

An improved particle filter based on genetic resampling

ZHAO Bin^{1,a}, HU Jian-wang^{1,a}, JI Bing^{1,a}

¹Department of Information Engineering, Ordnance Engineering College, Shijiazhuang 050003, China

^a871698474@qq.com

Key words: particle filter; genetic algorithm; sample impoverishment; resampling

Abstract. The resampling of particle filter algorithm causes sample impoverishment and results in the loss of diversity. An improved particle filter algorithm is proposed by combining genetic algorithm and PF. First of all, particles are selected according to the probability with the principle of selection in genetic algorithm which make particles with larger weights to be selected more likely, and the proposed algorithm can guide the entire process to the direction of evolution. Crossover and mutation are adopted to replace the strategy of resampling which simply copy particles with high weight, delete particles with low weight and implement the particle update and optimization. Finally, the simulation experiments show that the proposed algorithm can more effectively improve the diversity of particles.

Introduction

The particle filter [1] is mainly applied to approximately estimate Bayes probability, which is achieved by Monte Carlo method and applicable to any nonlinear stochastic system. Compared to KF, EKF, UKF method, it has great advantages in dealing with nonlinear and non-Gaussian system state estimation and has been widely applied in many fields such as target tracking [2]. However, in the application process, the particle filter algorithm will appear the phenomenon of particle degradation [3,4] after several iterations. Usually, there are two method to solve the problem in the framework of particle filtering: one is to improve the importance function by using better suggestion function [5], such as auxiliary particle filter and regularized particle filter and unscented particle filter to improve the quality of the Kalman particle; the other one is the introduction of resampling process [6], such as stratified resampling, systematic resampling, cumulative distribution resampling, residual resampling etc.. The algorithm above-mentioned is mainly to improve the effectiveness of the particle to solve the particle degeneration problem, when the resampling process completed, particle with small weight is removed, and particle with big weight is selected repeatedly. Although in a certain extent, the resampling process restrain the particle degeneracy phenomenon, the strategy adopted will damage the diversity of particle and cause sample impoverishment.

The literature [7] use genetic algorithm to improve the initial particle quality, ensuring the randomness and statistics, reducing the possibility of particle degradation, but there is no prominent effect on solving the problem of sample impoverishment. The genetic algorithm is applied to the particle resampling in literature [8], but in the selection operation, it just uses a single threshold, which restricts the offspring number of crossover and mutation operation, affecting the particle performance. In view of this, considering the filtering performance and the computational complexity, this paper presents an improved particle filter based on genetic resampling, the algorithm retains the standard particle filter resampling strategy, and on the basis, the crossover and mutation are introduced, to keep the diversity of particle and overcome the problem of particle degeneracy.

The particle filter principle

The principle of particle filter [9] is based on IS(Importance Sampling), through finding a set of random samples spreading in the state space to approximate the probability density function, and

using sample mean to replace integration to obtain the minimum variance estimation [10].

The nonlinear model is given by

$$x_{k+1} = f(x_k, w_k) \quad (1)$$

$$y_k = h(x_k, v_k) \quad (2)$$

Where x_k is system state variable, y_k is measurement, w_k and v_k are process noise and measurement noise, respectively.

Assuming that the important density function $\pi(x_{0:k} | y_{1:k})$ meet the requirements, a set of particles $\{x_{0:k}^{(i)}, i = 1, 2, \dots, N\}$ are sampled from the known proposal distribution $\pi(x_{0:k} | y_{1:k})$, so the estimation of posterior mean is

$$\hat{I}[f(x_{0:k})] = \sum_{i=1}^N \tilde{\omega}_k^{(i)} f(x_{0:k}^{(i)}) \quad (3)$$

Where $\tilde{\omega}_k^{(i)}$ is normalized weights, which can be expressed as

$$\tilde{\omega}_k^{(i)} = \frac{\omega_k(x_{0:k}^{(i)})}{\sum_{i=1}^N \omega_k(x_{0:k}^{(i)})} \quad (4)$$

Genetic resampling particle filter (GRPF)

Genetic algorithm (GA) was proposed by J.Holland^[11] in 1975, which is a random search optimization technique based on the principle of natural selection and heredity and realize the update and optimization of population by selection, crossover and mutation. Its main characteristic is to have internal parallelism and better global search ability; it uses optimization method of randomization which can adaptively adjust the search direction to automatically get the optimization search space and it needn't to determine the rules. Genetic algorithm keep an optimal population in each iteration, not just a best individual, and make the evolution of the whole population. Because of the good optimization performance of genetic algorithm, it is introduced into the resampling process in particle filter in this paper to overcome the problem of sample impoverishment.

Selection operation. The operation of the main method from sample particles with probability of as much as possible to select high value of the particle, in order to select particles as genetic operation parent particle, ensure excellent sex of parent individuals.

Taking the particle weights as the fitness of individuals, each individual fitness $\tilde{\omega}_k^{(i)}$ is

$$\tilde{\omega}_k^{(i)} = \omega_k^{(i)} / \sum_{i=1}^N \omega_k^{(i)} \quad (5)$$

$i = 1, 2, \dots, N$, which can obtain the cumulative probability of each individual $c_k^{(i)}$

$$c_k^{(i)} = \sum_{j=1}^i \tilde{\omega}_k^{(j)}, \quad i = 1, 2, \dots, N \quad (6)$$

Generating a random number u in the interval $[0,1]$, that is $u \sim U[0,1]$. There must be a $j \in [1, N]$ to make $c_k^{(j)} \leq u < c_k^{(j+1)}$, then selecting the particle $x_k^{(j)}$, repeating the operation N times, and choosing N particles.

Crossover operation. Assuming that the crossover probability is P_c , generating a random number $u \sim U[0,1]$. If $u < P_c$, choosing two particle $\{x_k^{(i)}, x_k^{(j)}\}$ randomly. According to the following rule to cross, and use the new individuals $\{\tilde{x}_k^{(i)}, \tilde{x}_k^{(j)}\}$ after crossover to replace the old individuals $\{x_k^{(i)}, x_k^{(j)}\}$.

$$\begin{aligned}\tilde{x}_k^{(i)} &= \alpha x_k^{(i)} + (1-\alpha)x_k^{(j)} \\ \tilde{x}_k^{(j)} &= \alpha x_k^{(j)} + (1-\alpha)x_k^{(i)}\end{aligned}\quad (7)$$

Where $\alpha \in [0,1]$, its values depend on the circumstances.

Mutation operation. Assuming that the mutation probability is P_m , generating a random number $u \sim U[0,1]$. If $u < P_m$, a particle $x_k^{(i)}$ is randomly selected to accept the mutation operation and replacing the old particle $x_k^{(i)}$ with the particle after mutation $\tilde{x}_k^{(i)}$.

$$\tilde{x}_k^{(i)} = x_k^{(i)} + \eta \quad (8)$$

Where $\eta \sim N(0, \sigma)$.

Genetic resampling particle filter algorithm flow. As mentioned above, using the traditional resampling methods to copy the high weight particles and remove low weight particles will cause the sample impoverishment. Therefore, the genetic algorithm is introduced into the particle filter resampling process which does not simply copy or delete particles, but utilize crossover, mutation operation in genetic algorithm to deal with the particle, the specific process is as follows.

Step one: initialization. When $k = 0$, N samples $\{x_0^{(i)}\}_{i=1}^N$ is obtained from $p(x_0)$, and all the particles weight are the same, which is $\omega_0^{(i)} = 1/N$.

Step two: importance sampling

(1) The particles $\{x_k^{(i)}\}_{i=1}^N$ is sampled from importance density function in k moments. When using the optimal sampling function, the algorithm is very difficult to achieve. A simple and useful alternative method is choosing $p(x_{k+1} | x_k)$ as the importance density function.

$$\pi(x_k | x_{k-1}^{(i)}, y_k) = p(x_k | x_{k-1}^{(i)}) \quad (9)$$

Through sampling from the function, we can particles $x_k^{(i)} \sim p(x_k | x_{k-1}^{(i)})$.

(2) the particle weight calculated by $\omega_k^{(i)} = p(y_k | x_k^{(i)})$ and normalized by $\tilde{\omega}_k^{(i)} = \omega_k^{(i)} / \sum_{i=1}^N \omega_k^{(i)}$.

Step three: judgment resampling condition. Calculate the number of effective samples N_{eff} [12], the smaller N_{eff} , the serious degradation, which is defined as $N_{eff} = N / (1 + \text{var}\{\tilde{\omega}_k^{(i)}\})$. Usually, this value cannot be precisely calculated, generally using $\hat{N}_{eff} = 1 / \sum_{i=1}^N \tilde{\omega}_k^{(i)}$ to approximate, it can jump to step five when \hat{N}_{eff} meet the requirements, otherwise executing step four continually.

Step four: resampling

The specific process as shown in Fig 1.

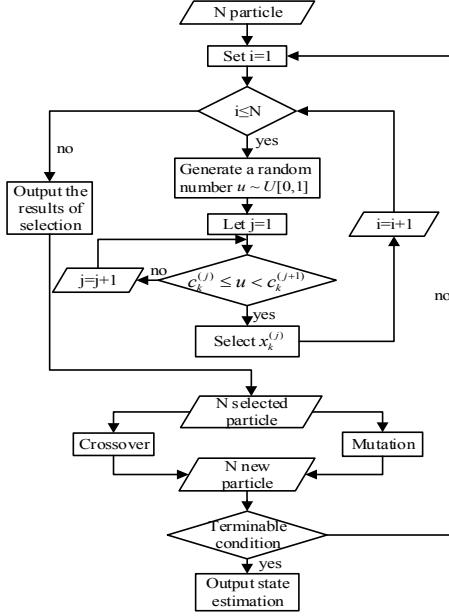


Fig.1 The flow chart of genetic resampling

Initially, N particles are randomly selected as the parent generation of genetic operation according to the weight. The particles with larger weight are repeatedly selected, on the contrary the particles with smaller weights aren't easily selected. Then, crossover and mutation are executed with a certain probability to generate N new particles, and the particles weights are updated continually. When the genetic operation meet the terminable conditions of iteration, the iteration is terminated to move to the next step.

Step five: state estimation. Estimation of K moment is

$$\hat{x}_k = \sum_{i=1}^N \tilde{\omega}_k^{(i)} x_k^{(i)} \quad (10)$$

Step six: time update

Set $k = k + 1$, turn to step two.

Simulation experiment and analysis

In order to verify the proposed algorithm performance, this paper selects the highly nonlinear system model, of which the state equation is

$$x_k = 0.5x_{k-1} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos[1.2(k-1)] + w_k \quad (11)$$

The measurement equation is

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

Where $w_k \sim N(0, Q)$, $v_k \sim N(0, R)$ both are zero mean Gaussian noise; assuming that the system initial parameters are $x_0 = 0.1$, $Q = 2$, $R = 5$, crossover probability is $P_c = 0.9$, mutation probability is $P_m = 0.1$, genetic factor are $\alpha = 0.5$, $\sigma = 5$, the amount of particles is 1000, the number of iterations is set to 50.

Experiment 1, the verification of algorithm estimated performance. The results of PF and GRPF are compared in the simulation experiment 1 ,as shown in Fig 2. In order to more intuitively illustrate the proposed algorithm performance, two kinds of algorithm are taken 10 times independent experiments respectively, each result is achieved across 50 Monte-Carlo simulation, the estimation mean square error is shown in Fig 3

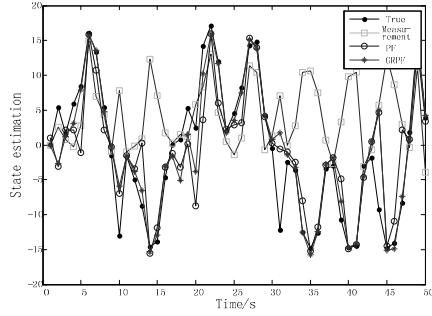


Fig.2 The state estimation results

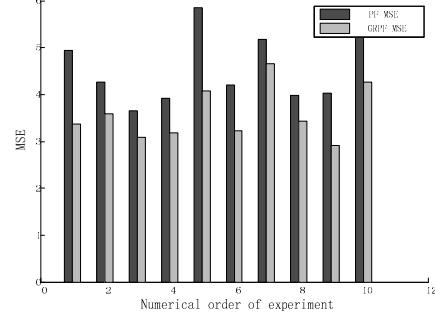


Fig.3 State estimation mean square error

In Fig 2, both two algorithms can get accurate state estimation in nonlinear model, but compared to PF, GRPF has better effect, the estimation results are closer to the true value.

We can see from Fig 3, the mean square error of the proposed algorithm is less than the standard particle filter algorithm, the proposed algorithm effectively improve the estimation accuracy and the filter performance.

Experiment 2 validation resampling performance. The above mentioned, the proposed algorithm can effectively improve the diversity of particles and solve the problem of sample impoverishment, the explanation is offered according to the simulation results. Through the simulation experiment, the particle distribution and the variance of weights in PF and GRPF are obtained respectively, as shown in Fig 4 and fig 5

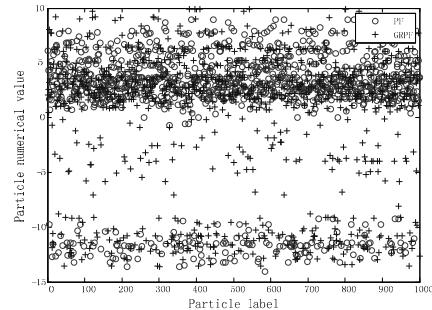


Fig.4 The distribution of particles

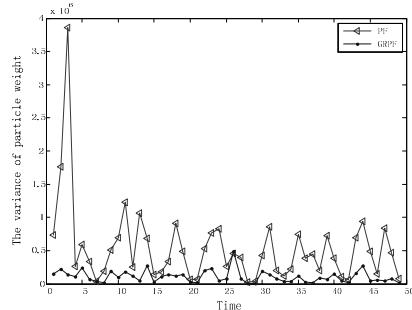


Fig.5 The particles weight variance

It is obvious that the particle distribution of GRPF is wider than PF in Figure 4, which increase the diversity of particles and solve the problem of sample impoverishment. However, only through the Figure 4, it's not enough to illustrate that all these particles are effective, but we can get the result in Figure 5, which shows the MSE of particle weight after 50 times Monte-Carlo simulation experiment. As can be seen, the MSE of PF algorithm is larger than GRPF in recursive, due to the highly nonlinear.

These can explain that GRPF algorithm have the ability to improve particle diversity and the effectiveness of particle, solve the problem of particle degradation. The retained particles play a greater contribution in state estimation, which makes the algorithm can still approximate the PDF of true state in highly nonlinear system and maintain a higher filtering accuracy.

Conclusion

This paper presents GRPF algorithm, which combines Monte-Carlo method with genetic algorithm to overcome the problem of sample impoverishment. The resampling process of PF is improved by genetic algorithm, which keep the advantage of resampling to deal with particle degeneration, besides, it can improve the diversity and efficiency of particles and get the state estimation accurately. The simulation results show that the performance of proposed algorithm is better than the traditional PF algorithm.

References

- [1] Gordon N J, Salmond D J, Smith A F M. Novel approach to nonlinear/non-gaussian bayesian state estimation[J]. IEEE Proceedings on Radar,Sonar and Navigation,1993, 140(2):107-113.

- [2] Phani Chavali,Arye Nehorai.Concurrent Particle Filtering and Data Association Using Game Theory for Tracking Multiple Maneuvering Targets[J]. IEEE Transactions on Signal Processing, 2013,61(20):4934- 4948.
- [3] Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering[J]. Statisticsand Computing, 2000(10) : 197-208 .
- [4] Kong A, Liu J, Wong W H. Sequential imputations and Bayesian missing data problems[J]. Journal of the American Statistical Association,1994,89(425) : 278-288 .
- [5] ZHANG Miao, HU Jian-wang, ZHOU Yun feng, et al. Comparison of Improved Particle Filtering Algorithms[J]. Electronics Optics & Control, 2009, 16(2):30-32.
- [6] Arulampalam M S, Maskell S, Gordon N,et al.A tutorial on particle filters for on-line nonlinear non-Gaussian Bayesian tracking[J]. IEEE Trans.on Signal Processing, 2002, 50(2):174-188.
- [7] YANG Ning, QIAN Feng, ZHU Rui. Improved Particle Filter Based on Genetic Algorithm[J]. JOURNAL OF SHANG HAI JIAO TONG UNIVERSITY, 2011,45(10):1526-1530.
- [8] XI Tiao, ZHANG Sheng-xiu, YUAN Kui, et al. Video Object Tracking Based on Particle Filter with Genetic Evolution Strategy[J]. Opto-Electronic Engineering, 2009,36(3):28-32.
- [9] RONG Zhan-hui, ZHAGN Jun, et al.Nonlinear Filtering Theory with Target Tracking Application[J]. BeiJing: National Defense Industry Press,2013
- [10]LI Xiao-wei, HU Zhen-tao, PAN Quan, et al. An Improved Particle Filtering Algorithm Based on Fuzzy Support Degree Sampling[J]. 2012,20(1):190-192.
- [11]J Holland. Adaptation in natural and artificial systems[M]. Ann Arbor:U niversity of Michigan Press, 1975
- [12]Liu J S,Chen R. Sequential Monte Carlo methods of dynamic system[J].Journal of American Statistician, 1998, 83(1):1032-1044.