Research on Particle Filter Algorithm Based on Neural Network

Ershen Wang
School of Electronic and Information Engineering
Shenyang Aerospace University
Shenyang,China
e-mail:wanges_2016@126.com

Tao Pang
School of Electronic and Information Engineering
Shenyang Aerospace University
Shenyang,China

Xingkai Li
School of Electronic and Information Engineering
Shenyang Aerospace University
Shenyang,China

Abstract—Aiming at the weight degeneracy phenomena in particle filter algorithm, the improved particle filtering algorithm based on neural network was presented. BP neural network and generalized regression neural network (GRNN) are adapted to improve resampling algorithm and importance probability density function. This algorithm utilizes the nonlinear mapping function of BP neural network. First of all, to sample from the importance density function of particle weight division, the weighted particle is splitted into two small weight particles. Then, abandon the weight of very small particles, and adjust the particles with smaller weight by using the neural network. This algorithm optimizes the sample from importance density function by generalized regression neural network. Through GRNN, the samples are adjusted. The samples are more nearer to the posterior probability density. Simulation results show that the particle filter algorithm based on neural network can improve the diversity of particle samples, increase the effective particle number, reduce the mean square error, and the filtering performance is improved. It is proved that this particle filter algorithm based on neural network is available and effective.

Keywords- Particle filter; Particle degeneracy ; Resampling ;BP neural network ; Generalized regression neural network (GRNN)

I. INTRODUCTION

A PF is an algorithm that provides iterative Monte Carlo approximations for a given sequence of state variable distribution. It approximates the required probability density function (PDF) by swarms of points in the state space. The basic principle of particle filter algorithm is that: First, based on the priori conditional distribution of system state vector the state space generate a group of random samples, these samples called particles; then based on the measurement value constantly adjust the particle weight and the particle distribution position, modified initial priori conditional distribution. The algorithm is a recursive filtering algorithm, commonly used to handle non-Gaussian and nonlinear systems state and parameter estimation. But there are particles degradation problems in the particle filter. And now it is widely used in navigation, automatic control, target tracking [1-4], and so on, and successfully applied to the dynamic system failure detection problems[5-6]. The basic particle filtering methods, such as sequential importance sampling method has a problem of particle degradation, as a result, solving this problem mainly relies on two key factors: choosing the importance density function and resampling strategy. Although the resampling strategy can solve the problem of particle degradation, at the same time will bring the sample depletion problem, that is the power value of particle is selected for many times, and it will contain many repeated sampling results, thus diversity of particles will be lost[7]. To solve this problem, the neural network is used to improve the performance of particle filter algorithm. Currently, the neural network algorithm is adapted to improve the performance of particle filter, such as neural network weight adjusting particle filter(NNWA-PF) and generalized regression neural network particle filter(GRNN-PF). These two algorithms can effectively increase the number of samples, reduce degradation and improve filtering the performance of particle filter[8].

Firstly, this paper introduces the principle of particle filter, and then introduces the principle of the neural network. And the combination of the particle filter and the neural network is discussed. Compared with the basic particle filter and improved particle filter based on neural network is analyzed. Finally, the numerical simulations are verified.

II. PARTICLE FILTER ALGORITHM BASED ON BP NEURAL NETWORK

BP neural network is a feed forward networks, which is the most widely used one of neural network models. BP neural network can learn and store a large amount of input -output model mapping relationships. Its learning rule is to use the steepest descent method, through the back-propagation to continuously adjust the weights and thresholds of the network, so that the minimum squared

error of the neural network can be obtained. BP neural network topology includes the input layer, hide layer and output layer[9].

Assuming the number of input samples p, that is $x_1, x_2, \cdots x_p$,and the corresponding teacher signal samples is $d_1, d_2, \cdots d_p$. The actual output is $y_1, y_2, \cdots y_p$ .The output error is as follows.

$$E = \frac{1}{2}\sum_{k=1}^{p}(d_k - y_k)^2 \quad (1)$$

By using the error to adjust the weights among the various layers, and using the gradient method to adjust weights in order to reduce the total error.

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} \quad (2)$$

Where η is a learning step. w_{jk} is the weight between the layers. Δw_{jk} is modified value of the amount of the weights. Using the total error to modify accordingly to the weights, which is namely the BP algorithm.

With regard to the problem of the diversity loss of the particle filter, which uses the resampling method. The particle filter algorithm and BP neural network will be combined to increase the weight of the particles located in the tail of the probability distribution. At the same time, a higher particle weights will be splitted two small particle weight. Thus, the diversity of the sample particles will be improved, and the filter performance will be improved.

The ideas for the algorithm are to add two steps on the basic particle filter algorithm. That is weight division and weight adjustment. The particles with very large weights are splitted into two smaller particles, whose weight is halved of the original particles. In order to maintain the same total number of particles, some of particles with very small weights are discarded. The particles having a small weight value is regarded as inputs to the neural network. And the state of the particles is regarded as the input value of the neural network.

The particle filter algorithm based on BP neural network can be described by the following steps:

A. Initialized

According to the priori probability $p(x_0)$, the initial particles $\{x_0^i\}_{i=1}^{N_s}$ are obtained, the weight value of the particles is $1/N_S$ 。

B. For k=1,2,…performing the following steps：

1) 1) State prediction

Priori particles extracted based on the state equation of system at time k.

$$\{\mathbf{x}_{k|k-1}(i) : i = 1, 2, \cdots, N\} \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$$

2) Update

Update particle weights at time k：

$$w_k^i = w_{k-1}^i p(z_k \mid x_k^i) = w_{k-1}^i p_{e_k}(z_k - h(x_k^i))$$

where， i=1， 2， 3... N_s .The weight values are normalized：

$$\tilde{w}_k^i = w_k^i \Big/ \sum_{i=1}^{N_s} w_k^i, \sum_{i=1}^{N_s} \tilde{w}_k^i = 1$$

C. Particle splitting

Sorting the particle size of the matrix by weight into high and low weight matrix . Spliting the high weight matrix q large weight particles into two smaller weights halved particles, while the low weight matrix q weight smallest particle discarded.

D. Weight adjustment

Sorting the particle weight matrix of step(3) in descending order. Seeking q smallest particles, using BP neural network to adjust the weights of the particles.

E. Resampling

From the set of particles（$\mathbf{X}^i_{k|k-1}, \tilde{w}_k^i$）, according to the values of the importance resampling , getting a new set of particles（$\hat{\mathbf{X}}^i_{k|k-1}, i = 1, \cdots, N$）.

F. Estimate

Calculating the current time system state estimation value：

$$\hat{\mathbf{x}}_k \approx \sum_{i=1}^{N_s} \tilde{w}_k^i \hat{\mathbf{X}}^i_{k|k-1}$$

Then, k=k+1 , go to steps (2).

III. PARTICLE FILTER ALGORITHM BASED ON GENERALIZED REGRESSION NEURAL NETWORK

Generalized regression neural network (GRNN) is a new neural network and it is the branch of RBF neural network. GRNN doesn't need to ascertain the prior equation, which uses probability density function instead of the forms of inherent equation. Through the implementation of Parzen non-parametric estimation, the joint probability density function from the independent and dependent variables in the observed sample is obtained. Then, GRNN can directly calculate out the return value of the independent variable caused by the dependent variable. There is no need for GRNN to set model forms. However, it needs to optimize the value because of the effective smooth factor σ in kernel implicit return unit. GRNN uses the same smooth factor for the hidden layers of the kernel function, and the network training process is essentially a one-dimensional network optimization process [10].

Supposing the joint probability density function of the random variables x and the random variables y is $f(x, y)$, and the return value from y to x_0 is as follows.

$$E(y|x_0)=\hat{y}(x_0)=\frac{\int_{-\infty}^{\infty} yf(x_0,y)dy}{\int_{-\infty}^{\infty} f(x_0,y)dy} \quad (3)$$

Applying Parzen nonparametric estimation, and the probability function $f(x_0,y)$ can be estimated by the sample data $\{x_i,y_i\}_{i=1}^{m}$.

$$f(x_0,y)=\frac{1}{n(2\pi)^{\frac{p+1}{2}}\sigma^{p+1}}\sum_{i=1}^{n}e^{-d(x_0,x_i)}e^{-d(y,y_i)} \quad (4)$$

$$d(x_0,x_i)=\sum_{j=1}^{p}[(x_{0j}-x_{ij})/\sigma]^2, d(y,y_i)=(y-y_i)^2 \quad (5)$$

The n represents sample capacity, p is the dimension of x, and σ is the width coefficient of Gauss function.The results is can be described as follows.

$$\hat{y}(x_0)=\frac{\sum_{i=1}^{n}y_i e^{-d(x_0,x_i)}}{\sum_{i=1}^{n}e^{-d(x_0,x_i)}} \quad (6)$$

The predicted value $\hat{y}(x_0)$ is the weighted sum of dependent variable values y_i for all the training samples . As the value of the smooth factor σ is large, the tendency of $d(x_0,x_i)$ is to 0, leading to the predicted value similar to the average of dependent variables of all samples. To the contrary, as the tendency of smooth factor is 0, that the predictive value is very close to the training sample, which may cause over-fitting. When σ has the proper vaules, seeking to the predictive value, the dependent variables for all the training samples are considered, causing that the dependent variables corresponding to samples closed to predict points own greater weight.

In the particle filter, according to the measured value z_k, the GRNN network can be trained to approach the likelihood function, and then use the network to adjust the sample values. The improved particle filter algorithm based on GRNN can be described.

1）First, creating an input vector and target vector for training. Obtained by sampling a set of samples, samples and their measurements predictive value constitute the input vector and target vector. The dimension of the input vector n and the learning samples m determines the structure of the network n×m×(n+1)×n (in this study, n = 1, m = 99).

2）After the network training, adjust samples of the particle algorithm in the form of input vector. Creating an n-dimensional vector can be as follows.

$$X_k^i=[x_k^i, x_k^i\pm j\Delta],\ j\Delta<L(j=1,2,\cdots n/2)$$

The L presents the adjustment range, then considering the trained nerual network as input vector. Finally, with the instruction of the output vector of the neural network, the samples x_k^i is replaced by the optimal points $x_k^i\pm j\Delta$. The adjusted samples are closer to the importance function to achieve the target of improving filter performance.

The improved particle filter algorithm based on GRNN are shown as follows.

①The initialization, k=0;

Creating the random samples $X_0^{(1)}, X_0^{(2)},\cdots X_0^{(N)}$ according to the prior probability density $p(X_0)$ (N is the number of the random samples)

②When k=1,2,..., then

A. State prediction

Creat the priori paiticle at the time of k according to the state equation of system

$$\{X_{k|k-1}(i); i=1,2,\cdots,N\}\sim p(X_k\mid X_{k-1})$$

B. Adjusting the samples

Adjusting the samples by using GRNN nerual network. Define the input of network as x_k^i, i=1,2…,N, and the target vector is z_k ,then begin to train the network.

$$\hat{Y}(X)=\frac{\sum_{i=1}^{n}Y_i e^{-\frac{[(z_k-x_k^i)^T(z_k-x_k^i)]}{2\sigma^2}}}{\sum_{i=1}^{n}e^{-\frac{[(z_k-x_k^i)^T(z_k-x_k^i)]}{2\sigma^2}}} \quad (7)$$

Constructing an n-dimensional vector $X_k^i=[x_k^i, x_k^i\pm j\Delta],\ j\Delta<L(j=1,2,\cdots n/2)$, L represents the adjustment range, and then regard the vector as the trained input vector of neural network.

$$Y'(X)=\frac{\sum_{i=1}^{n}\hat{Y}e^{-\frac{[(z_k-X_k^i)^T(z_k-X_k^i)]}{2\sigma^2}}}{\sum_{i=1}^{n}e^{-\frac{[(z_k-X_k^i)^T(z_k-X_k^i)]}{2\sigma^2}}} \quad (8)$$

Finally, select the minimum error points as optimum points through the instruction of the output of the nerual network, so the samples x_k^i are substituted by optimum points $x_k^i\pm j\Delta$.

Calculating the weight $\omega_k^{(j)}$ of particles.

$$\omega_k^{(i)}=\omega_{k-1}^{(i)}p(Z_k\mid X_k^{(i)}), i=1,\cdots N$$

The normalized weights are:

$$\tilde{\omega}_k^{(i)}=\frac{\omega_k^{(i)}}{\sum_{i=1}^{N}\omega_k^{(i)}},\ \sum_{i=1}^{N}\tilde{\omega}_k^{(i)}=1$$

C. Resampling

Compare with the threshold N_{thres} of the effective particle number after calculating the effective number $\hat{N}_{eff}$ of particles . If $\hat{N}_{eff} < N_{thres}$, then resample the particles. Randomly setting a threshold, when the accumulated value exceeds this threshold with the cumulative weight of particles,then retain the particles. Finally mapping the former weighted sample $\{x_{0:k}^{(i)}, w_k^{(i)}\}_{i=1}^{N}$ as the equal-weight samples $\{x_{0:k}^{(i)}, N^{-1}\}_{i=1}^{N}$. Otherwise, perform the following steps.

D. Estimate

Calculate the current systematic state estimation:

$$\hat{X}_k = \sum_{i=1}^{N} X_{k|k-1}^{(i)} \tilde{\omega}_k^{(i)}$$

E. $k = k+1$, *continue to calculate next value.*

IV. SIMULATION RESULTS AND ANALYSIS

A. Simulation model

To further verify the effectiveness of the algorithm, the paper cited the following model [11], which is one of the typical models to validate the performance of particle filter algorithm. State model and observation model can be described as follows.

$$x_{k+1} = 0.5x + \frac{25x}{1+x^2} + 8\cos 1.2(k-1) + w_k$$

$$y_k = \frac{x_k^{\ 2}}{20} + v_k$$

In the formulas, k represents a state in time, and x_k indicates state quantity, y_k represents a concept of measurement. The initial value of the state $x_0 = 0.1$, And w_k and v_k are the process noise and the measurement noise respectively. They obey a Gaussian distribution with zero mean and variance of the $\sigma^2_{\ m}$ and $\sigma^2_{\ n}$ .State variable x_k is a one-dimensional variables, which obeys Gaussian distribution.

B. Simulation Result and Analysis

The system is based on the above model equation and measurement equation. The number of the particle is 100 and the initial value of the particle x_0 .Particle filter algorithm iteration step t_f is100.Among,the initial value of x is 0.1. The figures show the filtering performance results.

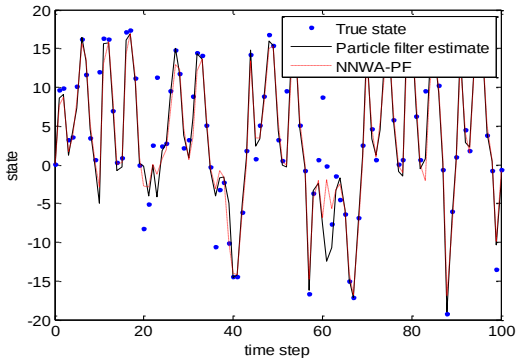

Figure 1. Comparison of the NNWA-PF and PF algorithms

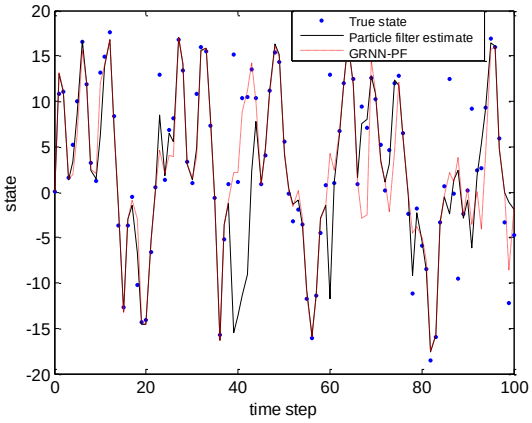

Figure 2. Comparison of the GRNN-PF and PF algorithms

In Fig.1 and Fig.2, the horizontal axis represents the number of iterations, and the longitudinal axis respectively represents the true value and the estimated value of the two algorithms before and after being improved. From the figures it can be seen the performance of improved particle filter based on generalized regression neural network (GRNN) is superior to the performance of basic particle filter. The different parameters processed by the two algorithms shows in Table 1.

TABLE I. PARAMETER COMPARISON OF PARTICLE FILTER ALGORITHMS

Algorithm	Particle number	Effective particle number	RMSE
PF	100	32.1359	5.0273
NNWA-PF	100	38.1209	3.675
GRNN-PF	100	37.2775	3.6947

V. CONCLUSIONS

BP neural network and generalized regression neural network were used to combine with basic particle filter to suppress the weight degeneracy phenomena in basic particle filter algorithm. The improved particle filter algorithms based on BP neural network and generalized regression neural network were discussed deeply. The simulation result of this research shows that this improved particle filter algorithm based on neural network can

improve the diversity of particle samples, increase the effective particle number, reduce the mean square error, and the filtering performance is improved. It is proved that this improved particle filter algorithm based on neural network is available and effective under non-linear and non-gaussian measurement noise environment.

ACKNOWLEDGMENT

The work are supported by the National Natural Science Foundation of China(No.61101161), the Aeronautical Science Foundation of China(No.2011ZC54010), the Natural Science Joint Foundation of Liaoning Province(No.2013024003).

REFERENCES

[1] Gustafsson F, Fredrik G. Particle Filters for Positioning, Navigation, and Tracking[J].IEEE Transactions on Signal Processing,2002, 50(2): 425-437.

[2] Qing Ming, Jo Kang-hyun.A novel particle filter implementation for a multiple-vehicle detection and tracking system using tail light segmentation[J].International Journal of Control, Automation and Systems, 2013,v11, n3, p577-585.

[3] CHEN Zhi-min,BO Yu-ming,WU Pan-long,et al.Novel particle filter algorithm based on adaptive particle swarm optimization and its application to radar target tracking[J].Control and Decision, 2013,28(2)：193-200.

[4] DE FREITAS J F G, NIRANJAN M, GEE A H, etal. Sequential Monte Carlo methods to train neural network models [J]. Neural Computation, 2000, 12(4): 955-993.

[5] XIA Nan,QIU Tian-shuang,LI Jing-chun,et al. A nonlinear filtering algorithm combining the kalman filter and the particle filter[J]. Acta Electronica Sinica, 2013,41(1):148-152. (in Chinese)

[6] LI P , KADIRKAMANATHAN V. Particle filtering based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems[J]. IEEE Trans. Syst., Man,Cybern. C, 2001,31(3):337-343.

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

[8] ZHU Zhi-yu.Particle filter algorithm and application[M].Beijing:Science Press, 2010:102-109. (in Chinese)

[9] CHEN Yangping,WANG Laixiong,HUANG Shitan.Neural network learning algorithm based on particle filter[J].Engineering Journal of Wuhan University, 2006,39(6):86-88. (in Chinese)

[10] Cheng Hongbing,Huang Guorong,Ni Shihong,et al.Desighn of self-organizing fuzzy neural network based on particle filter[J]. Chinese Journal of Scientific Instrument,2011,32(3):634-639. (in Chinese)

[1] FENG Chi, ZHAO Na. Research on MCMC particle filter algorithm[J].Applied Science and Technology, 2009,36(4):19-22.