Research on Resampling Algorithms for Particle Filter

Xiaohui Zeng [1], Yibing Shi
School of Automation Engineering,UESTC, Chengdu, China
e-mail: huizi003@126.com[1]

Yi Lian [2]
Motorola (China) Solutions, Chengdu, China
e-mail: lianyi8101@foxmail.com

***Abstract*—Particle filter has been widely applied in many fields in recent years. In order to reduce the particle degeneracy problem, some resampling strategies are introduced to improve the real-time performance of particle filtering. In this article, we present many resampling algorithms and their modified strategy, combining with the simulation experiments of two moving-target tracking models and the efficiency of some algorithms are evaluated.**

Keywords-particle filter; particle degeneracy; resampling algorithms; moving-target tracking; performance

I. INTRODUCTION

In the past decades of years, extensive research and work has been done on moving target tracking, especially for improving accuracy, robustness and real-time performance. Among these methods, particle filter has become a popular technique for problems associated with nonlinear non-Gaussian probability distribution since 1990s. As Kalman filter is optimal in the important case when the equations are linear and the noises are independent, additive, and Gaussian [1]. Although various approximate methods such as extended Kalman filter (EKF) and unscented Kalman filter (UKF) are applied widely for nonlinear target tracking, particle filter still has better performance in complex environment with noises.

Particle filter is a sequential Monte Carlo method based on particles of probability densities, which are represented by corresponding weights particles. So the method is also called sequential importance sampling(SIS).There are a lot of researchers working in this field and significant results gained in [2-5], but degeneracy of particles sometimes is very severe so that there are only a few particles significant weights. The degeneracy implies that the performance of the particle filter will deteriorate. There are two methods to reduce the degeneracy, using good importance sampling functions and resampling method. Some authors like Guo et al.[6] present new methods to sample based on Markov chain Monte Carlo(MCMC), and some advanced methods are derived from [7],[8].

This paper is organized as follows. Fundamentals of particle filtering are introduced in section 2 and various resampling schemes in section 3 for particle filtering. Simulation results of two target-tracking models are given in section 4, and conclusion is described in section 5.

II. FUNDAMENTAL OF PARTICLE FILTERING

Generally, the framework of Bayesian tracking consists of estimating dynamic state of objects in a nonlinear stochastic system, and a set of noisy observations. The time-varying object state is denoted by state vector x_n , and observation is denoted by observation vector, as the distribution $p(x_n|x_{n-1})$, where n indicates time. The model can be written in the form as follows,

$$\begin{aligned} x_n &= f(x_{n-1}, u_n), \quad \textit{State Transition Model} \\ y_n &= h(x_n, v_n), \quad \textit{Observation Model} \end{aligned} \tag{1}$$

where $f(\cdot)$ and $h(\cdot)$ are some known functions, and u_n and v_n are random noise vectors of given distributions. In many practical situations, it's difficult to draw samples from posterior distribution $p(x)$, thus another probability distribution $q(x)$ is used instead. All particles can approximate the distribution $q(x)$, and the density $p(x)$ is given by,

$$\hat{p}(x) = \frac{\sum_{i=1}^{N} \tilde{\omega}^i \delta(x - x^i)}{\sum_{i=1}^{N} \tilde{\omega}^i} \tag{2}$$

where

$$\tilde{\omega}^i = \frac{p(x^i)}{q(x^i)} \tag{3}$$

and $\delta(\cdot)$ is a Dirac delta function defined by,

$$\delta(x - x^i) = \begin{cases} 1 & x = x^i \\ 0 & otherwise \end{cases} \tag{4}$$

If the samples are drawn from an importance function $q(x_{1:n}|y_{1:n})$, then the weights are decided as follows,

$$\tilde{\omega}^i = \frac{p(x^i_{1:n}|y_{1:n})}{q(x^i_{1:n}|y_{1:n})} \tag{5}$$

So posterior distribution $p(x_{1:n}|y_{1:n})$ can be approximated with a new set of samples $\{x^i_{1:n}, \omega^i_n\}_{i=1}^N$, and the method is called importance sampling.

From the Bayesian theory, we can obtain an updated weight equation as follows, and a "weighted" approximation of $\tilde{p}(x_{1:n}|y_{1:n})$:

$$\tilde{\omega}_n^i = \tilde{\omega}_{n-1}^i \frac{p(y_n|\tilde{x}_n^i)p(\tilde{x}_n^i|\tilde{x}_{n-1}^i)}{q(x_n^i|x_{1:n-1}^i, y_{1:n})} \tag{6}$$

$$\tilde{p}(x_{1:n}|y_{1:n}) = \sum_{i=1}^{N} \omega_n^i \delta_x(x_{1:n}) \tag{7}$$

By normalizing the weights with,

$$\omega_n^i = \frac{\tilde{\omega}_n^i}{\sum_{j=1}^{N} \tilde{\omega}_n^j} \tag{8}$$

The sequential importance sampling method suffers a serious drawback: after a few iteration steps all but one of the normalized weights are very close to zero. As we mentioned above, with n increasing, the SMC estimate deteriorates. Because the degeneracy phenomenon is unavoidable, Liu et al. [5] introduced an approximate method to measure particle degeneracy by,

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (\omega_n^i)^2} \tag{9}$$

We can get the implementation of SISR algorithm. There are three steps:

1) Step One: Importance Sampling.

a) Initialization: according to state prior distribution, we can establish initial particles set sampling from,

$$\tilde{x}_n^i \sim q(x_n|x_{1:n-1}^i, y_{1:n}) \tag{10}$$

b) Get new set of particles: for $i = 1,2,\cdots,N$, through evaluating the importance weights from (6), get the new set of particles.

c) Normalize importance weights: for $i = 1,2,\cdots,N$, according to (7), normalize the importance weight of particles.

d) With (8), we can know that the smaller $\hat{N}_{eff}$, the worse the degeneracy. If $\hat{N}_{eff} < N_{threshold}$ then go to step three. Otherwise go to step two.

2) Step Two: Outputting Estimation.

a) When $\hat{N}_{eff} \geq N_{threshold}$, posterior distribution $p(x)$ can be approximated, and state estimation can be gained from $x_n^i = E(x_n^i) = \sum_{i=1}^{N} \omega_n^i \tilde{x}_n^i$.

b) Make the timing step from t to t + 1 and return back to step *one* to continue.

3) Step Three: Resampling.

a) For $i = 1,2,\cdots,N$, draw particles from $\{\tilde{x}_{1:n}^i, \omega_n^i\}_{i=1}^N$ and sample from an index $j(i)$ distributed with *N* elements which satisfies $P_r\{j(i) = l\} = w_n^l$, where $l = 1,2,\cdots,N$. Replace the current particle set with new ones.

b) Reset weights of particles as $\omega_n^i = 1/N$ and for $i = 1,2,\cdots,N$, $\tilde{x}_{1:n}^i = \tilde{x}_{1:n}^{j(i)}$.

So in the implementation of PF, resampling is the last but critical operation in particle filtering. Because with time, a small number of weights dominating the remaining maybe lead to poor approximation of the posterior density and inferior estimates, good resampling algorithm can alleviate the weight degeneracy problem and improve performance of particle filtering.

III. VARIOUS RESAMPLING SCHEMES

As we mentioned above, good resampling algorithm consists in selecting new particle positions and weights, so that discrepancy between the resampled weights can be reduced, meanwhile the resampled particle system be as good an approximation to $\{\tilde{x}_{1:n}^i, \omega_n^i\}_{i=1}^N$ as possible.

A. Basic Resampling Algorithms

From the literature, four traditional basic resampling algorithms can be identified such as Simple Random Sampling, Stratified Resampling, Residual Resampling (RR), and Systematic Resampling (SR) [9]. All of them can be used in the basic (SISR) particle filter.

- Simple Random Sampling, also called Multinomial Resampling (MR): it samples $N_n^{1:N}$ from a multinomial of parameters $(N, W_n^{1:N})$.
- Stratified Resampling: it distribute the samples evenly or unevenly according to their respective variance to the different strata dividing the whole space. Then an estimate can be gained from importance sampling combining independent simple random samples within every stratum.
- Residual Resampling (RR): it sets $\tilde{N}_n^i = \lfloor NW_n^i \rfloor$ firstly, then sample $\overline{N}_n^{1:N}$ from a multinomial of parameters $(N, \overline{W}_n^{1:N})$, where $\overline{W}_n^i \propto W_n^i - N^{-1}\tilde{N}_n^i$, then set $N_n^i = \tilde{N}_n^i + \overline{N}_n^i$.
- Systematic Resampling (SR): similar to stratified resampling, the interval is divided into N strata and one sample is taken from every stratum, but the samples are no longer independent. They have same position within a stratum that gives the minimal discrepancy for N samples.

And in many scenarios, residual resampling, stratified resampling and systematic resampling are better than simple resampling as they use variance reducing methods, from a theoretical point of view in [10]. However in practice, systematic resampling is easy to implement and outperforms other resampling schemes in most scenarios, it is often most widely-used resampling scheme. These four basic

resampling algorithms can solve the divergence problem of PF in many cases, but also increases in the computational cost and other problems.

B. Threshold-based Partial Resampling Algorithms

Since M. Bolic put forward threshold-based algorithms in[11] called Partial Resampling (PR) which combines the merits of both systematic and residual resampling, also reduces complexity and processing time. The concept of partial resampling is to do resampling only on particles with large weights and replace them with particles with unequal weights, while particles with moderate weights are not resampled. This method is also called Partial Deterministic Resampling (PDR) illustrated in [12].

Firstly, there are two defined threshold T_l and T_h, where $T_h > 1/N$ and $T_h > T_l > 0$ .Then we group the particles according to their weights into three sets, i.e. 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. And the resampling function is given as follows,

$$\widetilde{\omega}_n^{*(i)} = \begin{cases} \omega_n^{(i)} / r + W_l / (N_h + N_l) & \omega_n^{(i)} > T_h \\ \omega_{n-1}^{(i)}, & T_h > \omega_n^{(i)} > T_l \\ 0, & otherwise \end{cases} \quad (11)$$

W_l are negligible weights which satisfies $W_l = \sum_{i=1}^{N_l} \omega_n^i$ and $\omega_n^i < T_l$. So the particles with dominant weights are their weights calculated by function above and replicated r times. About r, the first $N_t = N_l - \lfloor N_l / N_h \rfloor N_h$ dominating particles are replicated $r = \lfloor N_l / N_h \rfloor + 2$ times, while the rest of $(N_h - N_t)$ dominating particles are replicated $r = \lfloor N_l / N_h \rfloor + 1$ times. In the paper [12], there are three PR resampling algorithms and three sets of threshold values were used, i.e. $T_h = \{2/N, 5/N, 10/N\}$ and $T_l = \{1/2N, 1/5N, 1/10N\}$ respectively. And we evaluate these 9 resampling algorithms in our experiment, that are denoted by PR1with TH1, PR1with TH2, PR1with TH3; PR2with TH1, PR2with TH2, PR2with TH3; PR3with TH1, PR3with TH2, PR3with TH3.

C. Weight Optimal Resampling Algorithms

Sophisticated techniques that help generate better resamples in the PF [13][14][15][16] have been developed in the computational efficiency of the method. And weight optimal resampling algorithms (WOR) are put forward to alleviate the weight degeneracy and sample impoverishment. The weight optimal function is given as follows,

$$\overline{\omega}_n = \frac{\sum_{i=1}^{N} \omega_n^i}{N} \quad (12)$$

$$\psi_n^i = \frac{K-1}{K} \omega_n^i + \frac{\overline{\omega}_n}{K} \quad (13)$$

Where K is proportionality factor ($K \geq 1$). We choose $K = 3$ and $K = 5$ to verify the performance of algorithms, denoted as WORS K1 and WORS K2.

Through the weight optimal algorithm, relative size of all weights will not be changed and diversity be maintained. Therefore probability density distribution before and after resampling is the same.

IV. Simulation and Results

In order to evaluate the different resampling algorithms in particle filter, we designed simulation program using MATLAB R2012a with 50 MC trials of resampling schemes in following two systems, UNGM and BOT. They are both known in the literature, which are used to compare the different 15 resampling algorithms for verifying the performance of the algorithms.

A. System A (Univariate non-stationary growth model)

$$\begin{aligned} x_{k+1} &= \frac{x_k}{2} + \frac{25 x_k}{1 + x_k^2} + 8\cos(1.2k) + n_k \\ y_k &= \frac{x_k^2}{20} + v_k \end{aligned} \quad (14)$$

With $x_0 \sim N(0,5)$, $n_k \sim N(0,10)$ and $v_k \sim N(0,1)$ is independent Gaussian noise sequences. Hence the quality indicator RMSE of $E[x]$ is used as a performance indicator.

We simulate these 15 resampling algorithms with different numbers of particles changing from 1000, 500 to 100 while using the same measurement data. The simulation results are shown in figure 1. The RMSE of standard PF without resampling is the highest, and the others are different in values, some of which can be clearly listed in the table 1.

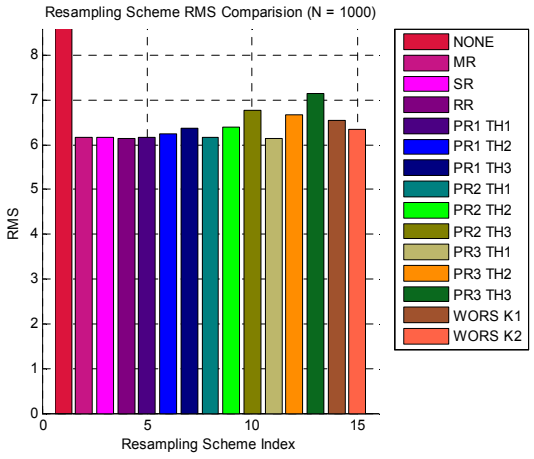

Figure 1. RMSE Comparison of 15 Resampling Algorithms for PF

TABLE I. RMSE VALUES OF ALGORITHM

Particle Numbers	RMSE	
	Maximum RMSE	*Minimum RMSE*
N = 1000	PR3 TH3, 7.158810	PR3 TH1, 6.150171
N = 500	PR3 TH3, 7.301530	PR3 TH1, 6.406826
N = 100	PR3 TH3, 7.37699	PR1 TH1, 6.257699

From perspective of [12], the most promising PR3 algorithm is the simplest one and it requires the smallest size of memory. However from experiment we can see that different thresholds affect performance of PF greatly, e.g. TH1 and TH3, between that one is the best the other is the worst.

Then we compare the running time of 15 resampling schemes as follows.

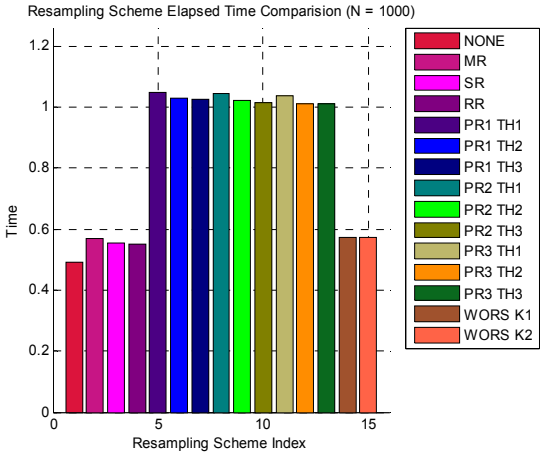

Figure 2. Time Comparison of 15 Resampling Algorithms for PF

We can see that running time of standard PF without resampling is obviously the lowest, the WOR algorithms are much better than PR algorithms because its simple algorithm structure and meanwhile keeping diversity. Some simulation data can be clearly shows in the table below.

TABLE II. RUNTIME OF PROGRAM

Particle Numbers	PROCESS TIME	
	Maximum runtime	*Minimum runtime*
N = 1000	PR1 TH1, 1.049375	RR, 0.549063
N = 500	PR1 TH1, 0.547813	SR, 0.284687
N = 100	PR1 TH1, 0.135937	SR, 0.069687

When particle number is less than 500, SR is the most commonly used since it is the fastest resampling algorithm for computer simulations, with lower conditional variance for all configurations of the weights. But PR1 is not a good choice when you want to get much faster performance.

B. System B (Bearings-only tracking model)

$$X_k = \Phi X_{k-1} + \Gamma \omega_k$$
$$Z_k = \arctan\left(\frac{Y_k}{X_k}\right) + v_k \tag{15}$$

A two-dimensional scenario with an unknown and time-varying number of targets, where $X_k = [x_k \quad \dot{x}_k \quad y_k \quad \dot{y}_k]^T$ is target state vector at time k, and $v_k \sim N(0, 0.005)$, and ω_k is the vector of independent zero-mean Gaussian white noise with $\omega_k \sim N(0, Q_k)$. Covariance matrix is set as $Q_k = diag\{Q_1, Q_1\}$ and $Q_1 = q^2 \begin{bmatrix} T^4/3 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix}$. The sampling period is $T = 1$. In the simulation below, WORS K1, WORS K2 and WORS K3 are adopted, while standard PF with none resampling is removed. And results are given as following figure ($N = 100$).

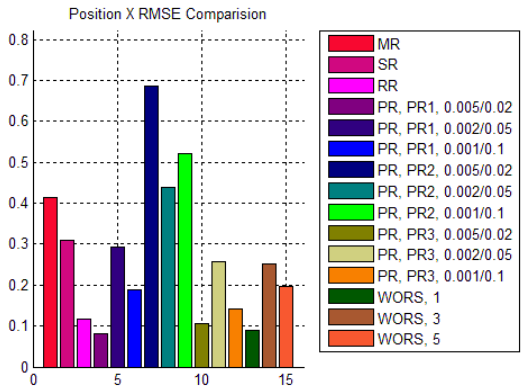

Figure 3. RMSE Comparison of X estimated position

And other figures of RMSE of Y position, velocity of $\dot{x}_k$ and $\dot{y}_k$ are omitted here, it is already can be inferred that the estimated positions and velocity based on these schemes are similar. We could infer that,

- For basic traditional resampling algorithms, the performance of PF can be shown in this order: RR > SR > MR.
- When compared with modified resampling algorithms ($N = 100$), it is found that PR1 with TH1, PR3 with TH1 also have smaller RMSE than others, which is similar with results from model UNGM. Thus we can adopt a strategy of keeping thresholds for $(1/2N, 2/N)$ the reasonable choice for proposed threshold-based resampling.

Then we compare the resampling time of 15 resampling schemes in following figure.

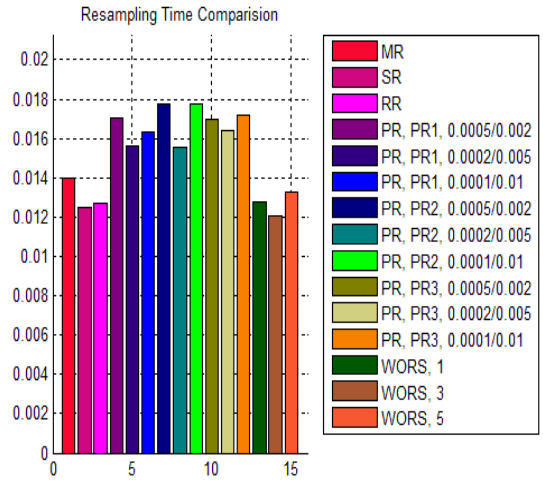

Figure 4. Resampling time comparison of 15 resampling algorithms

From figure 4 above, we can also get same results as in UNGM that especially when N=1000, simulation time of 15 algorithms are very long among which RR and SR are better. So a larger particle number leads to higher estimation accuracy, which requires more computation cost at the same time. To get a balance between the tracking accuracy and the computation cost is a problem worthy of consideration.

V. CONCLUSION

In this paper, by analyzing some resampling algorithms including systematic resampling, multinomial resampling, residual resampling and some modified resampling and so on, we hope to get a deeper research on different resamplings to improve the deficiency in PF. Resampling step can alleviate the weight degeneracy problem, however it suffers from the high computation cost and sample impoverishment. It means that the diversity of the particles is reduced due to the fact that particles with a high importance weight will be selected many times.

So according to analysis of PFs in two models, UNGM and BOT, simulation results show that using a different resampling algorithm might increase the quality of the estimates of the particle filter. And we further discuss how to determine which resampling scheme to choose in target-tracking model with different particle numbers and different conditions. As proposed new resampling algorithms whose processing time is often required and should be more suitable for hardware implementation. While some algorithms reduce complexity remarkably, but minimize performance degradation. Although improvements are not guaranteed using theoretical favorable resampling algorithms, simulations show the right resampling algorithms on average better estimates in PF.

ACKNOWLEDGMENT

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

REFERENCES

[1] Sorenson H W. Least-squares estimation: from Gauss to Kalman [J]. Spectrum, IEEE, 1970, 7(7),pp.63-68.

[2] Arulampalam M S, Maskell S, Gordon N, et al. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking [J]. Signal Processing, IEEE Transactions on, 2002, 50(2),pp. 174-188.

[3] Doucet A, De Freitas N, Gordon N. Sequential Monte Carlo methods in practice[M]. New York: Springer, 2001.

[4] Gilks W R, Berzuini C. Following a moving target—Monte Carlo inference for dynamic Bayesian models[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2001, 63(1),pp. 127-146.

[5] Liu J S. Metropolized independent sampling with comparisons to rejection sampling and importance sampling [J]. Statistics and Computing, 1996, 6(2): 113-119.

[6] Guo, Dong, Xiaodong Wang, and Rong Chen. "New sequential Monte Carlo methods for nonlinear dynamic systems." Statistics and Computing 15.2 (2005): 135-147.

[7] Doucet, Arnaud, Mark Briers, and Stéphane Sénécal. "Efficient block sampling strategies for sequential Monte Carlo methods." *Journal of Computational and Graphical Statistics* 15.3 (2006).

[8] Atchadé, Yves F., and Jeffrey S. Rosenthal. "On adaptive markov chain monte carlo algorithms." *Bernoulli* 11.5 (2005): 815-828.

[9] Hol J D. Resampling in particle filters[D]. Linköping, 2004.

[10] Douc R, Cappé O. Comparison of resampling schemes for particle filtering[C], Image and Signal Processing and Analysis, 2005. Proceedings of the 4th International Symposium on. IEEE, 2005: 64-69.

[11] Bolic M, Djuric P M, Hong S. New resampling algorithms for particle filters[C], Acoustics, Speech, and Signal Processing. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. IEEE, 2003, 2: II-589-92 vol. 2.

[12] Bolić M, Djurić P M, Hong S. Resampling algorithms for particle filters: A computational complexity perspective[J]. EURASIP Journal on Advances in Signal Processing, 1900, 2004(15): 2267-2277.

[13] Yu J, Tang Y, Liu W. Research on Diversity Measure in Particle Filter[C], Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on. IEEE, 2010, 2: 1146-1149.

[14] Yu J, Tang Y, Xu Jingmin. Research on ParticIe Filter with Adaptive Resampling ßased on Diversity Measure[J], Computer Science. Vol. 39 No.6,June 2012:231-234.

[15] Shi Z G, Zheng Y, Bian X, et al. Threshold-based resampling for high-speed particle PHD filter[J]. Progress In Electromagnetics Research, 2013, 136: 369-383.

[16] CHEN Jian, YAN ping, ZHANG Jing-yuan. Research on weight optimal combination particle filter algorithm. Computer Engineeang and Applications. 2009. 45(24):33-35.