# ONLINE ADAPTATION OF THE NUMBER OF PARTICLES OF SMC METHODS

*Víctor Elvira*[⋆]*, Joaquín Míguez*[†]*, Petar M. Djurić*[♯]

[⋆] Dept. of Signal Theory and Communic., Universidad Carlos III de Madrid, 28911 Leganés (Spain)
[†] School of Mathematical Sciences, Queen Mary University, London E1 4NS (UK)
[♯] Dept. of Electrical and Computer Eng., Stony Brook University, Stony Brook, NY 11794 (USA)
email: velvira@tsc.uc3m.es, j.miguez@qmul.ac.uk, petar.djuric@stonybrook.edu

## ABSTRACT

Particle filtering is a widely used sequential methodology that approximates probability distributions by using discrete random measures composed of weighted particles. A large number of particles improves the quality of the approximation but increases the computational requirements. Although there exists an abundant variety of particle filtering algorithms in the literature, there is lack of work devoted to selecting or adapting the number of particles systematically. In this paper we propose a novel methodology for online assessment of convergence of particle filtering. Based on theoretical analysis of the assessment, we propose an algorithm for the adaptation of the number of particles in online manner. The performance of the proposed algorithm is demonstrated for two state-space models.

***Index Terms*—** Particle filtering, sequential Monte Carlo, convergence assessment, predictive distribution, convergence analysis, computational complexity, adaptive complexity.

## 1. INTRODUCTION

In many problems of science and engineering, observations arrive sequentially and the interest is in inferring a hidden state that evolves over time and that is linked to observations through a state-space model. The Bayesian approach allows to include some prior information and to provide a probabilistic estimation of the sequence of hidden states. The inference can be performed in a closed form only in few scenarios (e.g., linear Gaussian state-space models using the well-known Kalman filter [1]), while in most of the cases, the solution must be approximated.

The publication of [2] introduced the particle filtering methodology (also known as sequential Monte Carlo), which has become one of the most popular tools for stochastic filtering. A particle filter approximates filtered distributions by sets of particles, where the number of used particles is critical. It is well known that when the number of particles grows to infinity, the approximation error vanishes (see for instance [3, 4]). However, when the number of particles grow, so do the computational costs. Therefore, this performance-computational cost tradeoff plays a key role in the selection of the number of particles, and it is of utmost importance for practitioners.

The number of particles is usually fixed throughout the simulation and is set in an ad-hoc manner. To our best knowledge, the problem of how to select the number of particles has been rarely addressed in the literature. Some algorithms have been proposed in [5, 6, 7], but all of them are heuristic and they do not provide any theoretical guarantees.

In this paper, we present a model-independent methodology for online assessment of the convergence of the particle filter and an algorithm that dynamically adapts the number of particles. The methodology is based on theory that guarantees the consistency of the proposed method [8]. The convergence assessment method measures the discrepancy between the observation at each time step and a set of "fictitious observations" generated from the predictive probability distribution of the observations. We propose an algorithm that adjusts the number of particles and that is light in computations. Numerical results show the validity of the method on two different state-space models.

The rest of the paper is organized as follows. We formulate the problem in Section 2. In Section 3 we introduce the method for online assessment of convergence and an algorithm for adapting the number of particles. We present simulation results in Section 4 and conclusions in Section 5.

## 2. PROBLEM FORMULATION

### 2.1. State-space models in discrete time

Mathematically, we describe the state-space model and the observation model by using the set of equations

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \boldsymbol{u}_t) \quad (1)$$

$$y_t = h(\mathbf{x}_t, \boldsymbol{v}_t) \quad (2)$$

where

$t \in \mathbb{N}$ is a discrete time index,
$\mathbf{x}_t \in \mathbb{R}^{n_x}$ is the hidden state of the system at time $t$,
$y_t \in \mathbb{R}$ is a scalar observation,
$\boldsymbol{u}_t \in \mathbb{R}^{n_u}$ is a noise vector with known distribution,
$\boldsymbol{v}_t \in \mathbb{R}^{n_v}$ is a noise vector with known distribution,
$g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is a known function, and
$h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_y}$ is a known function.

Although the method presented in this paper could be used in problems where some of the model parameters were unknown, here we focus on the case were all of them are known. The prior distribution of the state $p(\mathbf{x}_0)$ is also known.

V. Elvira and J. Míguez are also with the Gregorio Marañón Health Research Institute. V. Elvira was supported by the Community of Madrid Government under project AGES (S2010/BMD-2422), the Spanish Government (MINECO) under project OTOSiS (TEC2013-41718-R), and the European Union's 7th FP through the Marie Curie ITN MLPM2012 (Grant No. 316861). J. Míguez was supported by Ministerio de Economía y Competitividad of Spain (program Consolider-Ingenio 2010 CSD2008-00010 COMONSENS and project TEC2012-38883-C02-01 COMPREHENSION). P. M. Djurić was supported by NSF under award CCF-1320626.

***Algorithm 1*** *Bootstrap filter.*

1. Initialization. *At time $t = 0$, draw $M$ i.i.d. samples, $\mathbf{x}_0^{(i)}$, $i = 1, \ldots, M$, from the distribution $p(\mathbf{x}_0)$.*
2. Recursive step. *Let $\{\mathbf{x}_{t-1}^{(m)}\}_{m=1}^{M}$ be the particles (samples) generated at time $t-1$. At time $t$, proceed with the two steps below.*
   (a) *For $m = 1, ..., M$, draw a sample $\bar{\mathbf{x}}_t^{(m)}$ from the pdf $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ and compute the normalized weight as*
   $$w_t^{(m)} = \frac{p(y_t|\bar{\mathbf{x}}_t^{(m)})}{\sum_{k=1}^{M} p(y_t|\bar{\mathbf{x}}_t^{(k)})}. \quad (3)$$
   (b) *For $m = 1, ..., M$, let $x_t^{(m)} = \bar{x}_t^{(k)}$ with probability $w_t^{(k)}$, $k \in \{1, ..., M\}$.*

The problem consists in obtaining the densities of the hidden states by making use of all of the observations available at time $t$, i.e., approximating the distribution $p(\mathbf{x}_t|y_{1:t})$.

### 2.2. Bootstrap Particle Filter

Particle filters sequentially approximate the posterior distribution $p(\mathbf{x}_t|y_{1:t})$ by updating the previous approximation of $p(\mathbf{x}_{t-1} \mid y_{1:t-1})$ with the new observation $y_t$. The particle filter approximation of $p(\mathbf{x}_t|y_{1:t})$ is a random measure $\bar{\pi}_t^M = \{\mathbf{x}_t^{(m)}, w_t^{(m)}\}_{m=1}^{M}$, where $M$ is the number of particles, and $\mathbf{x}_t^{(m)}$ and $w_t^{(m)}$ are respectively the particles and the normalized weights. The bootstrap particle filter (BPF), also called sequential importance resampling (SIR) algorithm, was proposed in [2] and is the simplest particle filter.

The BPF algorithm is described in Algorithm 1. In Step 2.(a) the state predictive distribution is implicitly approximated as

$$p^M(\mathbf{x}_t \mid y_{1:t-1}) = \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}). \quad (4)$$

The Step 2.(b) represents the resampling step and, in particular, here we have used the so-called multinomial resampling algorithm [9, 10]. Nevertheless, many other resampling algorithms exist [11], and the convergence of the filter can be easily proved for various of them (see for instance [12]).

## 3. ONLINE SELECTION OF THE NUMBER OF PARTICLES

Our goal is to evaluate the convergence of the BPF in real time and, based on the assessment, adapt the number of used particles $M$. In practice, since only the observations are available, we will measure the accuracy of the approximation of the predictive observation distribution $p_t^M(y_t) \equiv p_t^M(y_t|y_{1:t-1})$.

We run the BPF in the usual way (as in Algorithm 1) with a light addition of computations described below. At each iteration, we generate $K$ "fictitious observations", denoted $\tilde{y}_t^{(1)}, \ldots, \tilde{y}_t^{(K)}$, from the approximate predictive observation pdf $p_t^M(y_t)$. If the BPF is operating with a small enough level of error, these fictitious observations come approximately from the same distribution as the acquired observation, i.e., $p_t^M(y_t) \approx p_t(y_t)$ [8, Theorem 1].

Below we describe the method, justify its theoretical validity, and discuss its computational complexity.

### 3.1. Generation of fictitious observations

The proposed method generates at each time $t$ $K$ fictitious observations (i.e., Monte Carlo samples), denoted $\{\tilde{y}_t^{(k)}\}_{k=1}^{K}$, from the approximate predictive observation pdf $p_t^M(y_t) = \frac{1}{M}\sum_{m=1}^{M} p(y_t|\bar{\mathbf{x}}_t^{(m)})$. Since the latter density is a finite mixture, drawing from $p_t^M(y_t)$ is in general straightforward. In order to generate $\tilde{y}_t^{(k)}$, it is enough to draw a sample $j^{(k)}$ from the discrete uniform distribution on $\{1, 2, ..., M\}$ and then generate $\tilde{y}_t^{(k)} \sim p(y_t|\bar{\mathbf{x}}_t^{(j^{(k)})})$.

### 3.2. Assessing convergence via invariant statistics

Let us first assume the perfect approximation case where $p_t^M(y_t) = p_t(y_t) = p(y_t|y_{1:t-1})$, i.e., there is no approximation error and, therefore, the fictitious observations $\{\tilde{y}_t^{(k)}\}_{k=1}^{K}$ and the observation $y_t$ come from the true distribution. We define the set $\mathcal{A}_{K,t} := \{y \in \{\tilde{y}_t^{(k)}\}_{k=1}^{K} : y < y_t\}$ and the associated r.v. $A_{K,t} := |\mathcal{A}_K| \in \{1, 2, ..., K\}$. Note that $\mathcal{A}_{K,t}$ is the set of fictitious observations which are smaller than the actual observation, while $A_{K,t}$ is the number of such observations (the cardinality of the set). If we let $\mathbb{Q}_K$ denote the probability mass function (pmf) of $A_K$, it can be shown that $\mathbb{Q}_K$ is uniform independently of the value and the distribution of $y_t$, which is a useful property that will be further exploited. This is rigorously proven by the Proposition below.

***Proposition 1*** *If $y_t, \tilde{y}_t^{(1)}, \ldots, \tilde{y}_t^{(K)}$ are i.i.d. samples from a common continuous (but otherwise arbitrary) probability distribution, then the pmf of the r.v. $A_{K,t}$ is*

$$\mathbb{Q}_K(n) = \frac{1}{K+1}, \qquad n = 0, ..., K. \quad (5)$$

***Proof***: Since $y_t, \tilde{y}_t^{(1)}, \cdots, \tilde{y}_t^{(K)}$ are i.i.d., all possible orderings of the $K+1$ samples are a priori equally probable, and the value of the r.v. $A_{K,t}$ depends uniquely on the relative position of $y_t$ after the samples are sorted (e.g., if $y_t$ is the smallest sample, then $A_{K,t} = 0$; if there is exactly one $\tilde{y}_t^{(i)} < y_t$ then $A_{K,t} = 1$, etc.). There are $(K+1)!$ different ways in which the samples $y_t, \tilde{y}_t^{(1)}, \cdots, \tilde{y}_t^{(K)}$ can be ordered, but $A_{K,t}$ can only take values from 0 to $K$. In particular, given the relative position of $y_t$, there are $K!$ different ways in which the remaining samples $\tilde{y}_t^{(1)}, \cdots, \tilde{y}_t^{(K)}$ can be arranged. Therefore, $\mathbb{Q}_K(A_K = n) = \frac{K!}{(K+1)!} = \frac{1}{K+1}$ for every $n \in \{0, 1, ..., K\}$. □

In practice, $p_t^M(y_t)$ is just an approximation of the predictive observation pdf $p_t(y_t)$, and therefore, the actual observation and the fictitious observations are not i.i.d. However, under some mild assumptions stated in [8, Section III], the a.s. convergence of $p_t^M(y_t)$ enables us to obtain an "approximate version" of the uniform distribution in Proposition 1, with the error vanishing as $M \to \infty$. To be specific, we introduce the set $\mathcal{A}_{K,M,t} := \{y \in \{\tilde{y}_t^{(k)}\}_{k=1}^{K} : y < y_t\}$, which depends on $M$ because of the mismatch between $p_t^M(y_t)$ and $p_t(y_t)$, and the associated r.v. $A_{K,M,t}$ with pmf $\mathbb{Q}_{K,M,t}$. The statistic $A_{K,M,t}$ is asymptotically distribution-invariant (independently of $t$ and the model) since $\mathbb{Q}_{K,M,t}(n) \to \frac{1}{K+1}$ when $M \to \infty$ (see [8, Theorem 2] for a proof).

### 3.3. BPF algorithm with adaptive number of particles

We propose a simple algorithm that dynamically adjusts the number of particles of the filter (in this case, the BPF) by exploiting the properties of the r.v. $A_{K,M,t}$. The proposed method is summarized

by Algorithm 2. The method is embedded into the basic BPF and can be seen as an extra Step 2.(c) in Algorithm 1.

Note that the algorithm adapts the number of particles, now $M_n$, where $n$ is a temporal index referring to the epochs in which the number of particles remains fixed. Each epoch lasts $W$ iterations and the number of particles is adapted at the end of it. The algorithm is initialized with $n = 1$ and a certain initial number of particles $M_1$. At each iteration, in Step 1(a), $K$ fictitious observations $\{\tilde{y}_t^{(k)}\}_{k=1}^K$ are drawn, and in Step 1(b), the statistic $a_{K,M,t} = A_{K,M,t}$ is obtained. At the end of an epoch, in Step 2(a), once a set of $W$ consecutive statistics have been acquired, $\mathcal{S}_t = \{a_{K,M,t-W+1}, a_{K,M,t-W+2}, ..., a_{K,M,t-1}, a_{K,M,t}\}$, a statistical test is performed for checking if $\mathcal{S}_t$ is a sequence of i.i.d. samples from the pmf given by Eq. (5).

There are several approaches that can be used to exploit the information contained in $\mathcal{S}_t$, and here we perform a Pearson's chi-squared test [13], where the $\chi_t^2$ statistic is computed according to Eq. (6). Then, a p-value $p_{K,t}^*$ for testing the hypothesis that the empirical distribution of $\mathcal{S}_t$ is uniform is computed. The value $p_{K,t}^*$ is obtained by comparing the $\chi_t^2$ statistic with the $\chi^2$ distribution with $K$ degrees of freedom. The p-value is then compared with two different significance levels: a low threshold $p_\ell$ and a high threshold $p_h$ If $p_{K,t}^* \leq p_\ell$, the number of particles is increased according to the rule $M_t = f_{\text{up}}(M_{t-1})$ whereas, if $p_{K,t}^* \geq p_h$, the number of particles is decreased according to the rule $M_t = f_{\text{down}}(M_{t-1})$. If $p_\ell < p_{K,t}^* < p_h$, we keep the number of particles unchanged.

The two significance levels $p_h$ and $p_\ell$ allow the practitioner to select the operation range in the performance-computational cost tradeoff. We can control the maximum and minimum computational complexity with $M_{\text{min}}$ and $M_{\text{man}}$, respectively, and the adaptation speed with $f_{\text{up}}(\cdot)$ and $f_{\text{down}}(\cdot)$, i.e., the rules for increasing and decreasing $M$, respectively.

***Algorithm 2*** *Adapting the number of particles*

1. *At time t,* ***[Statistic computation]***
   (a) *Draw* $\tilde{y}_t^{(k)} \sim p^M(y_t|y_{t-1}), k = 1, \ldots, K.$
   (b) *Compute* $a_{K,M,t} = A_{K,M,t}$*, i.e., the position of* $y_t$ *within the set of ordered fictitious observations* $\{\tilde{y}_t^{(k)}\}_{k=1}^K$.
2. *If* $t = nW$*,* ***[Convergence assessment and adaptation of*** $M$***]:***
   (a) *Compute the* $\chi_t^2$ *statistic over the empirical distribution of* $\mathcal{S}_t = \{a_{K,M,t}, a_{K,M,t-1}, ..., a_{K,M,t-W+1}\}$ *as*

$$\chi_t^2 = \sum_{j=0}^{K} \frac{(O_j - E_j)^2}{E_j}, \tag{6}$$

   *where* $O_j$ *is the frequency of the observations being in the* $j$*th relative position, i.e.,* $O_j = |a_{K,M,\tau} \in \mathcal{S}_t : a_{K,M,\tau} = j|$*, and* $E_j$ *is the expected frequency under the null hypothesis, i.e.,* $E_j = W \cdot \mathbb{Q}_K(j) = \frac{W}{K+1}$ *(see Eq. (5)).*
   (b) *Calculate the p-value* $p_{K,t}^*$ *by comparing the statistic* $\chi_t^2$ *to the* $\chi^2$*-distribution with* $K$ *degrees of freedom.*
   (c) *If* $p_{K,t}^* \leq p_\ell$*,*
   ***increase*** $M_n = \min\{f_{up}(M_{n-1}), M_{max}\}$.
   *Else, if* $p_{K,t}^* \geq p_h$*,*
   ***decrease*** $M_n = \max\{f_{down}(M_{n-1}), M_{min}\}$.
   *Else,*
   ***keep*** $M_n = M_{n-1}$.
   (d) *Set* $n = n + 1$.

### 3.4. Computational complexity

The cost of the added computations is in general negligible compared to the cost of running the BPF with $M$ particles. Note that extra cost of the algorithm comes from (1) drawing $K$ pseudo observations every time instant $t$, and (2) performing a chi-squared test every $W$ time instants. In general $K << M$, as we will show in Section 4.

## 4. SIMULATION RESULTS

In this section, we implement the proposed algorithm within the standard BPF, and we test it in two different scenarios. We have set the algorithm parameters as $W = 15$, $K = 5$, $M_{\text{max}} = 2^{12}$, $M_{\text{min}} = 2^4$, $f_{\text{up}}(M_{n-1}) = 2M_{n-1}$, and $f_{\text{down}}(M_{n-1}) = M_{n-1}/2$.

### 4.1. Stochastic volatility model

We tested the algorithm on a stochastic volatility model where the hidden state $x_t$ represents the log-volatility and is an AR(1) process [14, Chapter 14]. The model equations are

$$x_t = \alpha x_{t-1} + u_t, \tag{7}$$

$$y_t = \exp\left(\frac{x_t}{2}\right) v_t, \tag{8}$$

where $\alpha = 0.999$ is the AR parameter, and $u_t$ and $v_t$ denote independent zero-mean Gaussian random variables of variances $\sigma_u^2 = 1$ and $\sigma_v^2 = 0.5$, respectively. Note that $u_t$ is additive noise while $v_t$ is multiplicative.

| $[p_l - p_h]$ | $[0.2 - 0.6]$ | $[0.3 - 0.7]$ | $[0.4 - 0.8]$ |
|---|---|---|---|
| MSE | 2.18 | 1.44 | 1.30 |
| M | 23 | 882 | 1842 |
| p-val | 0.4712 | 0.4997 | 0.5071 |
| ex. time (s) | 0.87 | 6.93 | 31.20 |

**Table 1**. Stochastic volatility model (Subsection 4.1): $\alpha = 0.999$, $\sigma_u^2 = 1$, $\sigma_v^2 = 0.5$, $T = 3000$.

Table 1 displays the results of the MSE of the approximation of the posterior mean, the average number of particles, the p-values of the $\chi^2$ test, and the average execution time per run. The results have been averaged over 500 runs and the first half of time steps have been discarded (e.g., $\bar{M} = \frac{2}{T}\sum_{k=\frac{T}{2}+1}^{T} M_k$). We can see that the relation between the MSE, $\bar{M}$, and the pair of significance levels $[p_l - p_h]$ allows for selection of the operation range: a high operation range yields good performance (low MSE) at the cost of using a large number of particles (high $\bar{M}$). When we decrease the range, the algorithm decreases the number of particles, increasing consequently the approximation error.

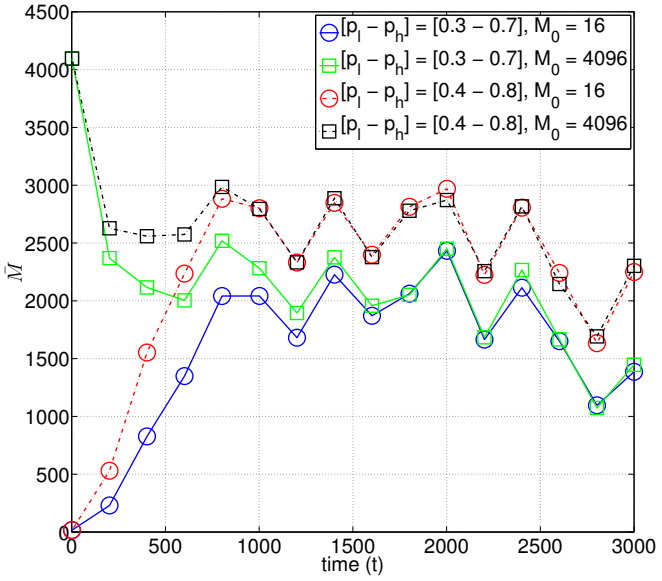


**Fig. 1**. Stochastic growth model (Subsection 4.2): Evolution in the number of particles averaged of 500 runs. Fixed underlying process for all runs.

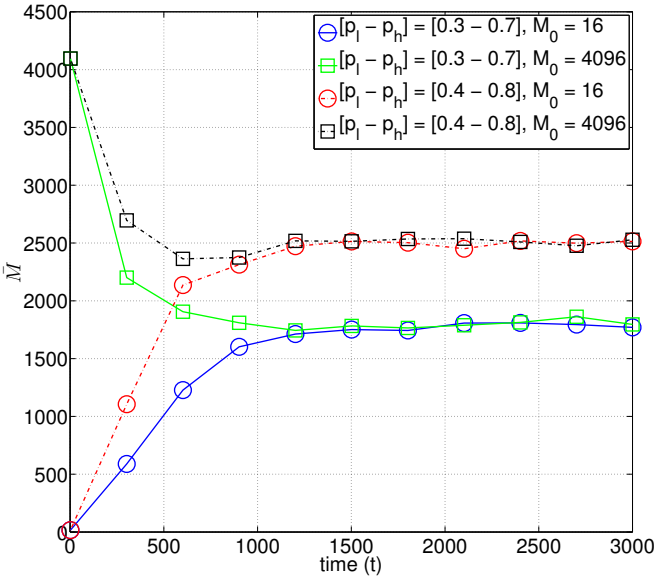


**Fig. 2**. Stochastic growth model (Subsection 4.2): Evolution in the number of particles averaged of 500 runs. Different random underlying process for every different run.

### 4.2. Stochastic growth model

The non-linear growth model (e.g., see [15]) is given by

$$x_t = \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(\phi t) + u_t, \quad (9)$$

$$y_t = \frac{x_t^2}{20} + v_t, \quad (10)$$

where $\phi = 0.4$ is a frequency parameter (in rad/s), and $u_t$ and $v_t$ denote independent zero-mean Gaussian random variables of variances $\sigma_u^2 = 2$ and $\sigma_v^2 = 0.1$, respectively.

Table 2 shows the MSE of the approximation of the posterior mean, the average number of particles, the p-values, and the execution time. For each run, the underlying process has been simulated, and again the first half of time steps have been discarded. The same conclusions can be extracted for this model: the pair $[p_l - p_h]$ allows for operating at different ranges of the performance/computational effort tradeoff.

Figure 1 shows the averaged evolution of the number of particles for two different operation ranges $[p_l - p_h] \in \{[0.3 - 0.7], [0.4 - 0.8]\}$, and two different initial number of particles $M_0 \in \{16, 4096\}$. We see that the initial number of particles does not have any effect after some iterations. In this simulation, the underlying process is simulated once and kept fixed for all runs. It can be seen that algorithm adapts the number of particles.

Figure 2 also shows the averaged evolution of the number of particles for the same operation ranges and initial number of particles. In this case, the underlying process is simulated for every different run. We see that after some iterations, regardless the initial number of particles, there is a stable number of particles that the algorithm automatically selects for every operation range.

| $[p_l - p_h]$ | $[0.2 - 0.6]$ | $[0.25 - 0.65]$ | $[0.3 - 0.7]$ | $[0.4 - 0.8]$ | $[0.5 - 0.9]$ |
|---|---|---|---|---|---|
| MSE | 17.78 | 11.88 | 5.13 | 3.92 | 3.46 |
| M | 158 | 522 | 1716 | 2452 | 3652 |
| p-val | 0.459 | 0.4765 | 0.4932 | 0.4956 | 0.4966 |
| ex. time (s) | 3.5 | 25.4 | 90.0 | 131.3 | 187.3 |

**Table 2**. Stochastic growth model (Subsection 4.2): $\phi = 0.4$, $\sigma_u^2 = 2$, $\sigma_v^2 = 0.1$, $T = 3000$.

## 5. CONCLUSIONS

The number of particles used in sequential Monte Carlo methods is usually set in an ad hoc way and kept fixed during the whole simulation. In this paper, we have proposed an online methodology for convergence assessment of the filter that allows for modification of the number of particles dynamically. We have presented a simple and light algorithm as an example, stating that the novel methodology opens the door for a range of algorithms for adapting the number of particles. We have shown the performance of the algorithm by numerical simulations in two different state-space models.

## 6. REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[2] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear and non-Gaussian Bayesian state estimation," *IEE Proceedings-F Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[3] P. Del Moral and L. Miclo, "Branching and interacting particle systems. Approximations of Feynman-Kac formulae with applications to non-linear filtering," *Lecture Notes in Mathematics*, pp. 1–145, 2000.

[4] A. Bain and D. Crisan, *Fundamentals of Stochastic Filtering*, Springer, 2008.

[5] D. Fox, "Adapting the sample size in particle filters through kld-sampling," *The International Journal of Robotics Research*, vol. 22, no. 12, pp. 985–1003, 2003.

[6] A. Soto, "Self adaptive particle filter," in *IJCAI*, 2005, pp. 1398–1406.

[7] O. Straka and M. Šimandl, "Particle filter adaptation based on efficient sample size," in *14th IFAC Symposium on System Identification*, 2006.

[8] V. Elvira, J. Miguez, and P. M. Djurić, "Adapting the number of particles in sequential monte carlo methods through an online scheme for convergence assessment," *arXiv:1509.04879 [stat.CO]*, 2015.

[9] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo Sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[10] R. Douc, O. Cappé, and E. Moulines, "Comparison of resampling schemes for particle filtering," in *Proc. $4^{th}$ Int. Symp. on Image and Signal Processing and Analysis*, September 2005, pp. 64–69.

[11] T. Li, M. Bolic, and P. M Djurić, "Resampling methods for particle filtering: Classification, implementation, and strategies," *Signal Processing Magazine, IEEE*, vol. 32, no. 3, pp. 70–86, 2015.

[12] D. Crisan, "Particle filters - a theoretical perspective," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., chapter 2, pp. 17–42. Springer, 2001.

[13] R. L. Plackett, "Karl pearson and the chi-squared test," *International Statistical Review/Revue Internationale de Statistique*, pp. 59–72, 1983.

[14] J. Durbin and S.J. Koopman, *Time series analysis by state space methods*, Number 38. Oxford University Press, 2012.

[15] P. M. Djurić and J. Míguez, "Assessment of nonlinear dynamic models by Kolmogorov–Smirnov statistics," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5069–5079, 2010.