COMPARATIVE ANALYSIS OF RESAMPLING PROCEDURES IN PARTICLE FILTER DESIGN

A. M. Isaev

Concern CSRI Elektropribor, St. Petersburg, Russia

✉ itmo_student@mail.ru

Abstract. Nonlinear filtering algorithms based on sequential Monte Carlo methods are considered. The scheme of an elementary particle filter is presented, and the main challenge of its implementation—the degeneracy of the algorithm—is explained. The essence of importance resampling procedures, intended to settle this challenge, is described. The resampling procedures widely used in navigation data processing and target tracking are reviewed. A criterion for determining the time point of resampling is given. The above resampling procedures are compared by analyzing the corresponding pseudocodes. As illustrative examples, two problems are considered: the estimation of a process representing the sum of a polynomial and the second integral of white noise, based on nonlinear measurements, and target tracking. For these problems, the resampling procedures are compared numerically by simulation using the method of statistical trials.

Keywords: Bayesian stochastic approach, nonlinear filtering, Monte Carlo methods, sequential importance resampling procedures, particle filter.

INTRODUCTION

The peculiarity of navigation data processing, guidance, and target tracking lies in the need to not only estimate the vector of unknown parameters but also to obtain quantitative characteristics of the estimation accuracy. Therefore, the algorithms for solving these problems are, in most cases, constructed within the so-called Bayesian stochastic approach [1–7] to calculate a mean-square optimal estimate and the corresponding covariance matrix of estimation errors. When the calculated covariance matrices match their actual values, the algorithm is said to be consistent [5].

Many problems in navigation and trajectory data processing are nonlinear; generally, it is impossible to construct an algorithm that produces an optimal estimate in real time due to its high computational complexity. Consequently, developers synthesize various suboptimal algorithms based on certain simplifications. When constructing suboptimal algorithms, much effort is applied to make them computationally simple, consistent, and sufficiently accurate (as close as possible to the potential accuracy, i.e., the accuracy yielded by the optimal algorithm).

Traditionally, the most challenging practical problems are those where the posterior probability density function (PDF) evolves into a complex, multi-peaked form [8]. Kalman-type algorithms [4, 9], which are typically used in navigation and trajectory data processing, often prove ineffective in the above problems. To solve them, the so-called sequential Monte Carlo methods (MCMs), also known as particle filters (PFs), and their various modifications have found wide application [10–16]. A distinctive feature of PFs is the ability to analyze the systematic error generated by the approximation during the estimation process and to achieve the desired accuracy by tuning the corresponding filter settings [17]. When the PF achieves an almost potential accuracy, the algorithm becomes consistent as well.

Despite all their advantages, PFs have several drawbacks. The main one, known as the degeneracy problem [18], is that measurement processing inevitably makes a weight close to 1 and the other weights close to 0; in such a case, the PF turns out inoperable. The degeneracy problem can be overcome by increasing the number of particles (particle count) used in the algorithm; however, the computational complexity of MCMs becomes significant accordingly [19]. This fact complicates the real-time implementation of PFs and, in some cases, makes the algorithm inapplicable even in the offline processing mode.

In most cases, this drawback of PFs can be eliminated through the so-called importance resampling procedures [20]. Here, the main idea is to redistribute particles from the regions of near-zero values of PDFs to those where the values are non-zero. According to an analysis of recent publications, PFs are typically designed using multinomial [21], stratified [22], systematic [23], residual [24], and residual systematic [19] resampling. However, neither the differences between these procedures, nor their advantages and disadvantages, are paid thorough attention in the literature.

Below, we describe the importance resampling procedures mentioned above, outline the general principles for their implementation in the design of PFs, and provide a comparative analysis of these procedures. The results of this study may be useful to developers dealing with measurement data processing, both to analyze the accuracy of suboptimal filtering algorithms and to construct algorithms for solving complex nonlinear problems where conventional Kalman-type algorithms prove ineffective.

The remainder of this paper is organized as follows. Section 1 presents the general nonlinear filtering problem under consideration, as well as the optimal algorithm for its solution. Section 2 is devoted to the construction of the elementary PF based on sequential Monte Carlo methods. In Section 3, we describe various importance resampling procedures traditionally used in PF design; in Section 4, we compare them numerically. The main findings of this paper are provided in the Conclusions.

1. PROBLEM STATEMENT AND ITS OPTIMAL SOLUTION

1.1. Formulation of the Problem

This paper addresses the nonlinear filtering problem of an *n*-dimensional Markov random sequence

$$x_k = f_k(x_{k-1}) + \Gamma_k w_k, \tag{1}$$

with the following notation: k is the discrete time instant; x_0 is the n_x-dimensional random Gaussian vector with a PDF $p(x_0) = N(x_0; \bar{x}_0, P_0)$ (hereinafter, $N(a; \bar{a}, A)$ designates a PDF of a Gaussian random vector a with a mean $\bar{a}$ and a covariance matrix A); w_k is the n_w-dimensional centered discrete Gaussian white noise with a known covariance matrix Q_k of dimensions $n_w \times n_w$, independent of x_0; $f_k(\bullet)$ is a known nonlinear n_x-dimensional vector function describing state dynamics; finally, Γ_k is a known matrix of dimensions $n_x \times n_w$.

It is required to estimate the sequence (1) based on m_y-dimensional discrete measurements of the form

$$y_k = h_k(x_k) + v_k, \tag{2}$$

where $h_k(\bullet)$ is a known nonlinear m_y-dimensional vector function describing the measurement model; v_k is the m_y-dimensional centered discrete Gaussian white noise independent of x_0 and w_k, with a known covariance matrix R_k.

1.2. Optimal Solution

As is known, when solving the filtering problem within the Bayesian approach**,** the mean-square optimal estimate and the corresponding conditional covariance matrix $P_k^{\text{opt}}(Y_k)$ are determined as follows [2, 4]:

$$\hat{x}_k^{\text{opt}}(Y_k) = \int x_k\, p(x_k / Y_k)\, dx_k, \tag{3}$$

$$P_k^{\text{opt}}(Y_k) = E_{p(x_k/Y_k)}\left\{\left(x_k - \hat{x}_k^{\text{opt}}(Y_k)\right)\left(x_k - \hat{x}_k^{\text{opt}}(Y_k)\right)^{\text{T}}\right\}, \tag{4}$$

where $p(x_k / Y_k)$ is the posterior (conditional) PDF of the vector x_k given the set of measurements $Y_k = [y_1^{\text{T}}, y_2^{\text{T}}, \ldots, y_k^{\text{T}}]^{\text{T}}$, and E means the expectation operator, with a subscript indicating the PDF used. We emphasize that this estimate minimizes the conditional (4) and unconditional [4]

$$\bar{P}_k^{\text{opt}} = E_{p(x_k, Y_k)}\left\{\left(x_k - \hat{x}_k^{\text{opt}}(Y_k)\right)\left(x_k - \hat{x}_k^{\text{opt}}(Y_k)\right)^{\text{T}}\right\} \tag{5}$$

covariance matrices of estimation errors.

In the above relations, the conditional covariance matrix (4) characterizes the current estimation error corresponding to a particular realization of the measurements Y_k, while its unconditional counterpart (5) characterizes the error calculated by probabilistic averaging over the entire possible set of measurements.

Clearly, to find the estimate (3), one needs the posterior PDF $p(x_k / Y_k)$, which is given by

$$p(x_k / Y_k) = \frac{p(y_k / x_k)\, p(x_k / Y_{k-1})}{\int p(y_k / x_k)\, p(x_k / Y_{k-1})\, dx_k}, \tag{6}$$

where $p(y_k / x_k)$ is the likelihood function, and $p(x_k / Y_{k-1})$ is the forecast density.

2. PARTICLE FILTERS

2.1. The Simplest Sequential Monte Carlo Method for Estimation

When solving estimation problems, MCM is interpreted as a method for approximating the posterior PDF; at each step k, it can be represented as [10]

$$p(x_k / Y_k) \approx \sum_{n=1}^{N} \tilde{\omega}_k^{(n)} \delta\left(x_k - x_k^{(n)}\right), \sum_{n=1}^{N} \tilde{\omega}_k^{(n)} = 1, \quad (7)$$

where $\delta(\bullet)$ stands for the multidimensional delta function; $x_k^{(n)}$ is a set of $n = \overline{1, N}$ independent realizations of the random vector, in the elementary case $x_k^{(n)} \sim p(x_k / x_{k-1}^{(n)}, Y_{k-1})$; $\tilde{\omega}_k^{(n)}$ are the normalized weights defined as

$$\tilde{\omega}_k^{(n)} = \omega_k^{(n)} \Big/ \sum_{n=1}^{N} \omega_k^{(n)},$$

and $\omega_k^{(n)}$ are the unnormalized weights satisfying (due to the recursive relation for the posterior PDF (6) and the Markov nature of the sequence (1)) the recursive relation

$$\omega_k^{(n)} = p(y_k / x_k^{(n)}) \omega_{k-1}^{(n)},$$

where $p(y_k / x_k^{(n)})$ is calculated, within a normalizing constant c, by

$$p(y_k / x_k^{(n)}) = c \exp\left\{-0.5\left(\left(y_i - h_i(x_i^{(n)})\right)^{\mathrm{T}} R^{-1}\left(y_i - h_i(x_i^{(n)})\right)\right)\right\}.$$

By substituting the approximation (7) into the expressions (3) and (4), based on the properties of the delta function, we easily derive the following rough equalities for the optimal estimate and the corresponding covariance matrix:

$$\hat{x}_k^{\mathrm{opt}}(Y_k) \approx \hat{x}_k(Y_k) = \sum_{n=1}^{N} \tilde{\omega}_k^{(n)} x_k^{(n)}, \quad (8)$$

$$P_k^{\mathrm{opt}}(Y_k) \approx P_k(Y_k) = \sum_{n=1}^{N} \tilde{\omega}_k^{(n)} x_k^{(n)} (x_k^{(n)})^{\mathrm{T}} - \hat{x}_k(Y_k)\left(\hat{x}_k(Y_k)\right)^{\mathrm{T}}.$$

An algorithm involving these expressions with the realizations $x_k^{(n)} \sim p(x_k / x_{k-1}^{(n)}, Y_{k-1})$ is essentially the elementary particle filter. Its pseudocode can be found, e.g., in [25].

Note that the realizations $x_k^{(n)}$ in (7) can be formed using another function $\pi(x_k / Y_k)$, different from $p(x_k / Y_{k-1})$, which is called an importance function. As underlined in [26], the choice of an appropriate importance function is crucial for PF design, but this issue goes beyond the scope of our work. The peculiarities of applying various importance functions were discussed in several publications; for example, see [27].

2.2. The Degeneracy Problem

The main advantage of PFs based on MCMs is that the computation error of the estimate (3) and the covariance matrix (4) vanishes as $N \to \infty$. This convergence creates the conditions for achieving any accuracy of PFs, up to the potential one corresponding to the mean-square optimal algorithm. However, despite all their benefits, PFs also suffer from several drawbacks. The main one is the so-called degeneracy problem of the algorithm. We will explain its essence below.

Clearly, when forming a sample $x_k^{(n)}$ at each kth step of the algorithm, some of its realizations will inevitably fall into regions with almost zero values of the posterior PDF. The weights $\omega_k^{(n)}$ corresponding to such realizations will be close to 0 as well. According to [18], when increasing the number of measurements, the number of realizations $x_k^{(n)}$ whose weights are significantly different from 0 will inevitably decrease until only one remains, whose normalized weight will be close to 1. Obviously, the resulting estimate (8) will have nothing in common with the optimal one.

The degeneracy problem can be overcome by increasing the number of realizations modeled in the PF. However, the computational complexity of the algorithm will grow significantly, complicating its real-time implementation and, in some cases, making the algorithm inapplicable even in the offline processing mode. (In the latter, there are no strict limitations on the volume of computations.)

Another recipe here is to use importance resampling procedures for PF design; they will be discussed below.

3. IMPORTANCE RESAMPLING PROCEDURES

3.1. The Essence of Importance Resampling

The main idea of importance resampling is to redistribute the particles $x_k^{(n)}$, $n = \overline{1, N}$, i.e., "place" them in regions where the posterior PDF is essentially non-zero, as determined by the corresponding weights $\tilde{\omega}_k^{(n)}$. This is achieved by simulating a new sample

(resampling) of particles $\tilde{x}_k^{(m)}$, $m=\overline{1, M}$, in accordance with a discrete distribution law specified as a set of N particles $x_k^{(n)}, n=\overline{1, N}$, and the corresponding probabilities (weights) $\Pr\left(\tilde{x}_k^{(m)}=x_k^{(n)}\right)=\tilde{\omega}_k^{(n)}$. We emphasize that the resampled set $\tilde{x}_k^{(m)}$, $m=\overline{1, M}$, is formed from the initial set $x_k^{(n)}$, $n=\overline{1, N}$, which retains the recursive nature of the algorithm.

Note that the sizes N and M of the original and resulting samples, respectively, generally differ; however, in most practical cases, $M=N$. Since the new sample $\tilde{x}_k^{(m)}$, $m=\overline{1, M}$, corresponds by form to the approximation (7), a new equal-weight approximation for it can be written as

$$p\left(x_k / Y_k\right) \approx \sum_{m=1}^{M} \tilde{q}_k^{(m)} \delta\left(x_k-\tilde{x}_k^{(m)}\right), \ \tilde{q}_k^{(m)}=1/M. \quad (9)$$

We underline that at the initial time (k = 0), the weights corresponding to the realizations $x_0^{(n)}$ will also be identical: in fact, the PDF $p(x_0)$ is represented by the approximation (9). Thus, the importance resampling procedure can be interpreted as a kind of restart of the PF.

To date, various methods have been proposed to organize resampling procedures [20]. In PF design, the most widely used methods are multinomial [21], stratified [22], residual [24], systematic [23], and residual systematic [19] resampling. According to a detailed analysis of the key relevant works, these schemes differ from one another only in the method for obtaining the realizations $\tilde{x}_k^{(m)}$. Below, we will describe the resampling procedures that have become widespread in navigation and trajectory data processing.

3.2. Multinomial, Stratified, and Systematic Resampling Procedures

Multinomial resampling is one of the first methods proposed. Sometimes this procedure is also called *simple random resampling*. Its main idea is to simulate a set of M realizations $u_k^{(m)}$, $m=\overline{1, M}$, of a random variable u_k with the uniform distribution on the interval $U=(0,1]$. Next, for each realization $u_k^{(m)}$, the realizations $x_k^{(n)}$, $n=\overline{1, N}$, are iterated through until finding one with a normalized weight $\tilde{\omega}_k^{(n)}$ that satisfies

$$S_k^{(n-1)}<u_k^{(m)} \le S_k^{(n)}, \text{ where } S_k^{(n)}=\sum_{n=1}^{n} \tilde{\omega}_k^{(n)}. \quad (10)$$

Stratified resampling and *systematic resampling* are two other common modifications of the importance resampling procedure. As in the multinomial resampling described above, the expression (10) is used here to simulate the realizations $\tilde{x}_k^{(m)}$. The main difference is the method for generating the realizations $u_k^{(m)}$.

In stratified resampling, the interval $U=(0,1]$ is divided into M equal half-intervals (strata) $U^{(m)}$, $m=\overline{1, M}$:

$$U=U^{(1)} \cup U^{(2)} \cup \ldots \cup U^{(M)}$$

$$=(0,1/M] \cup (1/M, 2/M] \cup \ldots \cup (1-1/M, 1].$$

Each realization $u_k^{(m)}$, $m=\overline{1, M}$, is generated independently as a variable uniformly distributed on the corresponding stratum.

In systematic resampling, $u_k^{(1)}$ is computed by analogy with stratified resampling, i.e., as a random variable uniformly distributed on the interval $(0,1/M]$, and all other realizations $u_k^{(m)}, m=\overline{2, M}$, are computed deterministically as

$$u_k^{(m)}=u_k^{(1)}+\frac{m-1}{M}, \ m=\overline{2, M}.$$

Note that in systematic resampling, the random number generator is called significantly fewer times than in stratified and multinomial resampling; therefore, it has smaller computational intensiveness.

3.3. Residual and Residual Systematic Resampling Procedures

Residual resampling [24] is another conventional modification of resampling. It consists of two stages as follows.

At the first stage of this procedure, each realization $x_k^{(n)}$, $n=\overline{1, N}$, is replicated (selected as a realization $\tilde{x}_k^{(m)}$) $M_{1,k}^{(n)}$ times, where

$$M_{1,k}^{(n)}=\left\lfloor M \tilde{\omega}_k^{(n)} \right\rfloor, \quad (11)$$

and $\lfloor \bullet \rfloor$ denotes the floor operator.

Direct analysis of the expression (11) shows that the particles with weights $\tilde{\omega}_k^{(n)}<1/M$ will not be replicated since $M_{1,k}^{(n)}=0$. In total, the first stage yields $M_{1,k}=\sum_{n=1}^{N} M_{1,k}^{(n)}$ particles.

The second stage of the residual resampling procedure is based on the so-called residual weights $\left\{\tilde{\omega}_k^{(n)}\right\}$ such that

$$\left\{M\tilde{\omega}_k^{(n)}\right\} = M\tilde{\omega}_k^{(n)} - \left\lfloor M\tilde{\omega}_k^{(n)} \right\rfloor. \quad (12)$$

By replacing $\left\lfloor M\tilde{\omega}_k^{(n)} \right\rfloor$ with $M_{1,k}^{(n)}$ and dividing both sides of (12) by M, we obtain

$$\left\{\tilde{\omega}_k^{(n)}\right\} = \tilde{\omega}_k^{(n)} - \frac{M_{1,k}^{(n)}}{M}. \quad (13)$$

At the second stage of the residual resampling procedure, the other $M_{2,k} = M - M_{1,k}$ realizations $\tilde{x}_k^{(m)}$ are generated using any of the above random resampling methods (multinomial or stratified); for this purpose, the residual weights (13) are pre-normalized:

$$\left\{\tilde{\omega}_k^{(n)}\right\}^* = \left\{\tilde{\omega}_k^{(n)}\right\}\frac{M}{M - M_{1,k}}.$$

It is intuitively clear that the two-stage nature of residual resampling entails additional computational costs. To overcome this drawback, residual systematic resampling was proposed.

Residual systematic resampling is a modification of residual resampling where the random resampling of the second stage is replaced by systematic resampling. In this case, the two stages of resampling can be combined so that each realization $x_k^{(n)}$, $n = \overline{1, N}$, is replicated $M_k^{(n)} = M_{1,k}^{(n)}$ times according to the expression

$$M_k^{(n)} = \left\lfloor M\left(\tilde{\omega}_k^{(n)} - \Delta u^{(n)}\right)\right\rfloor + 1,$$

$$\Delta u^{(n)} = \Delta u^{(n-1)} + \frac{M_{1,k}^{(n-1)}}{M} - \tilde{\omega}_k^{(n-1)},$$

where $\Delta u_k^{(1)}$ is computed similarly to $u_k^{(1)}$ in systematic resampling (i.e., as a random variable uniformly distributed on the half-interval $(0, 1/M]$).

As noted in [19], the resulting realizations $\tilde{x}_k^{(m)}$ generated by systematic and residual systematic resampling will coincide exactly, and the differences between these procedures may manifest only in their computational load. Indeed, in residual systematic resampling, inequality (10) is not verified, and the number of calculated values of $\Delta u^{(n)}$ is significantly smaller than that of $u^{(n)}$; therefore, the computational complexity of such a procedure is expected to be lower.

3.4. Criterion for Resampling

Resampling at each *k*th step, first, increases the initially significant computational complexity of PFs; second, it may yield a sample where the same element with weight 1/*N* is repeated *N* times. This problem is known as *sample impoverishment* [20]; in practice, special criteria are calculated to determine the timing of resampling. The most widely used criterion is the so-called *effective sample size*

$$N_k^E \approx \left(\sum_{n=1}^{N}(\tilde{\omega}_k^{(n)})^2\right)^{-1}.$$

The resampling procedure is executed when the criterion N_k^E is below a given threshold N_k^{Th}. According to the paper [20], this criterion takes values $1 \le N_k^E \le N$. If $N_k^{Th} = N$, resampling will be performed at each step. On the other hand, in the case $N_k^{Th} < 1$, the resampling procedure will never be executed. The criterion N_k^E characterizes the number of sample elements with significant weights (they cannot be ignored when constructing the estimates).

4. COMPARATIVE ANALYSIS OF RESAMPLING PROCEDURES

4.1. Analytical Comparison

Next, let us compare the importance resampling procedures described above. We begin with an analytical comparison by examining their pseudocodes, which can be found, e.g., in [20]. The main differences identified between the procedures are presented in the table below.

According to the table, the smallest number of calls to the random number generator is observed for the systematic and residual systematic procedures. In addition, such schemes are the simplest in terms of the number of computations.

Differences in importance resampling procedures

Comparison criteria	Importance resampling procedure				
	Multinomial	Stratified	Systematic	Residual	Residual systematic
The number of stages	1	1	1	2	1
The number of calls to the random number generator	M	M	1	$M_{2,k}$	1
Asymptotic computational complexity	$O(MN)$	$O(M)$	$O(M)$	$O(N)+O(M_{2,k})$	$O(N)$
The minimum number of replications of any particle	0	If $\tilde{\omega}_k^{(n)} > 1/M$, then $\lfloor M\tilde{\omega}_k^{(n)} \rfloor - 1$; otherwise, 0.	$\lfloor M\tilde{\omega}_k^{(n)} \rfloor$	$\lfloor M\tilde{\omega}_k^{(n)} \rfloor$	$\lfloor M\tilde{\omega}_k^{(n)} \rfloor + 1$
The maximum number of replications of any particle	N	$\lfloor M\tilde{\omega}_k^{(n)} \rfloor + 2$	$\lfloor M\tilde{\omega}_k^{(n)} \rfloor + 1$	$\lfloor M\tilde{\omega}_k^{(n)} \rfloor + M_{2,k}$	$\lfloor M\tilde{\omega}_k^{(n)} \rfloor + 1$

4.2. A Methodological Example

In this subsection, we compare the PFs constructed using various resampling procedures for the estimation problem of a process representing the sum of a first-degree polynomial and the second integral of centered white noise, based on nonlinear measurements.

Let the sequence (1) under estimation be described by

$$x_k = \Phi x_{k-1} + \Gamma w_k, \qquad (14)$$

with the matrices Φ, Γ, and the covariance matrix Q of the centered Gaussian white noise w_k having the form

$$\Phi = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad \Gamma = q\sqrt{\Delta t}\begin{bmatrix} \frac{\Delta t}{\sqrt{3}} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where q^2 denotes the power spectral density of the initial white noise, and Δt is the sampling interval. In particular, such a model is often used to describe measurement errors, e.g., errors in a navigation dead reckoning system for each coordinate [28], and approximate gravity anomaly (GA) deviations [29].

Let the nonlinear discrete measurements have the form

$$y_k = h(x_k) + v_k = a_1 + a_2 x_{1,k} + a_3 x_{1,k}^2 + a_4 x_{1,k}^3 + v_k,$$

where a_1, a_2, a_3, and a_4 are known numbers.

We perform simulations under the parameters $\bar{x}_0 = [-5 \quad 0.5]^{\mathrm{T}}$, $P_0 = \mathrm{diag}[\sigma_{0,1}^2 \; \sigma_{0,2}^2]$, $\sigma_{0,1} = 2.5$, $\sigma_{0,2} = 0.05$, $q = 0.05$, $\Delta t = 1$, $k = \overline{1,30}$, $r = 0.1$, $a_1 = 0.0875$, $a_2 = -0.1825$, $a_3 = 0.01$, and $a_4 = 0.01$.

We implement the following algorithms:

- the PF without resampling for $N = 10\,000$ ($\mu = 1$, hereinafter referred to as Algorithm 1);
- the PF with various resampling procedures for $N = 1000$ and $N_k^{Th} = N^{Th} = 0.2 \cdot N$:
  - the PF ($\mu = 2$, Algorithm 2) with multinomial resampling;
  - the PF ($\mu = 3$, Algorithm 3) with stratified resampling;
  - the PF ($\mu = 4$, Algorithm 4) with systematic resampling;
  - the PF ($\mu = 5$, Algorithm 5) with residual resampling;
  - the PF ($\mu = 6$, Algorithm 6) with residual systematic resampling.

The algorithms are compared using the method of statistical trials [30]. For each μth algorithm, we calculate the actual G_k^{μ} and estimated $\tilde{G}_k^{\mu}$ covariance matrices:

$$G_k^{\mu} \approx \frac{1}{L}\sum_{j=1}^{L}\left(x_k^{(j)} - \hat{x}_k^{\mu(j)}\left(Y_k^{(j)}\right)\right)\left(x_k^{(j)} - \hat{x}_k^{\mu(j)}\left(Y_k^{(j)}\right)\right)^{\mathrm{T}}, \qquad (15)$$

$$\tilde{G}_k^{\mu} \approx \frac{1}{L}\sum_{j=1}^{L} P_k^{\mu}(Y_k^{(j)}), \qquad (16)$$

where $x_k^{(j)}$, $Y_k^{(j)}$, $j = \overline{1, L}$, are the realizations of the random sequences obtained by simulation using the expressions (1) and (2), respectively; L is the total number of realizations; $\hat{x}_k^{\mu}(Y_k^{(j)})$ and $P_k^{\mu}(Y_k^{(j)})$ are the estimates and the corresponding covariance matrices for the μth algorithm.

To assess the computational costs of the algorithms, we calculate the coefficient

$$\Delta T^{\mu} = \frac{\tau^{\mu} - \tau^{*}}{\tau^{*}}, \ \tau^{\mu} = \frac{1}{L}\sum_{j=1}^{L} t_j^{\mu}, \quad \tau^{*} = \frac{1}{L}\sum_{j=1}^{L} t_j^{*},$$

where t_j^{μ} is the computer time for solving the estimation problem by the μth algorithm, and t_j^{*} is the same time characteristic for a reference algorithm.

Figure 1 shows the actual $\sqrt{\left[G_k^{\mu}\right]_{1,1}}$ (15) (solid lines) and calculated $\sqrt{\left[\tilde{G}_k^{\mu}\right]_{1,1}}$ (16) (dashed lines) root-mean-square errors (RMSE) of estimating the first component of the vector x_k for each algorithm with L = 1000. Hereinafter, $[A]_{i,i}$ means the ith diagonal element of a matrix A. Red corresponds to Algorithm 1, black to Algorithm 2, blue to Algorithm 3, magenta to Algorithm 4, green to Algorithm 5, and orange to Algorithm 6.

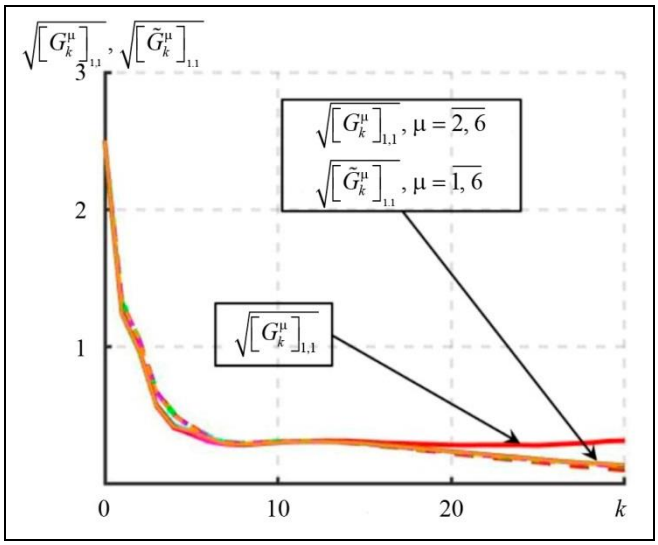

Fig. 1. The actual and calculated RMSEs.

According to the simulation results, the PFs constructed using various resampling procedures with N^* = 1000 ensure the required accuracy and are consistent. When decreasing the particle count, all algorithms lose consistency, and their accuracy drops. And the PFs based on resampling procedures with $N > N^*$ particles demonstrate the same performance, confirming the achievement of their potential accuracy.

Even for N = 10 000, the PF constructed without resampling is less accurate than the PFs with resampling and has no consistency.

Based on an analysis of computational costs, Algorithms 4 and 6 are the simplest among the PFs constructed using importance resampling procedures; their computational load is comparable. The computational loads of Algorithms 2, 3, and 5 are also close to each other, exceeding by nearly 5% those of the PFs based on systematic resampling.

Note that the difference in computational load may change when increasing the problem dimension and the simulation time. In this regard, we will perform a similar comparison for another problem, i.e., target tracking in the simplest setting; for example, see [5, 31].

4.3. Application of the PF to Target Tracking

Consider a right-handed 3D local Cartesian frame (Fig. 2) with the origin O, the perpendicular OX and OY axes in the horizontal plane, and the orthogonal OZ axis directed upward. By assumption, a radar is located at the origin of this frame; at discrete time instants k, it measures the range d_k to moving objects (targets), their elevation angle ψ_k, and azimuth θ_k.

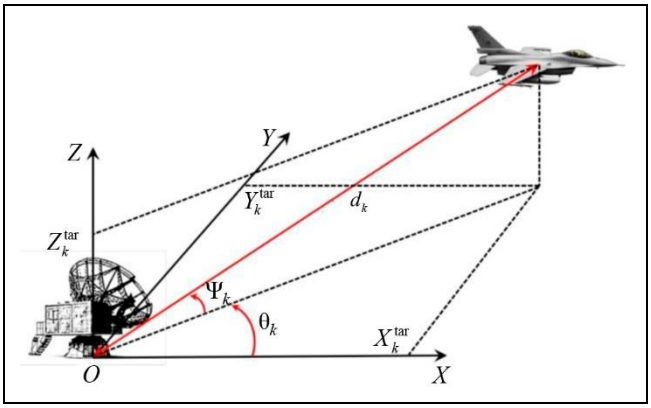

Fig. 2. The radar and target in the frame $OXYZ$.

At the initial time instant ($k = 0$), the radar detects a certain target with unknown coordinates $X_0^{\text{tar}}, Y_0^{\text{tar}}, Z_0^{\text{tar}}$; the target's dynamics can be described [31] by equation (14), where

$$x_k = \left[X_k^{\text{tar}} \ Y_k^{\text{tar}} \ Z_k^{\text{tar}} \ V_k^X \ V_k^Y \ V_k^Z \right]^{\text{T}},$$

$$\Phi = \begin{bmatrix} I_3 & \Delta t I_3 \\ 0_3 & I_3 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \frac{\Delta t^2}{2} I_3 \\ \Delta t I_3 \end{bmatrix}, \qquad (17)$$

$$Q = I_3 (\sigma^a)^2,$$

V_k^X, V_k^Y, and V_k^Z are the projections of the unknown velocity vector of the target onto the OX, OY, and OZ axes, respectively; I_3 and 0_3 denote an identity matrix and a zero matrix of order 3, respectively; and σ^a is a known number.

Given the radar's location at the origin, the measurements y_k are written as

$$y_k = \begin{bmatrix} d_k \\ \theta_k \\ \psi_k \end{bmatrix} = h_k(x_k) + v_k$$

$$= \begin{bmatrix} \sqrt{(X_k^{\text{tar}})^2 + (Y_k^{\text{tar}})^2 + (Z_k^{\text{tar}})^2} \\ \arctan(Y_k^{\text{tar}} / X_k^{\text{tar}}) \\ \arctan\left(\dfrac{Z_k^{\text{tar}}}{\sqrt{(X_k^{\text{tar}})^2 + (Y_k^{\text{tar}})^2}} \right) \end{bmatrix} + v_k . \quad (18)$$

It is required to estimate the target's coordinates X_0^{tar}, Y_0^{tar}, and Z_0^{tar} (the components of the vector x_k (14), (17)) from the measurements (18).

We perform simulations under the following parameters: $X_k^{\text{real}} = Y_k^{\text{real}} = 50$ km, and $Z_0^{\text{real}} = 5$ km as the target's true initial coordinates; $V_0^{X,\text{real}} = V_0^{Y,\text{real}} = 250\,\text{m/s}$ and $V_0^{Z,\text{real}} = 0\,\text{m/s}$ as the projections of the true velocity onto the *OX*, *OY*, and *OZ* axes at the initial time instant; the projections V_0^X, V_0^Y, and V_0^Z of the target's velocity vector are independent and uniformly distributed random variables with $p(V_0^{XO}) \sim U[150, 350]$ m/s, $p(V_0^{YO}) \sim U[150, 350]$ m/s, and $p(V_0^{ZO}) \sim U[0, 40]$ m/s, where $p(V) \sim U[a, b]$ denotes the uniform distribution on the interval $[a, b]$; $\sigma^a = 15$ m/s^2; $\Delta t = 0.1$ s; a simulation time of $T = 30$ s, i.e., $k = \overline{1, 300}$; finally, $R = \text{diag}[(\sigma^d)^2\ (\sigma^\theta)^2\ (\sigma^\psi)^2]$, where $\sigma^d = 25$ m and $\sigma^\theta = \sigma^\psi = 0.3°$.

When constructing the FP to generate the realizations $X_0^{\text{tar},(n)}, Y_0^{\text{tar},(n)}$, and $Z_0^{\text{tar},(n)}$, we employ the measurements y_0 of the range, elevation angle, and azimuth taken at the initial time instant (upon target detection). To this end, we first generate a set of realizations $\chi^{(n)} = \left[d_0^{(n)}\ \theta_0^{(n)}\ \psi_0^{(n)} \right]^{\text{T}}$, $\chi^{(n)} \sim y_0 + v^{\chi,(n)}$, where $v^{\chi,(n)} \sim N(v^\chi; 0, 2R)$ are the errors of the measurements (18) obtained by simulations with the doubled root-mean-square deviations to take the error $v \sim N(v; 0, R)$ of y_0 into account.

Next, we form a set of realizations corresponding to the target's position in the Cartesian frame:

$$X_0^{\text{tar},(n)} = d_0^{(n)} \cos(\psi_0^{(n)}) \cos(\theta_0^{(n)}),$$

$$Y_0^{\text{tar},(n)} = d_0^{(n)} \cos(\psi_0^{(n)}) \sin(\theta_0^{(n)}),$$

$$Z_0^{\text{tar},(n)} = d_0^{(n)} \sin(\psi_0^{(n)}).$$

Obviously, the realizations $X_0^{\text{tar},(n)}, Y_0^{\text{tar},(n)}$, and $Z_0^{\text{tar},(n)}$ will not completely match the actual PDF for the coordinates, but taken in the aggregate, they cover the region of coordinate uncertainty.

We perform simulations by the methodology described in [30], with averaging over L = 1000 realizations. Using the covariance matrices (15), (16), we calculate the actual $\sqrt{G_k^{R,\mu}}$ and the estimated $\sqrt{\tilde{G}_k^{R,\mu}}$ radial RMSEs of estimating the coordinates

$$\sqrt{G_k^{R,\mu}} = \sqrt{\left[G_k^\mu\right]_{1,1} + \left[G_k^\mu\right]_{2,2} + \left[G_k^\mu\right]_{3,3}}, \quad (19)$$

$$\sqrt{\tilde{G}_k^{R,\mu}} = \sqrt{\left[\tilde{G}_k^\mu\right]_{1.1} + \left[\tilde{G}_k^\mu\right]_{2.2} + \left[\tilde{G}_k^\mu\right]_{3.3}}. \quad (20)$$

We implement the same algorithms as in the methodological example, with the same particle count: $N = 10\,000$ and $N_k^{Th} = N^{Th} = 0.6N$. Figure 3 shows the radial actual (19) and estimated (20) RMSEs (the solid and dashed lines, respectively) for each algorithm under consideration. The color–algorithm correspondence here is the same as in subsection 4.2.

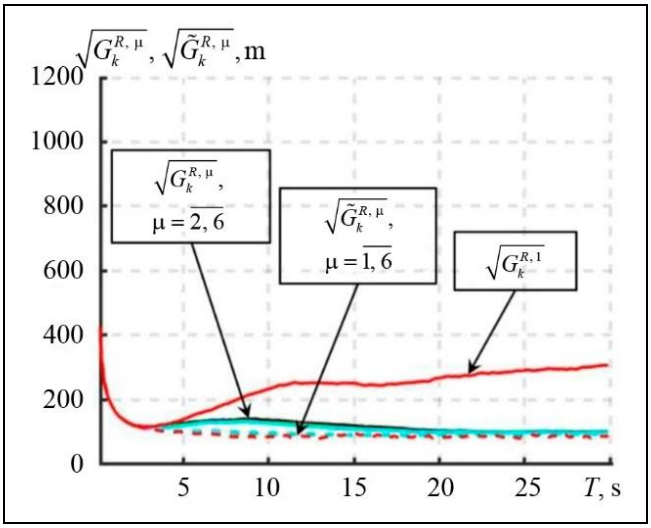

Fig. 3. The actual and estimated radial RMSEs.

The simulation results are similar to those obtained for the methodological example: the PFs based on various resampling procedures, with identical settings (particle count and N^{Th}), provide similar levels of accuracy and consistency. However, to achieve the potential accuracy and ensure consistency, a PF constructed without resampling requires much more particles than its counterpart with resampling.

An analysis of computational costs also confirms the conclusions of subsection 4.2. Among the FPs constructed using resampling procedures, Algorithms 4 and 6 are the simplest, with comparable computational loads. However, unlike the previous results of the methodological example, their advantage over Algorithms 2, 3, and 5, in terms of computational inten-

siveness, turns out to be insignificant in target tracking, amounting to nearly 0.5%. This situation can be explained as follows: when increasing the dimension of the estimated vector, the computational load of the PF (without resampling) grows significantly. The costs of the resampling procedures, although also increasing, become inconsiderable against this background.

CONCLUSIONS

In this paper, we have compared several importance resampling schemes used in particle filter design. According to the results, all the resampling procedures under consideration noticeably reduce the number of particles (particle count) compared to the conventional Monte Carlo method. No significant advantages, in terms of the achieved accuracy or consistency of PFs, can be obtained by choosing a particular resampling scheme for PF design. From a computational complexity standpoint, systematic and residual systematic resampling are somewhat simpler than the other schemes considered; however, these benefits are more pronounced when solving problems with the state vector of a small dimension.

The results of this study may be useful to developers of filtering algorithms for complex nonlinear measurement data processing tasks arising in modern control, navigation, and target tracking systems.

REFERENCES

1. Stratonovich, R.L., Conditional Markov Processes, *Theory of Probability and Its Applications*, 1960, vol. 5, no 2, pp. 156–178.
2. Jazwinski, A.H., *Stochastic Process and Filtering Theory*, New York: Academic Press, 1970.
3. *Applied Optimal Estimation*, Gelb, A., Ed., Cambridge: MIT Press, 1974.
4. Stepanov, O.A., *Primenenie teorii nelineinoi fil'tratsii v zadachakh obrabotki navigatsionnoi informatsii* (Application of Nonlinear Filtering Theory in Navigation Data Processing), St. Petersburg: Concern CSRI Elektropribor, 2003. (In Russian.)
5. Bar-Shalom, Y., Li, X., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation*, New York: Wiley-Interscience, 2001.
6. Simon, D., *Optimal State Estimation: Kalman H_∞ and Nonlinear Approaches*, NJ: John Wiley & Sons, 2006.
7. Särkkä, S., *Bayesian Filtering and Smoothing*, Cambridge: Cambridge University Press, 2013.
8. Isaev, A., Stepanov, O., and Litvinenko, Y., Comparative Analysis of Recursive and Nonrecursive Linearization–Based Estimation Algorithms, *Int. J. Dynam. Control*, 2025, vol. 13, art. no. 95.
9. Stepanov, O.A., Litvinenko, Yu.A., and Isaev, A.M., Comparative Analysis of Quasi-linear Kalman-type Algorithms in Estimating a Markov Sequence with Nonlinearities in the System and Measurement Equations, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2024, vol. 25, no. 11, pp. 585–595. (In Russian.)
10. Doucet, A., de Freitas, N., and Gordon, N., *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2001.
11. Konovalov, A.A., *Osnovy traektornoi obrabotki radiolokatsionnoi informatsii*. *Ch. 2* (Foundations of Trajectory Radar Data Processing. Part 2), St. Petersburg: Saint Petersburg Electrotechnical University (LETI), 2014. (In Russian.)
12. Matousek, J., Dunik, J., and Straka, O., Density Difference Grid Design in a Point–Mass Filter, *Energies*, 2020, vol. 13, no. 16, art. no. 4080.
13. Khmarskii, P.A., *Obrabotka rezul'tatov izmerenii koordinat i parametrov dvizheniya v sistemakh monitoringa vozdushnoi i nazemnoi obstanovki* (Processing the Measurements of Coordinates and Motion Parameters in Air and Ground Situation Monitoring Systems), Minsk: Belaruskaya Navuka, 2025. (In Russian.)
14. Chopin, N. and Papaspiliopoulos, O., *An Introduction to Sequential Monte Carlo*, Cham: Springer, 2020.
15. Wills, A.G. and Schön, T.B., Sequential Monte Carlo: A Unified Review, *Annual Review of Control, Robotics, and Autonomous Systems*, 2023, vol. 6, no. 1, pp. 159–182.
16. Cheng, Y., Ren, W., Xiu, C., and Li, Y., Improved Particle Filter Algorithm for Multi-Target Detection and Tracking, *Sensors*, 2024, vol. 24, no. 14, art. no. 4708.
17. Berkovskii, N.A. and Stepanov, O.A., Error of Calculating the Optimal Bayesian Estimate Using the Monte Carlo Method in Nonlinear Problems, *J. Comput. Syst. Sci. Int*., 2013, vol. 52, pp. 342–353.
18. Kong, A., Liu, J.S., and Wong, W.H., Sequential Imputations and Bayesian Missing Data Problems, *J. American Statistical Association*, 1994, vol. 89, no. 425, pp. 278–288.
19. Bolić, M., Djurić, S., and Hong, P.M., Resampling Algorithms for Particle Filters: A Computational Complexity Perspective, *EURASIP J. on Applied Signal Processing*, 2004, no. 15, pp. 2267–2277.
20. Li, T., Bolić, M., and Djurić, P.M., Resampling Methods for Particle Filtering: Classification, Implementation, and Strategies, *IEEE Signal Processing Magazine*, 2015, vol. 32, no. 3, pp. 70–86.
21. Gordon, N., Salmond, D., and Smith, A., Novel Approach to Nonlinear/Non–Gaussian Bayesian State Estimation, *IEE Proc., F Radar Signal Process*., 1993, vol. 140, no. 2, pp. 107–113.
22. Kitagawa, G., Monte Carlo Filter and Smoother and Non–Gaussian Nonlinear State Space Models, *J. Comput. Graph. Stat*., 1996, vol. 5, no. 1, pp. 1–25.
23. Carpenter, J., Clifford, P., and Fearnhead, P., An Improved Particle Filter for Nonlinear Problems, *IEE Proc., Radar Sonar Navigat.*, 1999, vol. 146, no. 1, pp. 2–7.
24. Liu, J.S. and Chen, R., Sequential Monte-Carlo Methods for Dynamic Systems, *J. Amer. Statist. Assoc*., 1998, vol. 93, no. 443, pp. 1032–1044.
25. Vasiliev, V.A., Isaev, A.M., Stepanov, O.A., and Toropov, A.B., Monte Carlo Methods in Measurement Data Processing Using Resampling Procedures, *Journal of Computer and Systems Sciences International*, 2025, vol. 64, no. 6, pp. 900–926.
26. Stepanov, O.A. and Toropov, A.B., Application of Sequential Monte Carlo Methods with Analytical Integration Procedures in Navigation Data Processing, *Trudy XII-go Vserossiiskogo soveshchaniya po problemam upravleniya (VSPU-2014)*

(Proceedings of XII All-Russian Meeting on Control Problems (AMCP-2014)), Moscow, 2014, pp. 3324–3337. (In Russian.)
27. Doucet, A., On Sequential Simulation–Based Methods for Bayesian Filtering, *Technical Report*, Cambridge: Cambridge University, 1998.
28. Stepanov, O.A., Vasil'ev, V.A., and Isaev, A.M., A Suboptimal Algorithm for Correcting a Navigation System with a Variable Nature of Its Errors Based on Geophysical Field Data, *Proceedings of the International Workshop on Navigation and Motion Control (NMC'2023)*, St. Petersburg, 2023, pp. 90–95. (In Russian.)
29. Stepanov, O.A., *Osnovy teorii otsenivaniya s prilozheniyami k zadacham obrabotki navigatsionnoi informatsii. Chast' 2 Vvedenie v teoriyu fil'tratsii* (Foundations of Estimation Theory with Applications to Navigation Data Processing Problems. Part 2: Introduction to Filtering Theory), St. Petersburg, Concern CSRI Elektropribor, 2025. (In Russian.)
30. Stepanov, O.A. and Isaev, A.M., A Procedure of Comparative Analysis of Recursive Nonlinear Filtering Algorithms in Navigation Data Processing Based on Predictive Simulation, *Gyroscopy Navig*., 2023, vol. 14, pp. 213–224.
31. Gustafsson, F., Gunnarsson, F., Bergman, N., et al., Particle Filters for Positioning, Navigation and Tracking, *IEEE Transactions on Signal Processing*, 2002, vol. 50, no. 2, pp. 425–437.

This paper was recommended for publication by B.V. Pavlov, a member of the Editorial Board.

Received October 13, 2025, and revised February 2, 2026. Accepted April 8, 2026

Author information

Isaev, Aleksei Mikhailovich. Junior Researcher, Concern CSRI Elektropribor, St. Petersburg, Russia
✉ itmo_student@mail.ru
ORCID iD: https://orcid.org/0000–0003–2111–0308

Cite this paper

Isaev, A.M., Comparative Analysis of Resampling Procedures in Particle Filter Design. *Control Sciences* **3,** 83–92 (2026).

Original Russian Text © Isaev, A.M., 2026, published in *Problemy Upravleniya*, 2026, no. 3, pp. 100–111.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com