

Particle Filter Improved by Genetic Algorithm and Particle Swarm Optimization Algorithm

Ming Li

School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China
Email: lim3076@163.com

Bo Pang, Yongfeng He and Fuzhong Nian

School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China
Email: pangbo59021620@163.com, yongfenghyf@126.com

Abstract—Particle filter algorithm is a filtering method which uses Monte Carlo idea within the framework of Bayesian estimation theory. It approximates the probability distribution by using particles and discrete random measure which is consisted of their weights, it updates new discrete random measure recursively according to the algorithm. When the sample is large enough, the discrete random measure approximates the true posteriori probability density function of the state variable. The particle filter algorithm is applicable to any non-linear non-Gaussian system. But the standard particle filter does not consider the current measured value, which will lead to particles with non-zero weights become less after some iterations, this results in particle degradation; re-sampling technique was used to inhibit degradation, but this will reduce the particle diversity, and results in particle impoverishment. To overcome the problems, this paper proposed a new particle filter which introduced genetic algorithm and particle swarm optimization algorithm. The new algorithm is called intelligent particle filter (IPF). Driving particles move to the optimal position by using particle swarm optimization algorithm, thus the numbers of effective particles was increased, the particle diversity was improved, and the particle degradation was inhibited. Replace the re-sampling method in traditional particle filter by using the choice, crossover and mutation operation of the genetic algorithm, avoiding the phenomenon of impoverishment. Simulation results show that the new algorithm improved the estimation accuracy significantly compare with the standard particle filter.

Index Terms—Particle Filter; Particle Swarm Optimization; Genetic algorithm; Particle Degeneracy; Particle Impoverishment

I. INTRODUCTION

Particle filter algorithm [1] is suitable for nonlinear, non-Gaussian system, thus it is concerned by more and more researchers and it has become one of the hot topics in the field of filtering. Because it has the stable, fast, efficient filtering performance, and other filtering algorithm can not compare with it, the particle algorithm has been widely used in target tracking [2], fault diagnosis [3], economic forecasting [4] and other fields.

Particle filter algorithm began in the 1950s and it was called “sequential importance sampling” (ISI). It approximated the probability distribution by using discrete random measure, and it has been applied to the field of physics and engineering. However, ISI algorithm had no much progress because of its computational complexity and degradation. Until 1993, Gordon proposed re-sampling concept to overcome the degradation of the algorithm, the first operational Monte Carlo filter appeared. Modern computing technology enables the Monte Carlo filtering method develops rapidly. Monte Carlo filtering methods were known as the bootstrap, survival of the fittest, condensation algorithm in different fields. Sequential Monte Carlo method is known as the particle filter. In recent years, with the development of modern computing technology and particle itself has great potential, both these make particle filter become a very active research area. A lot of researches have been devoted into the study of the particle filter. People have achieved certain results, and made an important contribution to the improvement of the particle filter. Various improved particle filter methods were proposed.

Particle filter algorithm is a filtering method which uses sequential Monte Carlo within the framework of Bayesian estimation theory. Its essence is to approximate the associated probability distribution using discrete random measure which is consisted of particles and their weights, and updating discrete random measure recursively according to the algorithm. However, the traditional particle filter does not take into account the current measurement, this makes particles sampled from the importance density function are quite different from the particles sampled from the real posterior probability density function. When the variance of importance weights is increasing with time, the weight concentrates on a small number of particles, and the weights of other particles are very small, we can even neglect these particles, so that a large number of calculations are wasted on the particles that do not work on the estimation, thus the particle set can not express the true posterior probability distribution, which is called particle degradation [5]. In order to overcome the particle

degradation, researchers use the re-sampling method, which removes the particles with smaller weights, copying particles with larger weights, this will produce the particle impoverishment [6]. Aiming at the above two problems, researchers have put forward a number of improvements. Unscented particle filter[7] uses unscented Kalman filter to generate the importance density function, because the introduction of the latest measurement value, thereby improving the performance of the particle filter, but also increasing the amount of computation greatly. Regularized particle filter[8] calculate the posterior probability in a continuous way by introducing the nuclear density function and nuclear bandwidth coefficient, but it is only a suboptimal filtering method. Particles swarm optimization particle filter has been proposed in Ref. [9], it drives the particles to the high likelihood region, thus improves particle degradation phenomenon to a certain extent, but it did not overcome the particle impoverishment completely. Genetic algorithm has been introduced into the particle filter [10], it improves the utilization of particles, overcoming the particle impoverishment, but the improvement of particle degradation is not obvious.

This paper proposed a particle filter based on intelligence algorithm. We use PSO algorithm to make the particles find the optimal position, drive particles to high likelihood area, inhibit the particles degradation; and then genetic algorithm is introduced into the particle filter to replace the re-sampling, avoid the phenomenon of particles degeneration, increase particle diversity, improve utilization of particles through selection, crossover and mutation operations, thus inhibiting particle degradation and improving the filtering performance further.

The rest of this paper is organized as follows. We give a brief description of standard particle filter in Section II. Then the new particle filter improved by genetic algorithm and particle swarm optimization algorithm is presented in Section III. In Section V we provide the experimental results. Finally, the conclusion is presented in Section VI.

II. THE STANDARD PARTICLE FILTER ALGORITHM

The particle filter is developed based on the framework of Bayesian theory [11]. Assume that the dynamic time-varying system description as follows:

$$\begin{cases} x_k = f_k(x_{k-1}, v_k) \\ z_k = h_k(x_k, w_k) \end{cases} \quad (1)$$

If the initial probability density function of the known state is $p(x_0|z_0) = p(x_0)$, the state prediction equation is

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \quad (2)$$

State update equation is

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \quad (3)$$

Where $p(z_k|z_{1:k-1}) = \int p(z_k|x_k)p(x_k|z_{1:k-1})dx_k$.

Eq.(2)and Eq.(3)describe the basic idea of the optimal Bayesian estimation. It is a recursive method of obtaining the posterior probability, but the integral in (2) can only obtain the analytical solution in some dynamic system.

The core technology of the particle filter is Monte Carlo method, which is also known as the stochastic simulation method, it can transform the calculus into summation of the limited sample points. If we can draw N independent identical distributed samples $\{x_{0:k}^i; i = 1, \dots, N\}$ from $p(x_{0:k}|z_{1:k})$, then the PDF of the state can approach

$$\hat{p}(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_{0:k} - x_{0:k}^i) \quad (4)$$

Where $\delta(\square) - Dirac\delta$, and the weights meet

$\sum_{i=1}^N w_k^i = 1$, when $N \rightarrow \infty$, $\hat{p}(x_{0:k}|z_{1:k})$ absolutely converges to $p(x_{0:k}|z_{1:k})$.

Particle filter is a recursive Bayesian estimation algorithm which achieves through non-parametric Monte Carlo method. The particle filter algorithm consists of two basic parts, one is sequential importance sampling, and the other is sampling importance re-sampling.

Sequential importance sampling can not sample directly from the PDF of the state usually, the Bayesian importance sampling method is that draw N samples $\{x_{0:k}^i; i = 1, \dots, N\}$ from $q(x_{0:k}/z_{1:k-1})$, which is an easy-to-sample distribution function, the PDF of state approach:

$$\begin{cases} \hat{p}(x_{0:k}|z_{1:k}) = \sum_{i=1}^N \hat{w}_k^i \delta(x_{0:k} - x_{0:k}^i) \\ \hat{w}_k^i = w_k^i / \sum_{i=1}^N w_k^i \end{cases} \quad (5)$$

Where $w_k^i \propto \frac{p(x_{0:k}^i|z_{1:k})}{q(x_{0:k}^i|z_{1:k})}$ is importance weight. For

recursive estimation, we select the importance distribution function as follows:

$$q(x_{0:k}|z_{1:k}) = q(x_k|x_{0:k-1}, z_{1:k})q(x_{0:k-1}|z_{1:k-1}) \quad (6)$$

Draw sample x_k^i from $q(x_k/x_{0:k-1}, z_{1:k})$, and the importance weight is

$$w_k^i \propto w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, z_{1:k})} \quad (7)$$

Then normalize the importance weights, we can obtain a set of weighted samples $\{x_{0:k}^i, \hat{w}_k^i, i = 1, \dots, N\}$, the PDF of the state can calculate according to equation (7), which is known as sequential importance sampling.

Particle degradation is the common problem of SIS algorithm, which is that the value of particle's weight is likely to become one after a number of iterations, and the rest of the weights are likely to zero, which does not work on the estimation of the state. The most effective way to reduce the impact of this phenomenon is to choose the good importance distribution function and use re-sampling method. To reduce the weights, a key step is the choice of the importance of the distribution function, under the conditions of $x_{0:k-1}^i$ and $z_{0:k-1}$, the distribution which make the weights minimum is $q(x_k^i | x_{0:k-1}^i, z_{1:k}) = p(x_k^i | x_{k-1}^i, z_k)$, which is called the optimal importance distribution function, a simple importance distribution function is the prior transition distribution of the state, it is $p(x_k | x_{k-1})$, and which is called Bootstrap particle filter.

Sampling importance re-sampling [12] is another method to solve the degradation of the SIS algorithm. It can eliminate the samples which have low importance weights, while increase the samples which have high importance weights. Replace sampling evenly from set $\{x_{0:k}^i; i = 1, \dots, N\}$ according to $\{w_k^i; i = 1, \dots, N\}$, map the weighted random measure $\{x_{0:k}^i, w_k^i\}$ to the weighted random measure $\{x_{0:k}^i, N^{-1}\}$. A priori probability density function $p(x_k | x_{k-1}^i)$ is as the importance density function, and $w_k^i = w_{k-1}^i p(z_k | x_k^i)$ at this time. The introduction of re-sampling overcomes the particle degradation in the particle filter, which make PF applied in many areas. However, reproducing particles with larger weights, and removing the particles with smaller weights in re-sampling process, which make the diversity of particles decreases, and lead to lower filtering performance and even divergence.

III. PARTICLE FILTER ALGORITHM BASED ON INTELLIGENT ALGORITHMS

A. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is proposed by American social psychologist Kennedy and electrical engineer Eberhart in 1995[13]. Particle swarm optimization algorithm has the characteristics of evolutionary computation and swarm intelligence. Kennedy and Eberhart found that it was a good optimization tool. Similar to the other algorithms, PSO algorithm achieves the optimal solution when searches in the complex space through collaboration and competition between individuals. PSO generates an initial population at first, that is to determine an adjustment value in the feasible solution space. Each particle will move in the solution space, and determine the particle's direction and distance by a speed. The particles will move follow the current optimum particles, and get the optimal solution by

iterative searches. In each generation, the particle will track two extreme values, one is the optimal solution that the particle itself finds so far, and the other is the optimal solution that the whole population finds so far. Kennedy and other researchers adopted the title of "particles" was a compromise consideration. If the quality and volume of the group members is very small, the particles can be called "points".

PSO algorithm is an effective global optimization algorithm, it operates based on the fitness value of the particles, guiding optimization searching by co-operation and competition between the particles. These particles without weights and volume, they fly in N-dimensional space at a certain speed, and their speed adjusts dynamically according to individual flying experience and flying experience of the group. The substance of the PSO algorithm is that guide the next iteration position of the particles by using their information, and the information of their individual extreme and global extreme [14].

PSO algorithm can be expressed as follows: initialize a particle swarm consisted of a set of m particles, the position of the ith particle in the n-dimensional space can be expressed as $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$, the speed can be expressed as $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$. $P_i = (p_{i1}, p_{i2}, \dots, p_{in})$ is the optimal location of the ith particle experienced, while $G_i = (g_1, g_2, \dots, g_n)$ is the optimal position of the particle swarm. When the two extreme values are found, each particle updates its velocity and position according to the following formula:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}[p_{ij}(t) - x_{ij}(t)] + c_2r_{2j}[p_{gj}(t) - x_{ij}(t)] \quad (8)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (9)$$

Where i represents the ith particle, j represents the j dimension of the particles; t denotes the t-generation; w is called the inertia factor, the front speed controls the current speed through inertia weight, larger w can strengthen the global search, while smaller w can strengthen the local search; c_1 makes the particle fly to its own optimal position, c_2 makes the particle fly to the global optimal position; r_1 、 r_2 are independent random function, they have the values between 0 and 1.

B. Genetic Algorithm

The genetic algorithm (GA) is proposed by Holland, a professor in the University of Michigan in the 1970s, it is a simulation of the natural selection and natural genetic mechanisms. It is a new optimization algorithm, is generated on the basis of Darwin's theory of evolution and modern genetics. The survival of the fittest principle is the most important part of Darwin's theory of evolution. It considers that each species adapts to the environment more and more in the development, the basic characteristics of each individual of the species is inherited by future generations, but future generations will produce some new changes different from the parents. When the environment changed, the individual

characteristics that can adapt to the environment will retain. Genetic principles are the most important part of the genetic, it considers that genetic is in cells in the form of password, and is included in the chromosome in the form of genes. Each gene has a special place and controls a particular nature. Therefore, the individuals of each gene have some kind of adaptability of the environment. Gene mutations and genomic hybridization may produce offspring adapted to the environment better. Genetic structures with high adaptability can survive in the process of survival of the fittest. The nature of biological evolution is a process of learning and optimization, it results in genetic algorithm of this intelligent algorithm through the simulation of biological evolutionary process.

The solution to the problem is obtained through gradual evolution in genetic algorithm. The calculation of genetic algorithm starts from a set of solutions, this group of solutions are called population, are expressed as genes in the algorithm. We constitute a new population using the solutions in the population, because we expect the new population is better than the older population. Of course, if the solution of the new population wants to have such a nature, it must select according to its fitness, the higher its fitness is, the more chance it has to construct a new population. This process repeated again and again until it met up the constraints we gave. Genetic algorithm has these properties, such as group search, global optimization, searching not target at a single point, no require auxiliary information, the application range is greatly expanded. Genetic algorithm is not easy to fall into local optimum in the search process, it has intrinsic parallel search mechanism. Genetic algorithm is inherently parallel and has the power of parallel computing, it is scalable, easy to combine with other computing technology. This paper uses genetic algorithm to improve the particle filter based on the above advantages of genetic algorithm.

This paper improved the particle filter by using the selection, crossover and mutation in genetic algorithm. Selection is that select a number of individuals from population according to a certain probability. Generally, the process of selection is that of the survival of the fittest based on the fitness, certain individuals selected are proportional to its fitness. The operation of selection determines the individual's fitness in the population according to the similarity of target and template. The greater similarity of the particles have, the greater weight of the particles have, then they have greater probability of being retained [15]. And then cross the individual left, the crossover process can be described as follows:

$$\tilde{x}_k^m = \alpha x_k^m + (1 - \beta)x_k^n \quad (10)$$

$$\tilde{x}_k^n = \beta x_k^n + (1 - \alpha)x_k^m \quad (11)$$

Where α and β are the weight coefficient,

$$\alpha = \frac{w_k^m}{w_k^m + w_k^n}, \beta = \frac{w_k^n}{w_k^m + w_k^n} . z_k^n \text{ is the fitness of particle n}$$

at time k. Finally, individual will mutate according to some probability: $\tilde{x}_k^j = x_k^j + \eta$. The purpose of the crossover and mutation is to increase the diversity of

individuals in the population, thus avoid the individuals falling into local solution [16].

In order to take into account the speed and accuracy of the algorithm, this paper uses an effective sampling scale,

that is $N_{eff} = 1 / \sum_{i=1}^N (w_k^i)^2$. Set an effective sample size

$N_{threshold}$ as the threshold value according to the actual projects. When $N_{eff} < N_{threshold}$, we improve the genetic algorithm further, so it need not improve the genetic algorithm at each moment, and reduce the complexity of the algorithm, improve the estimation speed.

C. An Optimized Particle Filter Algorithm based on Intelligent Algorithms

We can see there are many similarities between PSO and particle filter, we know this from the introduction of the PSO and the particle filter above. Firstly, PSO finds the optimal value by updating the velocity and position of the particle in the search space continuously. However, the particle filter approximates the real posterior probability distribution of the system by updating the position and weight of the particles. Secondly, the particle with the maximum fitness value represents the optimal value of the search space in PSO algorithm, particle with the maximum weights represents the most likely state of the system. Thirdly, PSO algorithm and the particle filter method have their own movement mechanisms, the particles update their position and velocity by the pursuit of individual optimal values and the global optimum in PSO algorithm, however, each particle in the particle filter algorithm updates their location by using the motion model at first, and then updates its own weight value through measurement model. Therefore, the PSO algorithm can improve the performance of standard particle filter based on the above similarities.

Using genetic algorithms to improve the particle filter is because genetic algorithm has the unique optimization capability, which can improve the usage efficiency of particles, make the particles less which are needed to approach the posterior probability distribution, and avoid re-sampling, it reduces the computation to a certain extent, it can improve the real-time of the algorithm effectively. Furthermore, since the genetic operators can increase the diversity of particles effectively, solve the particle degradation, thus it can improve the accuracy of the algorithm, prevent the phenomenon of filter divergence effectively, and improve the state estimation accuracy.

The specific steps of the improved algorithm proposed in this paper can be expressed as follows:

Step1 we obtain the value of the measurement and define the fitness function. The conventional particle filter uses a suboptimal importance function, therefore, the particle sampling process is sub-optimal, in order to optimize the particle filter sampling process, the latest measured values were introduced into the sampling process, we define the fitness function as:

$$z_k \square fitness = \exp\left[-\frac{1}{2R_k}(z_k - \hat{z}_{k|k-1}^i)^2\right] \quad (12)$$

Where R_k is the measurement noise variance; z_k is the latest measured value; $\hat{z}_{k|k-1}^i$ is the prediction value

Step2 Initialization: $k = 0$, we take N particles from the importance density function, and we use $\{x_{0:k}^i, w_k^i\}_{i=1}^N$ to denote those particles. We define the initial weights of each sample as $\{w_k^i = 1/N, i = 1, 2, \dots, N\}$, the priori probability of the transfer of the importance density function is $x_k^i \square q(x_k^i | x_{k-1}^i, z_k) = p(x_k^i | x_{k-1}^i)$.

Step3 Update the weights: according to the latest measured values, we update the current particle weights:

$$\begin{aligned} w_k^i &= w_{k-1}^i p(z_k | x_{k-1}^i) = w_{k-1}^i \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, z_k)} \\ &= w_{k-1}^i p(z_k | x_k^i) = w_{k-1}^i \exp\left[-\frac{1}{2R_k}(z_k - \hat{z}_{k|k-1}^i)^2\right] \end{aligned} \quad (13)$$

Step 4 we use PSO algorithm and the following formula to update each particle's speed and position, making the particles close to the true state .

$$\begin{aligned} v_{ij}(t+1) &= wv_{ij}(t) + c_1 r_{1j} [p_{ij}(t) - x_{ij}(t)] \\ &+ c_2 r_{2j} [p_{gj}(t) - x_{ij}(t)] \\ x_{ij}(t+1) &= x_{ij}(t) + v_{ij}(t+1) \end{aligned}$$

Let $c_1 = c_2 = 2$. If the particles are located near the true state, then the fitness of each particle in the particle swarm is very high. Conversely, if the individual optimal values of each particle in the particle swarm and the global optimal value of the particle swarm are very low, then the particles are not located in the vicinity of the true state, at this time, the particle set updates each particle's velocity and location according to the optimal value by using PSO algorithm, making the particles move to the true state. The essence of the PSO algorithm is to drive all particles to the high likelihood probability.

Step5 Weights normalization: $\bar{w}_k^i = w_k^i / \sum_{i=1}^N w_k^i$. We

determine whether $N_{eff} = 1 / \sum_{i=1}^N (w_k^i)^2 < N_{threshold}$, if so,

carry out Step 6; if not, skip to Step7.

Step 6 The operations of genetic:

(1) The operation of choice.

See the weighted particles set of k moment as $\{x_k^i, w_k^i\}_{i=1}^{N_k}$, see the w_k^i as the fitness of the corresponding particles in the particle. The variance of particles is calculated fitness importance weights of the variance. When the variance of weights importance of particles is smallest, state estimation is closest to the true state. Thus, we judge operation of choice according to the variance of size based on fitness. If variance of size is less than the average variance of the size, the selection operation retains all particles, refused to take crossover, mutation operation or jump to Step7. If the variance of size is greater than the average squared variance, we skip

selection operation and take the crossover and mutation operation.

(2) The operation of crossover.

Select two particles $(x_k^m, x_k^n)_{m,n=1}^{N_s}$ from the particle set randomly. Take cross-operation according to the following two equations:

$$\tilde{x}_k^m = \alpha x_k^m + (1 - \beta) x_k^n \quad (14)$$

$$\tilde{x}_k^n = \beta x_k^n + (1 - \alpha) x_k^m \quad (15)$$

The crossover guideline is that

if $p(z_k | \tilde{x}_k) > \max\{p(z_k | x_k^m), p(z_k | x_k^n)\}$, we accept the

particle \tilde{x}_k^m ; Otherwise, accept the particles whose probability is $p(z_k | \tilde{x}_k) / \max\{p(z_k | x_k^m), p(z_k | x_k^n)\}$. The

way we accept or give up the particle \tilde{x}_k is same as \tilde{x}_k^m .

(2) The operation of mutation.

Select a particle $(x_k^j)_{j=1}^{N_s}$ from the particle set randomly.

Then take mutation operation according to the following

formula: $\tilde{x}_k^j = x_k^j + \eta$. The variance criteria: if

$p(z_k | \tilde{x}_k) > p(z_k | x_k^j)$, we accept the particle \tilde{x}_k^j ;

Otherwise, accept the particles whose probability

is $p(z_k | \tilde{x}_k) / p(z_k | x_k^j)$.

Through the above method, Particles take crossover and mutation operations in order to get a new particle set $\{\tilde{x}_k, \bar{w}_k\}_{i=1}^{N_s}$.

Step7 State estimates: $\hat{x}_i = \sum_{i=1}^N w_k^i x_k^i$

Step8 We judge whether the moment of k is the last minute of target; and if so, then the algorithm terminates ;if not, let $k = k + 1$, then return to Step2, recursive and estimate the posterior probability of target state of next time.

The flow chart of the algorithm is shown below:

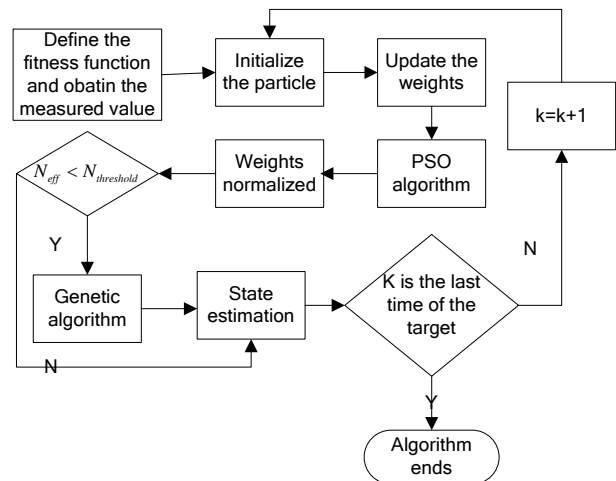


Fig.1. the flow chart of IPF

V. SIMULATION RESULTS AND EXPERIMENTAL ANALYSIS

This paper selects the non-static growth model (UNGM) and uses Matlab software to simulate the model. System process model and state model as follows:

$$x(t) = 0.5x(t-1) + \frac{25x(t-1)}{1+[x(t-1)]^2} \quad (16)$$

$$z(t) = x(t)^2/20 + v(t) \quad (17)$$

Select the measurement noise variance $R = 1$, the two process noise variances are $Q = 10$ and $Q = 20$ respectively. Take the number of particles $N = 100$ and $N = 500$ in PF algorithm, and the number of particles $N = 100$ in IPF algorithm proposed in this paper to simulate. Then we compare the results. In a single experiment, take the root mean square error formula:

$$RMSE = \left[\frac{1}{N} \sum_{i=1}^n (x_n - \hat{x}_n)^2 \right]^{1/2}$$

When the average effective sample size is \bar{N}_{eff} , in the same circumstances, the more effective sample, the higher estimation accuracy. The time step of 50, the times of simulation 50, the algorithm termination condition is the number $n=50$ of iterations. Fig.2 is single experimental simulation of the PF and IPF, Fig.3 is PF and IPF residual plots.

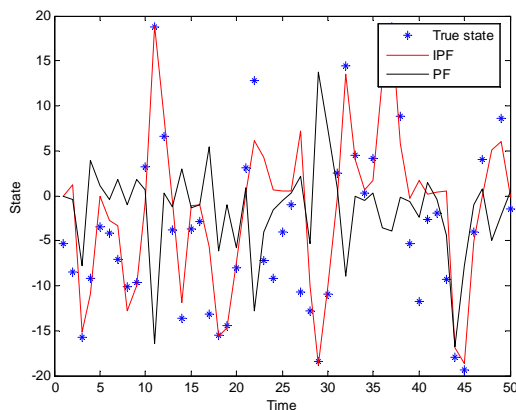


Fig.2. Single experimental simulation of the PF and IPF

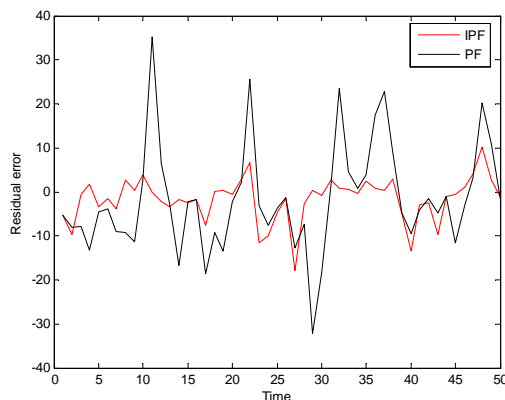


Fig.3. PF and IPF residual plots

The average data of the 50 experiments, the data is as following:

TABLE I
FILTERING PERFORMANCE STATISTICS

The process noise variance is 10 and the measurement noise variance is 1				
Algorithm	Particle number N	The average effective sample	RMSE	Running time
PF	100	12.681	2.7583	0.2415
PF	500	18.179	2.4314	0.5142
IPF	100	39.756	1.0957	0.3019
The process noise variance is 20, and the measurement noise variance is 1				
PF	100	13.428	4.4813	0.2421
PF	500	19.247	4.2851	0.5417
IPF	100	41.275	1.1439	0.3147

Fig.2 is single experiment the simulation diagram of the PF and IPF, it is clear that IPF is more accurate than PF.

Fig.3 is the residual plots of the PF and IPF, the residuals of the true value denotes the size of the estimation errors, from the figure we know that the PF fluctuations is small, while the IPF fluctuation is large, so estimated value of IPF is more close to the true value .

From the experimental data showed in the table I: in the same noise environment, IPF has the most effective average sample. That indicates IPF inhibits particle degradation and IPF is most effective in increasing the diversity of particles. Minimum RMSE values indicate that its estimated the highest accuracy and the time of estimating is also shorter.

When the particle number increases to 500, IPF algorithm is better than PF algorithm, in terms of algorithm estimation accuracy and time of algorithm estimating. In the case of an increase in noise, the RMSE values of the IPF algorithm change the minimum, which indicates IPF anti-noise performance of the best, shows IPF still inhibiting particle degradation in the case of an increase in noise and increasing the particle diversity, and keep the algorithm very accurately and very efficiently.

VI. CONCLUSION

In this paper, the Particle swarm optimization idea and Genetic algorithm were introduced into the particle filter to improve the particle filter. Particle Swarm Optimization drives particles to move to the high likelihood area, increases particle diversity, inhibit the degradation of particle. Use the crossover and mutation operations of genetic algorithm to replace the traditional re-sampling methods, thus avoiding the phenomenon of particle impoverishment, increasing effective particles, improving particle utilization. We set an effective particle threshold in the process of the particle filter which improved by particle swarm optimization and genetic algorithm optimization to ensure the precision of the algorithm. At the same time, the efficiency and real-time of the algorithm were improved. The simulation results show that the performance of particle filter algorithm

based on intelligent algorithms proposed in this article is much better than the conventional particle filter algorithm.

ACKNOWLEDGMENT

This research is supported by Natural Science Foundation of Gansu Province (No.1014RJZA028, No.1112RJZA029) and the Fundamental Research Funds for the Gansu Universities (No.1114ZTC144).

REFERENCES

- [1] Djuric, Dotecha. Particle Filters [J]. IEEE Signal Processing Magazine, 2003, (10): 19-38.
- [2] Doucet A, Godsill S, Andrieu C.A survey of convergence results on particle filtering methods for practitioners. IEEE Trans.on Signal Processing, 2002, 50(2):736-746.
- [3] Sankaranarayanan, Srivastava, Chellappa. Algorithmic and Architectural Optimizations for Computationally Efficient Particle Filtering [J]. IEEE Trans Image Processing, 2008, 17 (5): 737-748.
- [4] Yanhua Ruan, Willett, P, Marrs, A, Palmieri, F, Marano, S. Practical Fusion of Quantized Measurements Via Particle Filtering [J]. IEEE Trans Aerospace and Electronic Systems, 2008, 44 (1): 15 - 29.
- [5] Armin Burchardt, Tim Laue. Optimizing Particle Filter Parameters for Self-localization [J].Computer Science, 2011,65(56):145-156.
- [6] Andreas S. Stordal, Hans A. Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter[J]. Mathematics and Statistics, 2011,15(2):293-305.
- [7] Hai-dong Hu, Xian-lin Huang, Ming-ming Li. Federated Unscented Particle Filtering Algorithm for SINS / CNS / GPS system [J]. Journal of Central South University of Technology, 2010, 17 (4): 778-785.
- [8] Giremus A, Tourneret J. An improved regularized particle filter for GPS / INS integration. Signal processing Advances in Wireless Communication, 2005 IEEE 6th Workshop on. Page 1013-1017.
- [9] FANG Zheng, TONG Guo-feng, XU Xin-he. Particle swarm optimized particle filter. Control and Decision, 2007,22(03):273-277.
- [10] Colin R.Reeves. Genetic Algorithms. International Series in Operations Research & Management Science, 2010, Volume 146, 109-139.
- [11] Mingyang Li, Mourikis, A.I.,Shelton,C.R. A particle filter for monocular vision-aided odometry. IEEE International Conference on Robotics and Automation, 2011:5663-5669.
- [12] Seongkeun Park. A New Evolutionary Particle Filter for the Prevention of Sample Impoverishment.IEEE Transactions on Evolutionary Computation, 2009, 13(4):801-809.
- [13] Kennedy J, Eberhart R. Particle swarm optimization [C]. Proc of the IEEE Int Conf on Neural Networks, Piscataway: IEEE Service Center, 1995: 1941-1948.
- [14] Jong-Bae Park. An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems. IEEE Transactions on Power Systems, 2010,25(1):156-166.
- [15] Thiago F.Noronha, Mauricio G.C.Resende and Celso C.Ribeiro. A biased random-key genetic algorithm for routing and wavelength [J]. Business and economics, 2011,50(3):503-518.
- [16] Wei-Chang Yeh, Mei-chi Chuang. Using Multi-objective genetic algorithm for partner selection in green supply chain problems [J].Expert Systems with Applications, 2011, 38(4):4244-4253.



MingLi was born in Lanzhou, Gansu Province, China in 1959. He received the bachelor degree in mathematics from the Xian University of Technology, Shanxi, China, in 1982. His research interests are intelligent Information Processing, signal processing, face analysis and object tracking.

Currently, he is the President of Department of Computer at Lanzhou University of Technology. In recent 5 years, He has authored about 30 papers in international journals, National issues, and international conference proceedings. In 2007, he published Principles of Database and Its Application [M]. Chengdu: Southwest Jiaotong University Press. In 2010, he published Intelligence Information Processing and Its Application [M]. Beijing: Publishing House of Electronics Industry.

Prof. Li has served as a reviewer for Journal of Lanzhou University of Technology

Bo Pang was born in Jixi, Haerbin Province, China in 1985. He received the bachelor degree in automation Engineering at Haerbin University of Science and Technology in 2008. He is getting his Master's in communication and information system at Lanzhou University of Technology. Since July 2010, he has taken up the subject of Machine Vision Group. His research interests include intelligent information processing, object tracking, signal processing, human face recognition and person identification. He has just published papers in journals and conferences.

Yongfeng He was born in Handan, Hebei Province, China in 1986. She received the bachelor degree in computer science and technology at Tangshan Teachers' College in 2010. She is getting her Master's in computer application technology at Lanzhou University of Technology. Since July 2011, she has taken up the subject of Machine Vision Group. Her research interests include intelligent information processing, object tracking, signal processing, human face recognition and person identification. She has just published papers in journals.