

# Variance Reduction of Resampling for Sequential Monte Carlo

Xiongmeng Dai<sup>a</sup> and Gerald Baumgartner<sup>b</sup>

*Division of Computer Science and Engineering, Louisiana State University, Baton Rouge, 70803, LA, U.S.A.*

**Keywords:** Resampling, Sequential Monte Carlo, Hidden Markov Model, Repetitive Ergodicity, Deterministic Domain.

**Abstract:** A resampling scheme provides a way to switch low-weight particles for sequential Monte Carlo with higher-weight particles representing the objective distribution. The less the variance of the weight distribution is, the more concentrated the effective particles are, and the quicker and more accurate it is to approximate the hidden Markov model, especially for the nonlinear case. Normally the distribution of these particles is skewed, we propose repetitive ergodicity in the deterministic domain with the median for resampling and have achieved the lowest variances compared to the other resampling methods. As the size of the deterministic domain  $M \ll N$  (the size of population), given a feasible size of particles under mild assumptions, our algorithm is faster than the state of the art, which is verified by theoretical deduction and experiments of a hidden Markov model in both the linear and non-linear cases.


## 1 INTRODUCTION


Sequential Monte Carlo (SMC) or Particle Filter (Gordon et al., 1993) is a set of Monte Carlo methods for solving nonlinear state-space models given noisy partial observations, which are widely used in signal and image processing (Särkkä et al., 2007), stock analysis (Casarin et al., 2006; Flury and Shephard, 2011; Dai and Baumgartner, 2023a; Dai and Baumgartner, 2023d), Bayesian inference (Del Moral et al., 2006; Dai and Baumgartner, 2023b; Dai and Baumgartner, 2023c) or robotics (Fox, 2001; Montemerlo et al., 2002; Thrun, 2002). It updates the predictions recursively by samples composed of weighted particles to infer the posterior probability density. While the particles will be impoverished as the sample forwards recursively, it can be mitigated by resampling where the negligible weight particles will be replaced by other particles with higher weights (Doucet et al., 2000; Del Moral et al., 2012; Doucet et al., 2001).

In the literature, several resampling methods and corresponding theoretical analysis (Künsch et al., 2005; Chopin et al., 2004; Douc and Moulines, 2007; Gilks and Berzuini, 2001) can be found. The frequently used algorithms are residual resampling (Liu and Chen, 1998), multinomial resampling (Gordon et al., 1993), stratified resampling (Smith, 2013),

and systematic resampling (Kitagawa, 1996; Arulampalam et al., 2002). A justified decision regarding which resampling strategies to use might result in a reduction of the overall computation effort and high accuracy of the estimations of the objective. However, for resampling, most of these strategies traverse repetitively from the original population, the negligible weight particles fail to be discarded completely, although the diversity of the particle reserve, it causes unnecessary computational load for a very heavily skewed distribution of particles and affects the accuracy of estimations of the posterior distribution. From the perspective of complexity and variance reduction with promising estimation, we propose a repetitive deterministic domain ergodicity strategy, where more concentrated and effective particles are drawn to approximate the objective. Our proposal can be widely used in large-sample approximations.

In this paper, we concentrate on the analysis of the importance sample resamplings built-in SMC for the hidden Markov model. In Section 2, we present a brief introduction to SMC. Here, a brief introduction to the hidden Markov model and the sequential importance sampling method will be given. Our method will be introduced in Section 3, where we introduce the origin of our method, and how to implement each step in detail, and then the theoretical asymptotic behavior of approximations using our method is provided. The practical experiments will be validated by Section 4, where performance and complexity analy-

<sup>a</sup>  <https://orcid.org/0009-0001-8540-3276>

<sup>b</sup>  <https://orcid.org/0009-0007-1162-159X>

sire are presented. The summary of our contributions is outlined in Section 5.

## 2 RESAMPLING IN SMC FOR HIDDEN MARKOV MODEL

Consider the state-space model, which is also known as a hidden Markov model, described by

$$X_t | X_{t-1} \sim f(X_t | X_{t-1}), \quad Y_t | X_t \sim g(y_t | X_t). \quad (1)$$

The initial state  $X_0 \sim \mu(X_0)$ ,  $X_t, (t = 1, 2, \dots, n)$  is a latent variable to be observed, the measurements  $Y_t$  are assumed to be conditionally independent given  $X_t$ , the most objective is to estimate  $X_t$ .

The recursive Bayesian estimation can be used and it is described as:

(a) Prediction:

$$\pi(X_t | y_{1:t-1}) = \int f(X_t | X_{t-1}) \pi(X_{t-1} | y_{1:t-1}) dX_{t-1}. \quad (2)$$

(b) Update:

$$\pi(X_t | y_{1:t}) = \frac{g(y_t | X_t) \pi(X_t | y_{1:t-1})}{\int g(y_t | X_t) \pi(X_t | y_{1:t-1}) dX_t}. \quad (3)$$

From (2) and (3) the integral part is unreachable, especially, for high-dimensional factors involved, we fail to get the close form of  $\pi(X_t | y_{1:t})$  (Särkkä, 2013; Doucet and Johansen, 2009).

Sequential Monte Carlo is a recursive algorithm where a cloud of particles are propagated to approximate the posterior distribution  $\pi(X_{0:t} | y_{1:t})$ . Here, we describe a general algorithm that generates at time  $t$ ,  $N$  particles  $\{X_{0:t}^{(i)}\}_{i=1}^N$  with the corresponding empirical measure  $\hat{\pi}(X_{0:t} | y_{1:t}) = \sum_{i=1}^N w_t^i \delta_{X_{0:t}^{(i)}}(dX_{0:t})$ , a discrete weighted approximation of the true posterior  $\pi(X_{0:t} | y_{1:t})$ ,  $\delta_{X_{0:t}^{(i)}}(dX_{0:t})$  denotes the delta-Dirac mass located at  $X_t$ ,  $dX_{0:t}$  equals to  $X_{0:t} - X_{0:t}^i$ . The particles are drawn recursively using the observation obtained at time  $t$  and the set of particles  $\{X_{0:t-1}^{(i)}\}_{i=1}^N$  drawn at time  $t-1$ , accordingly, where  $\hat{\pi}(X_{0:t-1} | y_{1:t-1}) \approx \pi(X_{0:t-1} | y_{1:t-1})$ . The weights are normalized using the principle of importance sampling such that  $\sum_{i=1}^N w_t^i = 1$ . If the samples  $X_{0:t}^i$  are drawn from an importance density  $q(X_{0:t}^i | y_{1:t})$ , we have

$$w_t^i \propto \frac{\pi(X_{0:t}^i | y_{1:t})}{q(X_{0:t}^i | y_{1:t})}. \quad (4)$$

Suppose at time step  $t-1$ , we have existed samples to approximate the current posterior distribution

$\pi(X_{0:t-1} | y_{1:t-1})$ , if we get a new observation  $y_t$  at time  $t$ , a recursive approximation to  $\pi(X_{0:t} | y_{1:t})$  with a new set of samples can be obtained by importance sampling, the corresponding factorization (Arulampalam et al., 2002) is described by

$$q(X_{0:t} | y_{1:t}) := q(X_t | X_{0:t-1}, y_{1:t}) q(X_{0:t-1} | y_{1:t-1}). \quad (5)$$

Then, we can get the new samples  $X_{0:t}^i \sim q(X_{0:t} | y_{1:t})$  by propagating each of the existing samples  $X_{0:t-1}^i \sim q(X_{0:t-1} | y_{1:t-1})$  with the new state  $X_t^i \sim q(X_t | X_{0:t-1}, y_t)$ . To derive the weight update equation, we follow the ergodic Markov chain properties of the model, the full posterior distribution  $\pi(X_{0:t} | y_{1:t})$  can be written recursively in terms of  $\pi(X_{0:t-1} | y_{1:t-1})$ ,  $g(y_t | X_t)$  and  $f(X_t | X_{t-1})$  (Arulampalam et al., 2002):

$$\pi(X_{0:t} | y_{1:t}) = \frac{p(y_t | X_{0:t}, Y_{1:t-1}) p(X_{0:t} | y_{1:t-1})}{p(y_t | y_{1:t-1})},$$

$$\pi_{1:t} \propto g(y_t | x_t) f(x_t | x_{t-1}) \pi_{1:t-1}, \quad (6)$$

where  $\pi_{1:t}$  is short for  $\pi(X_{0:t} | y_{1:t})$ . By substituting (5) and (6) into (4), we have

$$w_t^i \propto \frac{g(y_t | X_t^i) f(X_t^i | X_{t-1}^i) p(X_{0:t-1}^i | y_{1:t-1})}{q(X_t^i | X_{0:t-1}^i, Y_{1:t}) q(X_{0:t-1}^i | Y_{1:t-1})}$$

$$= w_{t-1}^i \frac{g(y_t | X_t^i) f(X_t^i | X_{t-1}^i)}{q(X_t^i | X_{0:t-1}^i, Y_{1:t})}. \quad (7)$$

We assume the state  $X_t$  is ergodic Markovian, thus,  $q_t = q(X_t^i | X_{0:t-1}^i, Y_{1:t}) = q(X_t^i | X_{t-1}^i, Y_{1:t})$ , from this point, we only need to store the  $X_t^i$ , and obtain the thinning recursively update weight formula (Gordon et al., 2004):

$$w_t^i \propto w_{t-1}^i \frac{g(y_t | x_t^i) f(x_t^i | x_{t-1}^i)}{q(x_t^i | x_{t-1}^i, y_t)}. \quad (8)$$

The corresponding empirical posterior filtered density  $\pi(X_t | y_{1:t})$  can be approximated as

$$\hat{\pi}(X_t | y_{1:t}) = \sum_{i=1}^N w_t^i \delta_{X_t^i}(dX_t). \quad (9)$$

It can be shown that as  $N \rightarrow \infty$ ,  $\hat{\pi}(X_t | y_{1:t})$  converges to  $\pi(X_t | y_{1:t})$ .

Ideally, the importance density function should be the posterior distribution itself,  $\pi(X_{0:t} | y_{1:t})$ . The variance of importance weights increases over time, which will decrease the accuracy and lead to degeneracy that some particles make up negligible normalized weights. The brute force approach to reducing the effect of degeneracy is to increase  $N$  as large as possible. However, as the size of the sample increases, the computation of the recursive step will

also be exponentially costly. Generally, we can try two ways to improve: (I) suitable importance density sampling; (II) resampling the weights. Here we focus on the latter. A suitable measure of the degeneracy of an algorithm is the effective sample size  $N_{eff}$  introduced in (Arulampalam et al., 2002):  $N_{eff} = \frac{N}{1 + \text{Var}(w_t^{*i})}$ ,  $w_t^{*i} = \frac{\pi(X_t^i | y_{1:t})}{q(X_t^i | X_{t-1}^i, y_t)}$ , while the close solution is unreachable, it could be approximated (Liu, 2008) by  $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (w_t^i)^2}$ . If the weights are uniform,  $w_t^i = \frac{1}{N}$  for each particle  $i = 1, 2, \dots, N$ ,  $N_{eff} = N$ ; If there exists the unique particle, whose weight is 1, the remaining are zero,  $N_{eff} = 1$ . Hence, small  $N_{eff}$  easily leads to a severe degeneracy (Gordon et al., 2004). We use  $\hat{N}_{eff}$  as an indicator to measure the condition of resampling for our experiments in section 4.

We will introduce our proposal based on the repetitive deterministic domain traverse in the next section.

### 3 REPETITIVE ERGODICITY IN DETERMINISTIC DOMAIN WITH MEDIAN FOR RESAMPLING

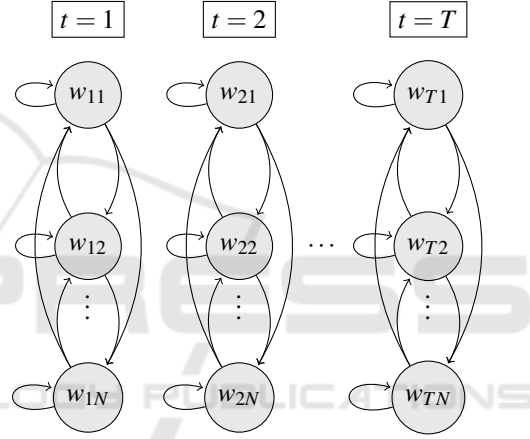
#### 3.1 Multinomial Sampling

A Multinomial distribution provides a flexible framework with parameters  $p_i, i = 1, \dots, k$  and  $N$ , to measure the probability that each class  $i \in 1, \dots, k$  has been sampled  $N_i$  times over  $N$  categorical independent tests. It can be used to resample the tag in our proposal in two steps. Firstly, we obtain the samples from a uniform generator  $u^i \sim U(0, 1], i = 1, \dots, N$ ; secondly, we evaluate the index  $j$  of samples with the generalized inverse rule, if the cumulative sum of samples  $\sum_{i=1}^j w_i$  larger or equal to  $u^i$ , this index  $j$  will be labeled, then the corresponding sample  $w_i$  will be resampled, this event can be mathematically termed as  $g'(w_i) = \mathbb{I}_{w_i=w_j}$ .

#### 3.2 Deterministic Domain Construction

The population of weights is divided into two parts. The first part is the weights, larger than the average  $\frac{1}{N}$ , they are considered as the candidate firstly to be sampled, we keep  $r_i = \lfloor N\hat{w}_i^i \rfloor$  replicates of  $\hat{w}_i^i$  for each  $i$ , where  $\hat{w}_i^i$  is the renormalized unit.  $r_i$  will be filtered one by one from the population, and the corresponding tag  $j$  will be saved into an array. We find, this part also follows the multinomial

distribution  $W^i \sim \text{Multinomial}(M; \hat{w}^1, \dots, \hat{w}^M)$ , We extract the samples from the population with the rule of multinomial sampling shown in section 3.1. This step is the first layer of the traverse from the population, we achieve the ancestor subset. Since the distribution of these particles is very heavily skewed, we extract the median of these particles to add to the set of descendant particles. Then, we renormalized the weights in the subset, and traverse again to differentiate the larger weights and other units, until we get the feasible size of the set to be considered as the potential deterministic domain. The particles in this domain follow a very strict Markovian, the evolution can be expressed as follows, where  $w_{tN}$  denotes the weight corresponding to the particle  $\hat{X}_t^{(N)}$  at  $t$ .



We define the integer part event,  $g'(\hat{w}_i) = \mathbb{I}_{\hat{w}_i=\hat{w}_j}$ , similarly for the following repetitive part,  $\bar{g}(\bar{w}_i) = \mathbb{I}_{\bar{w}_i=\bar{w}_j}$ . We count the units involved in the occurrence of the event  $g'(\hat{w}_i)$  and  $\bar{g}(\bar{w}_i)$ , then extract these units based on the tags  $j$ , which forms the final deterministic domain.

#### 3.3 Repetitive Ergodicity in Deterministic Domain with Median Schema

Our goal is to retract and retain units with large weights, while the remaining ones with low weights can be effectively replaced in the populations. We set the desired number of resampled units as the size of populations under the premise of ensuring unit diversity as much as possible.

We normalized all the units to keep the same scaled level for comparison, after that, the units with higher weights above the average level will appear as real integers (larger than zero) by  $Ns = \text{floor}(N \cdot w)$ ,

the remaining will be filtered to zero. This is the prerequisite for the deterministic domain construction. In  $Ns$  subset, there exist multiple categorical units, that follow the multinomial distribution. We sample these termed large units with two loops, the outer loop is to bypass the index of the unit zero, and the inner loop is to traverse and sample the subset where different large units distribute, there more large weights will be sampled multiple times.

The last procedure is to repetitively traverse in the deterministic domain with median, where each unit will be renormalized and the corresponding cumulative summation is used to find the index of the unit with the rule of the inverse cumulative distribution function. Each desired unit will be drawn by the multinomial sampler to rejuvenate the population recursively. The complexity of our method is  $O(M)$ . As the size of the deterministic domain  $M \ll N$  (the size of population), given a feasible size of particles, our algorithm is faster than the state of the art. The total implement schema is shown in Algorithm 1.

### 3.4 Theoretical Asymptotic Behavior of Approximations

#### 3.4.1 Central Limit Theorem

Suppose that for each  $t \in [1, T]$ ,  $\tilde{X}_t^{(1)}, \dots, \tilde{X}_t^{(M)}$  are independent, where  $\tilde{X}_t^{(m')}, m' \in [1, M]$  denotes the median of the originator particles. For others  $\tilde{X}_t^{(i)}, i \neq m'$  belong to the deterministic domain; the probability space of the sequence recursively changes with  $t$  for sequential Monte Carlo, such a collection is called a triangular array of particles. Let  $S_m := \tilde{X}_t^{(1)} + \dots + \tilde{X}_t^{(M)}$ . We expand the characteristic function of each  $\tilde{X}_t^{(i)}$  to second-order terms and estimate the remainder to establish the asymptotic normality of  $S_m$ . Suppose that both the expectation and the variance exist and are finite:

$$\begin{aligned} \mathbb{E}_{\pi_t}(X_t^{(i)}) &= \int_{X_t \in \Omega} X_t^{(i)} \pi_t(dX_t) q(X_t, dX_{t+1}) < \infty, \\ \delta_{t,i}^2(X_t^{(i)}) &= \mathbb{E}[(X_t^{(i)} - \mathbb{E}(X_t^{(i)}))^2] < \infty. \end{aligned} \quad (10)$$

**Theorem 1** For each  $t$  the sequence  $\tilde{X}_t^{(1)}, \dots, \tilde{X}_t^{(M)}$  sampled from the originator particles  $X_t^{(1)}, \dots, X_t^{(N)}$ , suppose that are independent, where  $\tilde{X}_t^{(m')}, m' \in [1, M]$  denotes the median of the originator particles. For the rest  $\tilde{X}_t^{(i)}, i \neq m'$  belong to the deterministic domain; let  $\Psi$  be a measurable function and assume that there

**Input:** The input weight sequence:  $w$ ; the desired number of resampled particles:  $N$

**Output:** The resampled tag of the weight sequence  $tag$

```

if  $nargin == 1$  then
    |  $N \leftarrow length(w)$ ; // Desired size
end
 $M \leftarrow length(w)$ ;
 $w \leftarrow w/sum(w)$ ; // Normalization
 $tag \leftarrow zeros(1, N)$ ;
 $Ns \leftarrow floor(N.*w)$ ; // Integer parts
 $R \leftarrow sum(Ns)$ ;
 $i \leftarrow 1$ ; // Extract deterministic part
 $j \leftarrow 0$ ;
while  $j < M$  do
    |  $j \leftarrow j + 1$ ;
    |  $count \leftarrow 1$ ;
    | while  $count \leq Ns(j)$  do
    | |  $tag(i) \leftarrow j$ ;
    | |  $i \leftarrow i + 1$ ;
    | |  $count \leftarrow count + 1$ ;
    | end
end
 $[W, I] = sort(w)$ ; // Median extraction
 $r = floor((N + 1)/2)$ ;
 $tag(i) = I(r)$ ;
 $i = i + 1$ ; // Deterministic domain
 $w \leftarrow tag/sum(tag)$ ;
 $q \leftarrow cumsum(w)$ 
while  $i \leq N$  do
    |  $sampl \leftarrow rand$ ;
    |  $j \leftarrow 1$ ;
    | while  $q(j) < sampl$  do
    | |  $j \leftarrow j + 1$ ; // Update the tag
    | | end
    |  $tag(i) \leftarrow tag(j)$ ;
    |  $i \leftarrow i + 1$ ;
end
    
```

Algorithm 1: Repetitive Deterministic Domain with Median Traversal Resampling.

exists  $\tilde{\mathbf{X}}_t \subset \mathfrak{R}$  satisfying

$$\int_{x \in \mathfrak{R}} \pi(dx) \mathbb{E}_x \left[ \sum_{t=1}^T |\Psi(\mathbf{X}_t)|^{2+\varepsilon} \right] < \infty \quad (11)$$

and

$$\sup_{x \in \mathfrak{R}} \mathbb{E}_x \left[ \sum_{t=1}^T |\Psi(\mathbf{X}_t)| \right] < \infty, \quad (12)$$

$$\mathbb{E}_{\pi_t}[\Psi] := \int_{\mathfrak{R}} \pi(dx) \mathbb{E}_x \left[ \sum_{i=1}^N \Psi(\mathbf{X}^{(i)}) \right] < \infty.$$

If  $\tilde{\mathbf{X}}_t$  is aperiodic, irreducible, positive Harris recurrent with invariant distribution  $\pi$  and geometrically

ergodic, and if, in addition,

$$\begin{aligned}\delta_{t,i}^2(\Psi) &:= \int \pi(dx) \mathbb{E}_x \left[ \left( \Psi(\tilde{X}_t^{(i)}) - \mathbb{E}_{\pi_t}[\Psi] \right)^2 \right] < \infty, \\ s_m^2 &:= \lim_{M \rightarrow \infty} \sum_{i=1}^M \delta_{t,i}^2(\Psi),\end{aligned}\quad (13)$$

$\{\Psi(\tilde{X}_t^{(i)})\}$  satisfies

$$\lim_{M \rightarrow \infty} \sum_{i=1}^M \left\{ \Psi(\tilde{X}_t^{(i)}) - \mathbb{E}_{\pi_t}[\Psi] \right\} \sim N(0, s_m^2). \quad (14)$$

**Proof** Let  $Y_{t,i} = \Psi(\tilde{X}_t^{(i)}) - \mathbb{E}_{\pi_t}[\Psi]$ , by (Billingsley, 1995),  $\left| e^{iy} - \sum_{k=0}^M \frac{(iy)^k}{k!} \right| \leq \min\left\{ \frac{(y)^{M+1}}{(M+1)!}, \frac{2(y)^M}{M!} \right\}$ , when  $M = 2$ , we have

$$\left| e^{iy} - \left( 1 + iy - \frac{1}{2}y^2 \right) \right| \leq \min\left\{ \frac{1}{6}|y|^3, |y|^2 \right\}. \quad (15)$$

We first assume that  $\Psi(\cdot)$  is bounded, from the property of characteristic function, the left-hand side can be written as  $\left| \mathbb{E} \left[ e^{i\lambda Y_{t,i}} | \mathfrak{R} \right] - \left( 1 - \frac{\lambda^2 \delta_{t,i}^2(\Psi)}{2} \right) \right|$ , therefore, the corresponding character function  $\phi_{t,i}(\lambda)$  of  $Y_{t,i}$  satisfies

$$\left| \phi_{t,i}(\lambda) - \left( 1 - \frac{\lambda^2 \delta_{t,i}^2(\Psi)}{2} \right) \right| \leq \mathbb{E} \left[ \min\left\{ |\lambda Y_{t,i}|^2, \frac{1}{6} |\lambda Y_{t,i}|^3 \right\} \right]. \quad (16)$$

Note that the expected value exists and is finite, the right-hand side term can be integrated by

$$\int_{|Y_{t,i}| \geq \varepsilon \delta_{t,i} \sqrt{M}} \mathbb{E} \left[ \min\left\{ |\lambda Y_{t,i}|^2, \frac{1}{6} |\lambda Y_{t,i}|^3 \right\} \right] dx. \quad (17)$$

As  $M \rightarrow +\infty, \{Y_{t,i}\} \rightarrow \mathbf{0}$ , then,  $E \left[ \min\left\{ |\lambda Y_{t,i}|^2, \frac{1}{6} |\lambda Y_{t,i}|^3 \right\} \right] \rightarrow 0$ , which satisfies Lindeberg condition:

$$\lim_{M \rightarrow \infty} \sum_{i=1}^M \frac{1}{s_m^2} \int_{|Y_{t,i}| \geq \varepsilon \delta_{t,i} \sqrt{M}} Y_{t,i}^2 dX = 0, \quad (18)$$

for  $\varepsilon > 0, s_m^2 = \sum_{i=1}^M \delta_{t,i}^2(\Psi)$ .

$$\lim_{M \rightarrow \infty} \left| \phi_{t,i}(\lambda) - \left( 1 - \frac{\lambda^2 \delta_{t,i}^2(\Psi)}{2} \right) \right| = 0. \quad (19)$$

By (26.11) in page 345 (Billingsley, 1995),

$$\phi_{t,i}(\lambda) = 1 + i\lambda \mathbb{E}[X] - \frac{1}{2} \lambda^2 \mathbb{E}[X^2] + o(\lambda^2), \lambda \rightarrow 0. \quad (20)$$

By Lemma 1 in page 358 (Billingsley, 1995),

$$\begin{aligned}& \left| \prod_{i=1}^M e^{-\lambda^2 \delta_{t,i}^2(\Psi)/2} - \prod_{i=1}^M \left( 1 - \frac{1}{2} \lambda^2 \delta_{t,i}^2(\Psi) \right) \right| \\ & \leq \sum_{i=1}^M \left| e^{-\lambda^2 \delta_{t,i}^2(\Psi)/2} - 1 + \frac{1}{2} \lambda^2 \delta_{t,i}^2(\Psi) \right| \\ & \leq \sum_{i=1}^M \left[ \frac{1}{4} \lambda^4 \delta_{t,i}^4(\Psi) \sum_{j=2}^{\infty} \frac{\frac{1}{2^{j-2}} \lambda^{2j-4} \delta_{t,i}^{2j-4}(\Psi)}{j!} \right] \\ & \leq \sum_{i=1}^M \frac{1}{4} \lambda^4 \delta_{t,i}^4(\Psi) e^{|\frac{1}{2} \lambda^2 \delta_{t,i}^2(\Psi)|}.\end{aligned}\quad (21)$$

Thus,

$$\begin{aligned}\prod_{i=1}^M e^{-\lambda^2 \delta_{t,i}^2(\Psi)/2} &= \prod_{i=1}^M \left( 1 - \frac{1}{2} \lambda^2 \delta_{t,i}^2(\Psi) \right) + o(\lambda^2) \\ &= \prod_{i=1}^M e^{-\lambda^2 \delta_{t,i}^2(\Psi)/2} + o(\lambda^2) \\ &= e^{-\frac{\lambda^2 s_m^2}{2}} + o(\lambda^2).\end{aligned}\quad (22)$$

The characteristic function  $\prod_{i=1}^M \phi_{t,i}(\lambda)$  of  $\sum_{i=1}^M Y_{t,i} = \sum_{i=1}^M \left\{ \Psi(\mathbf{X}_t^{(i)}) - \mathbb{E}_{\pi_t}[\Psi] \right\}$  is equal to  $e^{-\frac{\lambda^2 s_m^2}{2}}$ , thus, (14) holds.

## 4 EXPERIMENTS

In this part, the results of the comparison of these resampling methods are validated from the experiments with the linear Gaussian state space model and non-linear state space model, respectively. We ran the experiments on an HP Z200 workstation with an Intel Core i5 and an #82 – 18.04.1 – Ubuntu SMP kernel.

### 4.1 Linear Gaussian State Space Model

This linear model is expressed by:

$$\begin{aligned}X_0 &\sim \mu(X_0), \\ X_t | X_{t-1} &\sim N(X_t; \phi X_{t-1}, \delta_v^2), \\ Y_t | X_t &\sim N(y_t; X_t, \delta_e^2),\end{aligned}\quad (23)$$

we keep parameters the same as (Dahlin and Schön, 2019) to compare with the different resampling methods. Where  $\theta = \{\phi, \delta_v, \delta_e\}, \phi \in (-1, 1)$  describes the persistence of the state, while  $\delta_v, \delta_e$  denote the standard deviations of the state transition noise and the observation noise, respectively. The Gaussian density is denoted by  $N(x; \mu, \delta^2)$  with mean  $\mu$  and standard deviation  $\delta > 0$ . In Figure 1, we use 20 particles to track the probability distribution of the state,

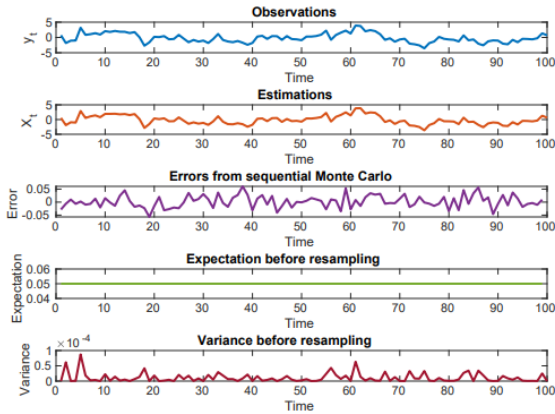


Figure 1: Repetitive deterministic traverse resampling for linear Gaussian state space model.

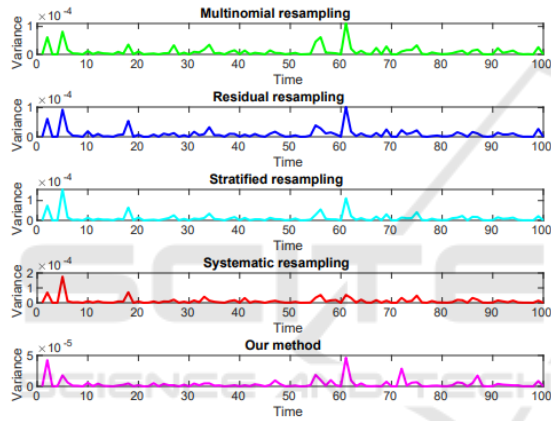


Figure 2: Variance analysis for different resampling strategies (note the different scales in the Variance axis).

composed of 100 different times, the ground truth is from the Kalman filter (Welch et al., 1995), the error denotes the difference between the estimation by SMC and the ground truth. Initially, the expectation of weights for 20 particles is equal to  $\frac{1}{20}$ , which means that these particles have equal functions to track the state.

For the resampling procedure, we compare the variance from different classical resampling methods, shown in Figure 2. The variance from the deterministic traverse method is the smallest. Thus, the effective particles are more concentrated after resampling based on our proposal.

The computational complexity is another factor the resampling algorithms are compared on, Figure 3 shows the execution times for different particles distributed, generally, it depends on the machines and random generator, during our simulations, the time consumption is different under the same condition of resampling method and number of particles. Further-

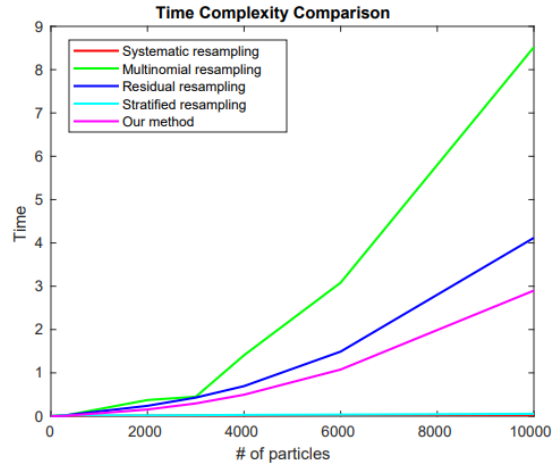


Figure 3: Time complexity analysis for different resampling strategies.

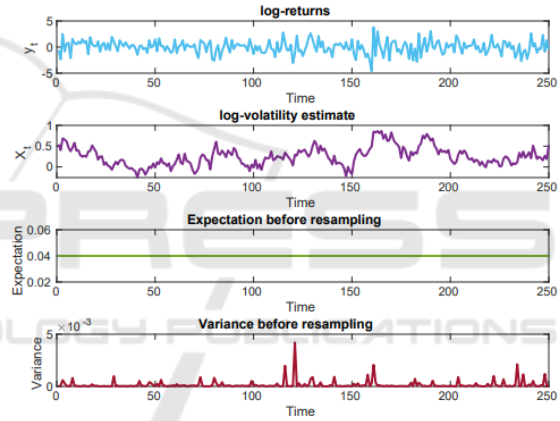


Figure 4: (a) The daily log-returns. (b) The estimated log-volatility with 95% confidence intervals of the NASDAQ OMXS30 index for the period from January 2, 2015 to January 2, 2016. (c) The expectation of weights for particles before resampling. (d) The variance of weights for particles before resampling.

more, we find under the same resampling methods, the time consumed for the small size of particles is much more than that of the larger ones. The computational stability of particles with resampling methods is very sensitive to the units from a specific population. For safety, we conduct multiple experiments to achieve the general complexity trend.

In Figure 3, all the experiments are conducted under the same conditions, for large-size particles, the stratified and systematic strategies are favorable. In Table 1, we can find under small-size particles (less than 150), our method performs best.

Table 1: Time complexity analysis of different resampling strategies.

| # of particles | Multinomial resampling | Residual resampling | Systematic resampling | Stratified resampling | <b>Our method</b> |
|----------------|------------------------|---------------------|-----------------------|-----------------------|-------------------|
| 5              | 0.0026                 | 0.0021              | 0.0019                | 0.0023                | <b>0.0016</b>     |
| 15             | 0.0036                 | 0.0025              | 0.0022                | 0.0033                | <b>0.0018</b>     |
| 50             | 0.0057                 | 0.0027              | 0.0024                | 0.0047                | <b>0.0022</b>     |
| 80             | 0.0087                 | 0.0032              | 0.0029                | 0.0051                | <b>0.0027</b>     |
| 100            | 0.0127                 | 0.0038              | 0.0034                | 0.0067                | <b>0.0030</b>     |
| 150            | 0.0161                 | 0.0043              | 0.0038                | 0.0088                | <b>0.0036</b>     |

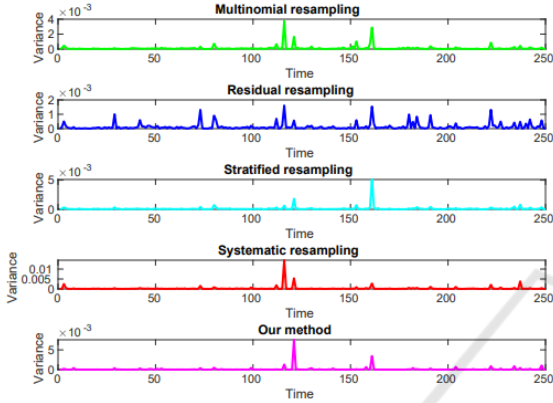


Figure 5: Variance analysis of different resampling strategies for nonlinear state space model (note the different scales in the Variance axis).

## 4.2 Nonlinear State Space Model

We continue with a real application of our proposal to track the stochastic volatility, a nonlinear State Space Model with Gaussian noise, where log volatility considered as the latent variable is an essential element in the analysis of financial risk management. The stochastic volatility is given by

$$\begin{aligned}
 X_0 &\sim N(\mu, \frac{\sigma_v^2}{1-\rho^2}), \\
 X_t | X_{t-1} &\sim N(\mu + \rho(X_{t-1} - \mu), \sigma_v^2), \\
 Y_t | X_t &\sim N(0, \exp(X_t)\tau),
 \end{aligned} \tag{24}$$

where the parameters  $\theta = \{\mu, \rho, \sigma_v, \tau\}$ ,  $\mu \in \mathbb{R}$ ,  $\rho \in [-1, 1]$ ,  $\sigma_v$  and  $\tau \in \mathbb{R}_+$ , denote the mean value, the persistence in volatility, the standard deviation of the state process and the instantaneous volatility, respectively.

The observations  $y_t = \log(p_t/p_{t-1})$ , also called log-returns, denote the logarithm of the daily difference in the exchange rate  $p_t$ , here,  $\{p_t\}_{t=1}^T$  is the daily closing price of the NASDAQ OMXS30 index (a weighted average of the 30 most traded stocks at the Stockholm stock exchange). We extract the data from [Quandl](#) for the period between January 2, 2015 and January 2, 2016. The resulting log-return

is shown in Figure 4. We use SMC to track the time-series persistency volatility, large variations are frequent, which is well-known as volatility clustering in finance, from the equation (42), as  $|\phi|$  is close to 1 and the standard variance is small, the volatility clustering effect easier occurs. We keep the same parameters as (Dahlin and Schön, 2019), where  $\mu \sim N(0, 1)$ ,  $\phi \sim TN_{[-1,1]}(0.95, 0.05^2)$ ,  $\delta_v \sim Gamma(2, 10)$ ,  $\tau = 1$ .

We use 25 particles to track the persistency volatility, the expectation of weights of particles is  $\frac{1}{25}$ , shown in Figure 4, it is stable as the same with Figure 1, the variance is in  $10^{-3}$  orders of magnitude under random sampling mechanism.

In Figure 5, the variance from our proposal shows the minimum value at different times, nearly all the plots share the common multimodal feature at the same time, it stems from the multinomial distribution that both of them have when they resample a new unit.

## 5 CONCLUSIONS

Resampling strategies are effective in Sequential Monte Carlo as the weighted particles tend to degenerate. However, we find that the resampling also leads to a loss of diversity among the particles. This arises because in the resampling stage, the samples are drawn from a discrete multinomial distribution, not a continuous one. Therefore, the new samples fail to be drawn as a type that has never occurred but stems from the existing samples by the repetitive schema. We have presented a repetitive deterministic domain with median traversal for resampling and have achieved the lowest variances compared to other resampling methods. As the size of the deterministic domain  $M \ll N$  (the size of population), our algorithm is faster than the state of the art, given a feasible size of particles, which is verified by theoretical deduction and experiments of the hidden Markov model in both the linear and the non-linear case.

There is a special case in which the median of particles belong to a deterministic domain, the particles with weight  $\tilde{w}_t^i \leq \frac{1}{N}$  have been discarded, thus, the

particle set after resampling is biased. Since the resampling scheme is repeated in the scaling domain, the skewed particles are discarded completely, although the total variance obtained by our method is minimized compared to other resampling methods. Since these sample points are biased when used for target tracking, it will affect the accuracy of the target to some extent. Alternatively, if we give a certain permissible error threshold for target localization and tracking, we can also try the feasibility of our method in this threshold, because it will be solved in a shorter time with the minimum total variance compared to other methods.

The broader impact of this work is that it can speed up existing sequential Monte Carlo applications and allow more precise estimates of their objectives. There are no negative societal impacts, other than those arising from the sequential Monte Carlo applications themselves.

## REFERENCES

- Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.
- Billingsley, P. (1995). Measure and probability.
- Casarin, R., Trecroci, C., et al. (2006). *Business Cycle and Stock Market Volatility: A Particle Filter Approach*. Università degli studi, Dipartimento di scienze economiche.
- Chopin, N. et al. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Annals of Statistics*, 32(6):2385–2411.
- Dahlin, J. and Schön, T. B. (2019). Getting started with particle Metropolis-Hastings for inference in nonlinear dynamical models. *Journal of Statistical Software*, 88(2):1–41.
- Dai, X. and Baumgartner, G. (2023a). Chebyshev particles. *arXiv preprint arXiv:2309.06373*.
- Dai, X. and Baumgartner, G. (2023b). Operator-free equilibrium on the sphere. *arXiv preprint arXiv:2310.00012*.
- Dai, X. and Baumgartner, G. (2023c). Optimal camera configuration for large-scale motion capture systems.
- Dai, X. and Baumgartner, G. (2023d). Weighted Riesz particles. *arXiv preprint arXiv:2312.00621*.
- Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436.
- Del Moral, P., Doucet, A., and Jasra, A. (2012). On adaptive resampling strategies for sequential monte carlo methods.
- Douc, R. and Moulines, E. (2007). Limit theorems for weighted samples with applications to sequential Monte Carlo methods. In *ESAIM: Proceedings*, volume 19, pages 101–107. EDP Sciences.
- Doucet, A., De Freitas, N., Gordon, N. J., et al. (2001). *Sequential Monte Carlo methods in practice*, volume 1. Springer.
- Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.
- Doucet, A. and Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12(656-704):3.
- Flury, T. and Shephard, N. (2011). Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models. *Econometric Theory*, pages 933–956.
- Fox, D. (2001). Kld-sampling: Adaptive particle filters and mobile robot localization. *Advances in Neural Information Processing Systems (NIPS)*, 14(1):26–32.
- Gilks, W. R. and Berzuini, C. (2001). Following a moving target—Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146.
- Gordon, N., Ristic, B., and Arulampalam, S. (2004). Beyond the Kalman filter: Particle filters for tracking applications. *Artech House, London*, 830(5):1–4.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET.
- Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.
- Künsch, H. R. et al. (2005). Recursive Monte Carlo filters: algorithms and theoretical analysis. *The Annals of Statistics*, 33(5):1983–2021.
- Liu, J. S. (2008). *Monte Carlo Strategies in Scientific Computing*. Springer Science & Business Media.
- Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044.
- Montemerlo, M., Thrun, S., and Whittaker, W. (2002). Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 695–701. IEEE.
- Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. Cambridge University Press.
- Särkkä, S., Vehtari, A., and Lampinen, J. (2007). Rao-Blackwellized particle filter for multiple target tracking. *Information Fusion*, 8(1):2–15.
- Smith, A. (2013). *Sequential Monte Carlo Methods in Practice*. Springer Science & Business Media.
- Thrun, S. (2002). Particle filters in robotics. In *UAI*, volume 2, pages 511–518. Citeseer.
- Welch, G., Bishop, G., et al. (1995). An introduction to the Kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.