M, and classical particle filters with $N = 100, 1000, 10000$. The green vertical lines show values of $\sqrt{100}, \sqrt{1000}, \sqrt{10000}$ in order to see the performance of the independent resampling particle filter when $M = \sqrt{N}$. For comparison, the RMSE of a basic island particle filter (Algorithm (1) from [12]) with 5 islands containing $\frac{M^2}{5}$ particles each is shown as well.

this simulation to the high process noise; the gain in particle diversity due to our resampling method enables to better track the target in difficult scenarios. Conversely, other tests have shown that for $\sigma_Q = 1$ or less the difference between the two resampling methods vanishes. This is because in such a scenario, the trajectory becomes linear so the estimation is well performed by a classical particle filter.

4.2. Independent resampling with varying number of particles

On the other hand, we consider a fixed number of samples N and we perform our independent resampling method with $M = \sqrt{N}$ in scenarios where the degeneration phenomenon affects the classical particle filter. In particular, this phenomenon appears when $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{0:k}) = f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$ and the variance of the measurement noise is small compared to that of the process noise. It has been observed that our independent resampling technique can produce better performance at a lower number of samples. In Fig. 2 we have displayed the performance of our independent resampling technique as a function of M, with $\sigma_Q = \sqrt{10}$, $\sigma_\rho = 0.05$ and $\sigma_\theta = \frac{\pi}{3600}$. Independent resampling performs better at $M = \sqrt{N}$ for $N = 100$ and $N = 1000$, and converges quickly as M increases to the same performance as $N = 10000$. These results are mainly due to the tendency of classical resampling to duplicate only one particle when their weights are degenerated, while independent resampling ensures that the algorithm duplicates M of them, keeping a better particle diversity.

5. CONCLUSION

In this paper we proposed to modify the resampling step of PF in order to produce particles which, given the weighted set of particles at previous instant, are still drawn from the same pdf as those produced by classical multinomial resampling, but are moreover independent. We discussed independent vs. dependent resampling, proposed an implementation of our resampling technique, and validated our results by simulations.

6. REFERENCES

[1] Y.C. Ho and R. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Transactions on Automatic Control*, vol. 9, pp. 333–339, 1964.

[2] M. Sanjeev Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear / non-Gaussian Bayesian tracking," *IEEE Tr. Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[3] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science. Springer, New York, 2001.

[4] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[5] R. Douc, O. Cappé, and É. Moulines, "Comparison of resampling schemes for particle filtering," in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Zagreb, Croatia, September 2005.

[6] J. D. Hol, T. B. Schön, and F. Gustafsson, "On resampling algorithms for particle filtering," in *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop (NSSPW)*, Cambridge, UK, September 2006.

[7] T. Li, M. Bolić, and P. M. Djuric, "Resampling Methods for Particle Filtering: Classification, implementation, and strategies," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 70–86, May 2015.

[8] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 278–88, March 1994.

[9] J. S. Liu and R. Chen, "Blind deconvolution via sequential imputation," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 567–76, June 1995.

[10] O. Cappé, É. Moulines, and T. Rydén, *Inference in Hidden Markov Models*, Springer-Verlag, 2005.

[11] Y. Petetin and F. Desbouvries, "Optimal SIR algorithm vs. fully adapted auxiliary particle filter: a non asymptotical analysis," *Statistics and computing*, vol. 23, no. 6, pp. 759–775, 2013.

[12] C. Vergé, C. Dubarry, P. Del Moral, and É. Moulines, "On parallel implementation of sequential Monte Carlo methods: the island particle model," *Statistics and Computing*, vol. 25, no. 2, pp. 243–260, 2015.

[13] C. A. Naesseth, F. Lindsten, and T. B. Schön, "Nested Sequential Monte Carlo Methods," in *32nd International Conference on Machine Learning (ICML)*, Lille, France, July 2015.

[14] N. Jaoua, E. Duflos, P. Vanheeghe, and F. Septier, "Bayesian nonparametric state and impulsive measurement noise density estimation in nonlinear dynamic systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 5755–5759.