

A Hierarchical Resampling Algorithm with Adaptive Interval for Particle Filter

Xiaohui-Zeng^{1,2}, Yibing-Shi² and Yi-Lian³

¹*School of Communication Engineering, Chengdu University of Information and Technology, Chengdu, China*

²*School of Automation Engineering, UESTC, Chengdu, China*

³*Motorola (China) Solutions, Chengdu, China*
huizi003@126.com

Abstract

In this article, we present an improved resampling algorithm for particle filtering, which is based on partial resampling and residual resampling. This algorithm provides an approach to selectively carry out hierarchical resampling operations on three sets of particles divided by large, medium and small weights, and especially to do skip resampling for partial small particles with an adaptive interval M . Simulation results verify that the proposed algorithm could reduce the depletion problem, maintain a good diversity of particles and improve the accuracy of PF performance.

Keywords: *particle filtering, hierarchical weights, partial resampling, skip resampling, adaptive interval*

1. Introduction

Particle filters (PF) have been applied with great success to solve state estimation problems and prediction problems in dynamic systems since the 1990s. Compared with traditional filtering ways such as Kalman filtering and extended Kalman filtering, particle filtering shows its superior performance in complex environments with noises. In the past decades, extensive research and work has been done on particle filtering, including improving the performance and overcoming the computational burden [1].

In fact, the particle filter is a sequential Monte Carlo method based on particles of various probability densities, which are represented by corresponding weights. Thus continuous distributions are approximated by discrete random measures, which are samples of the unknown states and space, and then they are computed by recursive use of the importance sampling principle. As a result of the sequential importance sampling (SIS) method and the particles are predicted according to the state equation during the prediction step of Bayes filtering. Throughout this process, the particles in a swarm are adapted to an evolving target distribution with their weights calculated by using the likelihood of new observation combined with the former weights. Sometimes, in order to avoid the degeneracy of particle into a unique particle of high weight, a resampling step becomes a very necessary addition to dismiss the particle samples with lower weights and produce multiple copies of those good samples to help generating better future samples in the SIS setting [2, 3]. However this resampling also causes the known problem of sample impoverishment, which could lead to non-robust estimation in the particle filter. Usually there are three ways to mitigate sample impoverishment. The first one is to add state noise covariance or sample noise [3], with roughening strategies used to rejuvenate the diversity of particles proposed by Gordon. The second one is to construct an

advanced kernel, via a MCMC step [4], or the regularized kernel [5]. The third one is to use certain new hybrid methods or intelligent choices to deal with sample impoverishment. Various artificial intelligent techniques such as particle swarm optimization, genetic algorithm, or ant colony optimization techniques form a coherent perspective to optimize the distribution of particles. Sophisticated techniques that help generate better resamples in the PF[6-9] have been developed to improve the sample diversity and the filtering performance, while maintaining computational efficiency.

In this paper, we focus our attention on improving the performance of PF while using adaptive resampling schemes on the particle sets, and especially how to efficiently make use of particles with small weights. Also we provide an adaptive interval M during the algorithm called “the skip resampling step”, in order to balance the need for diversity with the need for computational efficiency.

This paper is organized as follows. In section 2 we recall the particle filter principles including resampling process. We begin section 3 with a presentation of resampling algorithm development and several resampling methods for particle filtering. Then a new and improved partial resampling algorithm together with its definitions is proposed here. Section 4 is devoted to simulation experiments of two models. Carrying on with implementation of MPF algorithm, the validity and performance of the PF is confirmed. Finally conclusions are presented in sections 5.

2. Fundamentals of Particle Filtering

Generally, the framework of Bayesian tracking consists of estimating the dynamic state of objects in a nonlinear stochastic system based on a set of noisy observations. The time-varying object state is denoted by the state vector x_t , and the observation by the vector y_t , where t indicates discrete time. Based on hidden Markov models (HMM's), we suppose that the transition equation describes the prior distribution of a hidden Markov process $\{x_t, t \in N\}$, and the observation equation describes the likelihood of the observations $\{y_t, t \in N\}$. So the state transition model and observation model are written in the following form respectively:

$$x_t = f(x_{t-1}, u_t) \quad (1)$$

$$y_t = g(x_t, v_t) \quad (2)$$

Where $f(\cdot)$ and $g(\cdot)$ are known functions, while u_t and v_t are random noise vectors of given distributions. Within a Bayesian framework, the posterior distribution of x_{t-1} can be denoted as $p(x_{t-1}|y_{t-1})$, the predictive distribution for x_t is

$$p(x_t|y_{t-1}) = \int f(x_t|x_{t-1})p(x_{t-1}|y_{t-1})dx_{t-1} \quad (3)$$

Also, if we have an observation y_t , the new posterior distribution at time t becomes

$$p(x_t|y_t) \propto g(y_t|x_t)f(x_t|x_{t-1})p(x_{t-1}|y_{t-1}) \quad (4)$$

Since it is generally impossible to sample from the state posterior $p(x_t|y_t)$, an importance sampling is implemented under the recursive Bayesian framework; it is designated as the Sequential Importance Sampling (SIS), which is the basis of PF. Suppose $p(x) \propto \pi(x)$ is a probability density but difficult to draw samples. Using importance sampling, we can easily generate the new states x_t from an importance density $q_t(\cdot)$ by $x_t \sim q_t(x_t|x_{t-1})$ which depends on particles at time $t-1$ and observed

data y_t . Also the posterior density can be approximated from the weighted particles $\{x_t^i, \omega_t^i\}_{i=1}^N$ as follows:

$$p(x_t|y_t) \cong \sum_{i=1}^N \omega_t^i \delta_x(x_t - x_t^i) \quad (5)$$

Where $\sum_{i=1}^N \omega_t^i = 1$ and $\omega_t^i \propto \frac{\pi(x_t^i)}{q(x_t^i)}$ is the normalized weight of the i th particle.

Also the density can be expressed by

$$p(x_t|y_t) = p(x_{t-1}|y_{t-1}) \times \frac{g(y_t|x_t)f(x_t|x_{t-1})}{p(y_t|y_{t-1})} \quad (6)$$

By denoting importance function $\pi(x_t) = p(x_t|y_t)$, the important weight is given by

$$\omega_t^i \propto \omega_{t-1}^i \times \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, y_t)} \quad (7)$$

Put (7) into (5), we can obtain an approximation to the true posterior density $p(x_t|y_t)$. However, after a few iterations in the particle propagation process, the variance of the importance weights can only increase (stochastically) over time, the weights will concentrate on a few particles only and most particles will have negligible weight close to zero. Then the particles x_{t-1} should be resampled according to their weights w_{t-1} before making a draw for the new state $x_t \sim f(x_t|x_{t-1})$. This phenomenon, called sample degeneracy, is also one of the inherent faults of SIS suffers from.

Therefore the basic idea of resampling methods is to eliminate degeneracy trajectories, a suitable criterion of degeneracy of the algorithm is the effective sample size N_{eff} introduced by Liu in [11]. It is important to obtain a criterion based on which one would know when to perform a resampling step accordingly. Since N_{eff} cannot be calculated directly, an estimate \hat{N}_{eff} of N_{eff} is given by

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (\omega_t^i)^2} \quad (8)$$

\hat{N}_{eff} is the so-called Effective Sample Size (ESS), and the lower \hat{N}_{eff} will cause higher degeneracy. Once \hat{N}_{eff} drops below a given threshold \hat{N}_{th} , it means that there is a need to operate resampling.

We can obtain the general implementation of the sequential importance sampling and resampling (SISR) algorithm explained in Fig.1.

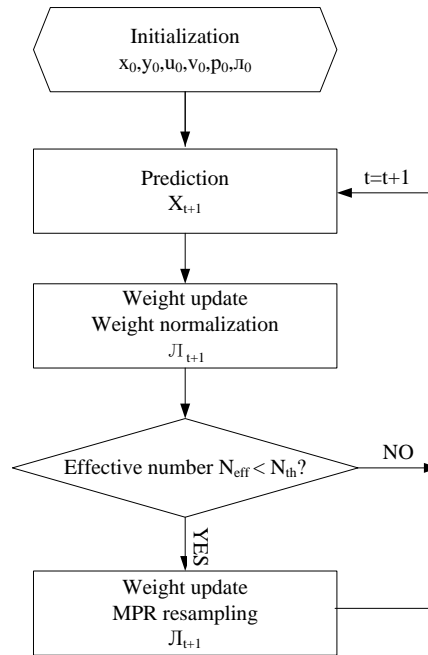


Figure 1. A Diagram of General SISR Algorithm

In the particle filter implementation process, resampling is the last but the most critical operation. Because of the iterating process, a small number of particles dominating the remaining weights might lead to poor approximations for the posterior density and to inferior estimates; only a good resampling method can alleviate the weight degeneracy problem and improve the accuracy of particle filtering.

3. Resampling Scheme and the Improved Algorithms

3.1. Resampling Schemes

It is well known that some of the classic resampling schemes [13] include systematic resampling (SR, Carpenter et al. 1999, Kitagawa 1996) and residual resampling (RR, Higuchi 1997, Liu and Chen 1998), as well as multinomial resampling (MR, Doucet 1998, Pitt and Shephard 1999). Traditional methods may suffer from information loss, due to random number comparison with particles when they equally reset all the weights [14]. On this basis, researchers have worked in the field of resamplings, and a variety of methods have been proposed in the literatures. For example, Bolic introduced new resampling algorithms called partial resampling in examples of [15, 16], which are based on deterministic sampling in residual resampling and partial rejection control proposed by Liu in [11, 12]. Also Chen presented a research about a weight-optimal combination particle filter algorithm [17] in 2009, and Yu proposed a new partial stratified resampling in [18], and some diversity measures for resampling algorithm of PF in [19]. From 2012 to the present, some other advanced resampling strategies have been proposed to modify the only-weight-based resampling. By means of roughening or jittering in [11, 20] to combat the weakness of the particle filter, some ideas are investigated in [21-24] to fight the degeneracy and impoverishment problem of PF.

In the resampling algorithm, each resampled particle is thought equally valid in representing the posterior density associated with the state of the target, thus the process of implementing particles propagation is very important and it is presented in the following diagram.

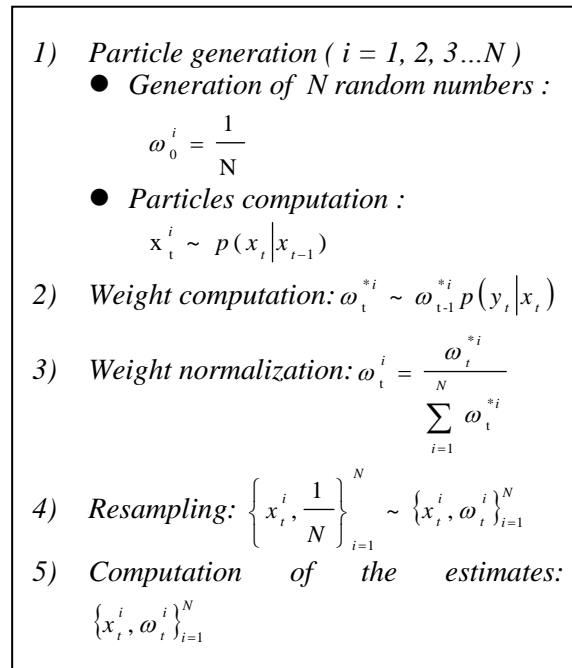


Figure 2. Steps in the Particles Propagation Process

In the following section, we propose an improved partial resampling scheme to deal with particles with different weights by replicating them with a different strategy, which considers the spatial distribution information (e.g., position), and overcomes the side effect of the resampling method at the same time. The definitions proposed are given as follows.

3.2. Three Definitions for the Improved Partial Resampling Scheme

– *Definition 1: A Threshold for Resampling*

In order to alleviate the computational load, resampling should be carried out only sometimes when a criterion is met, rather than after every iteration. Thus we use (8) to help decide whether to do resampling or not. Once \hat{N}_{eff} drops below a given threshold \hat{N}_{th} , resampling steps are necessary to be performed. Since the ESS criterion takes values between 1 and N (N is the total number of particles), $\hat{N}_{th} = \frac{N}{2}$ is chosen as in conventional literatures.

– *Definition 2: Hierarchical Organization for Particle Weights*

In this section, we compare the weights of particles by using a hierarchical organization. First, we also define two thresholds T_l and T_h , where $T_h > T_l > 0$. Then we group the particles into three sets according to their weights: the number of particles with weights greater than T_h and less than T_l can be denoted by N_h and N_l respectively, while the number of particles with weights between T_l and T_h is denoted as N_m . From the theoretical point of view on partial resampling (PR) [16][25], the particles with dominant weights and moderate weights are calculated by function and replicated many times. However, the small particles with weight less than T_l are discarded. An illustration of PDR is given in Fig.3, where there are 12 particles in total in the resampling process.

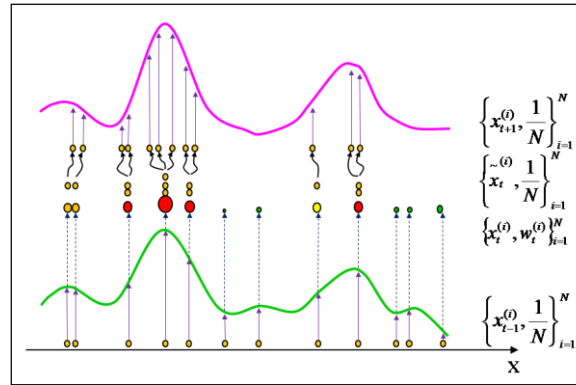


Figure 3. Diagram of a Partial Resampling Process

It can be seen in the above that through the hierarchical organization of particle weights, the red ones with large weights are replicated respectively $\{2, 3, 2, 2\}$ times, while yellow ones with medium weights are all replicated once. It is worth noting that the five green particles in Fig.3 are thought to be negligible and discarded. However, in fact particles with small weights may represent the posterior density associated with spatial distribution information and should not be abandoned all. Thus we propose the method in this work which is a modification of PR algorithm. It is to deal with samples in three sets, with a more efficient use of data below the $N/2$ threshold, thus avoiding the sample impoverishment of particles and improving the approximation of the posterior density estimates. We define the copy times of particles as r , the number of offspring of particles. The hierarchical steps of the algorithm are given as follows:

- 1) Step one: for particles of $\omega^i > T_h$, replicate them a $\max\left(\left\lfloor \frac{\omega^i}{\bar{\omega}} \right\rfloor, 1\right)$ times after normalization, which is denoted by r_h . Also there is $\bar{\omega} = \frac{\sum_{i=1}^N \omega^i}{N}$.
- 2) Step two: for particles of $\omega^i < T_l$, the skip resampling scheme is adopted in definition 3. The corresponding number of replications is denoted by r_l .
- 3) Step three: for particles of $T_h > \omega^i > T_l$, we don't change their weights, then the corresponding number of replications is denoted by r_m .
- 4) Step four: perform residual-systematic resampling(RSR) resampling as in [16]. Since RSR allows for only one iteration loop and processing time that is independent of the distribution of the weights at the input. And the RSR is performed on all particles, weights of that have to be normalized before they are processed by the RSR method.

Of course the numbers of particles always satisfy: $N = N_l + N_m + N_h = r_h + r_m + r_l$.

In the paper [16], three sets of threshold values have been used, i.e. $T_h = \{2/N, 5/N, 10/N\}$ and $T_l = \{1/2N, 1/5N, 1/10N\}$ respectively. After we evaluated these resampling algorithms in experiments, finally $T_h=2/N$ and $T_l=1/2N$ is chosen as a best set scheme due to the minimum RMSE compared with others in simulations.

– *Definition 3: Skip Resampling for Partial Particles*

In this section we will explain how to perform skip resampling in *step two*. Especially for particles with small weights, we will replicate some of them with spacing interval M ($M \geq 1$) from the set of N_l particles, so M can be defined adaptively by following two methods.

- 1) In the first method, spacing interval M is given by,

$$M = \frac{N_l}{N - N_m - r_h} \quad (9)$$

After sorting the particles with low weights in the set of N_l , the algorithm will also perform a skip resampling with the interval M for particles in the set. If $M > 1$, of course $r_l = N_l / M$; if $1 > M > 0$ or $M < 0$, we will not copy particles with low weights and resize $r_m = N - r_h$ at the same time to guarantee $N = N_l + N_m + N_h = r_h + r_m$. This method is proposed as MPR5.

- 2) In the second method, a variable M' is used instead of a fixed value M , which is always updated by half,

$$M' = 1 - \left(\frac{1}{2}\right)^n, n = 1, 2, \dots, r_l \quad (10)$$

After sorting the particles with low weights in N_l , the algorithm will also perform a skip resampling with variable M' for r_l times to particles in the set. However the replicated particles will gradually approach to particles with higher weight in the set by interval of variable M' . This process is similarly to a binary search algorithm in the set of N_l . If $N_l + N_h - r_h > 0$, we will copy particles for r_l times by $r_l = N_l + N_h - r_h$; if $N_l + N_h - r_h < 0$, we will not copy particles with low weights and resize $r_m = N - r_h$. This method is proposed as MPR6.

Through the two improved partial resampling algorithms, the relative size of all the weights will not be changed and the diversity of particles can be maintained because we will replicate dominating particles, moderate particles and even some particles that have small weights. In this way it can avoid the problem of sample impoverishment in PF without increasing much computational load.

4. Simulation and Results

In the following two systems, both known in the literature are used to compare several existing resampling algorithms with the proposed algorithm, in order to verify the performance of the MPR algorithm. Since the main purpose of the MPR algorithm is to optimize PR algorithms while increasing the quality of the estimates of the particle filter, especially on smaller weighted particles. This is desirable in PC simulations and some DSP applications.

■ System A (Univariate non-stationary growth model)

$$x_{k+1} = \frac{x_k}{2} + \frac{25 x_k}{1 + x_k^2} + 8 \cos 1.2k + n_k \quad (11)$$

$$y_k = \frac{x_k^2}{20} + v_k \quad (12)$$

Where $x_0 \sim N(0,5)$, $n_k \sim N(0,10)$ and $v_k \sim N(0,1)$ are independent Gaussian noise sequences. Since particle filters are stochastic in each simulation, the quality

indicator average root-mean-square error (RMSE) of $E[x]$ is used as a performance indicator.

We do MC simulation for 200 times with different numbers of particles changing from 100,500,800 to 1000. The proposed resampling algorithms are compared with other resampling algorithms. The simulation results of average RMSE are shown in Figure 4.

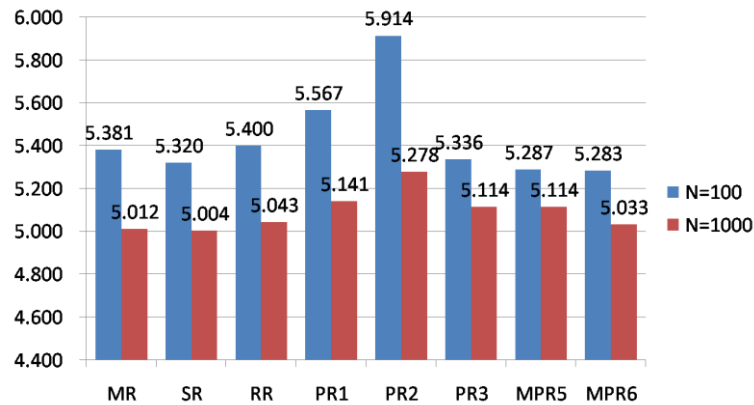


Figure 4. Average RMSE of 8 Resampling Algorithms

It is very clear in the figure that our MPR5 and MPR6 have better performance than than some classic resampling algorithms but also the modified resampling algorithm in [16], no matter $N=100$ or $N=1000$. The average executing time of five PR algorithms are listed as follows.

Table 1. The Running Time Comparisons of Algorithms

PARTICLE NUMBER	RESAMPLING ALGORITHMS	RUNNING TIME(ms)	PARTICLE NUMBER	RESAMPLING ALGORITHMS	RUNNING TIME(ms)
100	NONE	0.0941	1000	NONE	0.5492
100	MR	0.1074	1000	MR	0.6273
100	SR	0.1044	1000	SR	0.6088
100	RR	0.1072	1000	RR	0.6078
100	PR1	0.1593	1000	PR1	1.0693
100	PR2	0.1598	1000	PR2	1.0684
100	PR3	0.1551	1000	PR3	1.0498
100	MPR5	0.1590	1000	MPR5	1.0665
100	MPR6	0.1616	1000	MPR6	1.0677

From the table 1 above, we can see that with N increasing, the performance of PF will be improved and the RMSE reduced, meanwhile it also brings more computational load. However no matter $N=100$ or $N=1000$. The efficiencies of MPR5 and MPR6 are better than those of PR1 and PR2. Though running time of MPR algorithms are somewhat longer than PR3, MR, SR and RR, the increased computational costs is often acceptable which due to the algorithm design of inserting skip resampling in some steps.

■ *System B (Bearings-only tracking model)*

$$X_k = \Phi X_{k-1} + \Gamma \omega_k \quad (13)$$

$$Z_k = \arctan (Y_k / X_k) + v_k \quad (14)$$

In this section we use a two-dimensional scenario with an unknown and time-varying number of targets, where $x_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k]^T$ is the target state vector at time k . Also $v_k \sim N(0,0.01)$, and ω_k is the vector of independent zero-mean Gaussian white noise with $\omega_k \sim N(0, Q_k)$. The covariance matrix is set as $Q_k = \text{diag} \{Q_1, Q_1\}$ and $Q_1 = q^2 \begin{bmatrix} T^4/3 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix}$. The sampling period $T=1$ and $q^2 = 0.1$. In the simulation below, the Monte Carlo simulation is performed 100 times. For comparison, the standard SIS with MR, SR, RR and Stratified resampling for particle filter is also presented. Each resampling algorithm has been run with different numbers of particles $N = 100$ and $N=1000$. Only the results for $N=100$ is given in Fig.5.

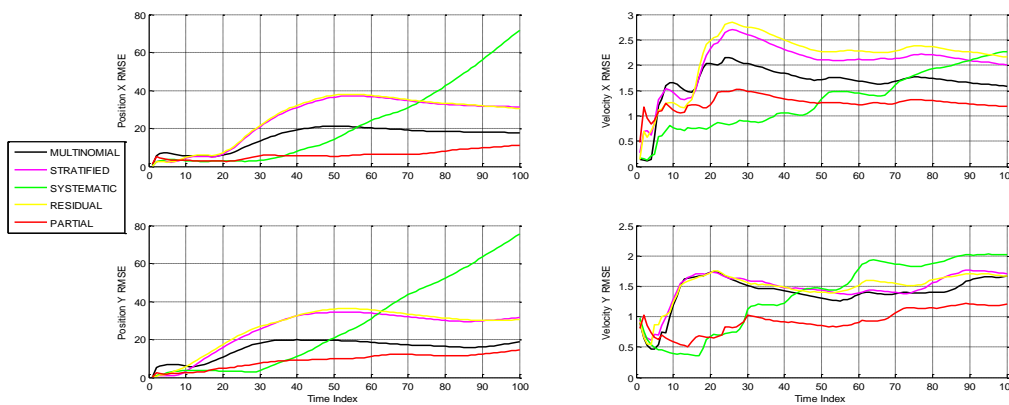


Figure 5. Position and Velocity RMSE Of X, Y

Fig.5 shows average RMSE of the estimated positions and velocities of X, Y. It can be observed that our modified PR algorithm performs more accurately than both the traditional MR, SR, RR and Stratified resampling algorithms in literature. Then average executing time of 9 resampling algorithms are listed as follows.

Table 2. The Running Time Comparisons of 9 Algorithms

PARTICLE NUMBER	PR ALGORITHMS	RUNNING TIME(ms)	PARTICLE NUMBER	PR ALGORITHMS	RUNNING TIME(ms)
100	MR	0.133	1000	MR	7.117
100	Stratified R	0.112	1000	Stratified R	0.275
100	SR	0.098	1000	SR	0.227
100	RR	0.097	1000	RR	1.748
100	PR1	0.74	1000	PR1	4.321
100	PR2	0.743	1000	PR2	4.327
100	PR3	0.693	1000	PR3	4.236
100	MPR5	0.775	1000	MPR5	4.697
100	MPR6	0.785	1000	MPR6	4.987

The results in Table 2 are the average running time of 9 resampling algorithms, the running time will increase along with particle numbers growing but it's not obvious. The proposed resampling algorithm MPR5 and MPR6 both include a classification in three groups and computation of skip resampling, so they run slower than others. It is clear that on average, resampling in MPR algorithms would be performed much faster because the particles with moderate weights are not resampled. The worst case occurs when all the negligible particles must be resampled, and time complexity is $O(N)$, which is an acceptable computation load with the similar good performance as MR algorithm. So in general MPR5 and MPR6 perform averagely better in PR algorithms.

5. Conclusions

As we mentioned in the above, good resampling consists of selecting new particle positions and weights, so that the resampled particle system becomes as good an approximation to $\{x_t^i, \omega_t^i\}_{i=1}^N$ as possible.

In this paper, by analyzing the deficiencies existing in PF, we found that when the starting number of particles is small, particle depletion and sample impoverishment may appear easily. In order to resolve them, we propose two variants of new partial resampling algorithms with adaptive interval M and three definitions to improve estimate quality of PF and efficiency of algorithm by weight classification and particle hierarchical organization. Compared with several resampling algorithms including systematic resampling, multinomial resampling, residual resampling, and some other partial resampling schemes, our method not only combine the merits of both systematic and residual resampling, also improve the performance of the original PR through simulations on univariate non-stationary growth model and bearings-only tracking model. Although the proposed new algorithm are not guaranteed as the theoretically favorable resampling algorithm, simulations followed by performance analysis of PFs show that our proposed algorithms can provide on average better estimates with a good-enough computational efficiency.

Acknowledgments

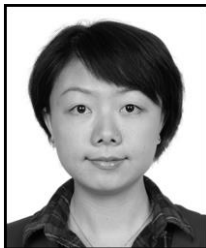
This work was supported by the Artificial Intelligence Key Laboratory in Sichuan University of Science and Engineering of Sichuan Province (No.2012RZJ21) and the National Natural Science Foundation of China (No. 61306094).

References

- [1] O. Cappé, S.J. Godsill and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo", Proceedings of the IEEE, vol. 95, no. 5, (2007), pp. 899-924.
- [2] M.S. Arulampalam, S. Maskell and N. Gordon, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking", Signal Processing, IEEE Transactions on, vol. 50, no. 2, (2002), pp. 174-188.
- [3] N.J. Gordon, D.J. Salmond and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", IEE Proceedings F (Radar and Signal Processing). IET Digital Library, vol. 140, no. 2, (1993), pp. 107-113.
- [4] W.R. Gilks and C. Berzuini, "Following a moving target—Monte Carlo inference for dynamic Bayesian models[J]", Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 63, no. 1, (2001), pp. 127-146.
- [5] C. Musso, N. Oudjane, F. Le Gland, "Improving regularized particle filters[M]", //Sequential Monte Carlo methods in practice. Springer New York, (2001), pp. 247-271.
- [6] S. Park, J.P. Hwang, E. Kim and H. Kang, "A new evolutionary particle filter for the prevention of sample impoverishment", IEEE Transactions on Signal Processing, vol. 13, no. 4, (2009), pp.801-809.
- [7] N.M. Kwok, F. Gu and W. Zhou, "Evolutionary particle filter: re-sampling from the genetic algorithm perspective", 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2005), (2005), pp. 2935-2940.

- [8] J. Zhao and Z. Li, "Particle filter based on particle swarm optimization resampling for vision tracking", *Expert Systems with Applications*, vol. 37, no. 12, (2010), pp. 8910-8914.
- [9] J. Zhong and Y.F. Fung, "Case study and proofs of ant colony optimization improved particle filter algorithm", *IET Control Theory and Applications*, vol. 6, no. 5, (2012), pp. 689-697.
- [10] J.S. Liu and R. Chen, "Sequential Monte Carlo Methods for Dynamic System", *Journal of the American Statistical Association*, vol. 93, no. 443, (1998), pp. 1032-1044.
- [11] J.S. Liu, R. Chen and T. Logvinenko, "A theoretical framework for sequential importance sampling with resampling. Sequential Monte Carlo methods in practice", New York: Springer-Verlag, (2001), pp. 225-246.
- [12] J.S. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling", *Statistics and Computing*, vol. 6, no. 2, (1996), pp. 113-119.
- [13] J.D. Hol, "Resampling in particle filters", *Linköping*, (2004).
- [14] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering", *Image and Signal Processing and Analysis. Proceedings of the 4th International Symposium on. IEEE*, (2005), pp. 64-69.
- [15] M. Bolic, P.M. Djuric and S. Hong, "New resampling algorithms for particle filters", *Acoustics, Speech, and Signal Processing. Proceedings, (ICASSP'03)*, vol. 2, (2003), pp. 2-589.
- [16] M. Bolic, P.M. Djuric and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective", *EURASIP Journal on Advances in Signal Processing*, vol. 15, no. 1900, (2004), pp. 2267-2277.
- [17] J. Chen, P. Yan and Z. Jing-yuan, "Research on weight optimal combination particle filter algorithm", *Computer Engineering and Applications*. vol. 45, no. 24, (2009), pp. 33-35.
- [18] J. Yu, Y. Tang and W. Liu, "Research on diversity measure in particle filter", *Intelligent Computation Technology and Automation (ICICTA) International Conference on. IEEE*, vol. 2, (2010), pp. 1146-1149.
- [19] J. Yu, Y. Tang and J. Xu, "Research on particle filter with adaptive resampling based on diversity measure", *Computer Science*, vol. 39, no. 6, (2012), pp. 231-234.
- [20] T. Li, T.P. Sattar and Q. Han, "Roughening methods to prevent sample impoverishment in the particle PHD filter", *Information Fusion (FUSION), 16th International Conference on. IEEE*, (2013), pp. 17-22.
- [21] T. Li, T.P. Sattar and S. Sun, "Deterministic resampling: unbiased sampling to avoid sample impoverishment in particle filters", *Signal Processing*, vol. 92, no. 7, (2012), pp. 1637-1645.
- [22] T. Li, "A Gap between Simulation and Practice for Recursive Filters: On the State Transition Noise", *arXiv preprint arXiv*, (2013), pp. 1308-1056.
- [23] P.B. Choppala, P.D. Teal and M.R. Frean, "Soft resampling for improved information retention in particle filtering", *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE*, (2013), pp. 4036-4040.
- [24] X. Fu and Y. Jia, "An improvement on resampling algorithm of particle filters", *Signal Processing, IEEE Transactions on*, vol. 58, no. 10, (2010), pp. 5414-5420.
- [25] M. Bolic, "Architectures for efficient implementation of particle filters", New York, USA: State University of New York at Stony Brook, (2004).

Author



Zeng Xiaohui, she received the B.S. and M.S. degrees in communication engineering and automation engineering from the University of Electronic and Science and Technology of China, in 2003 and 2008, respectively. She is now a Ph.D. student at the UESTC, China, and she is also working for Chengdu University of Information and Technology. Her research is related to the development and modification of nonlinear filtering algorithms and tracking signals with efficient artificial intelligence methods implementation.

