Hybrid Differential Evolution Particle Filter for Nonlinear Filtering

Chaozhu Zhang and Lin Li

College of Information and Communication Engineering
Harbin Engineering University, Harbin, Heilongjiang 150001, China
zhangchaozhu@hrbeu.edu.cn, lilin0214@yeah.net

Abstract **─** In this paper we propose a novel method for solving the nonlinear problem of the radar target tracking. The algorithm consists of a Particle Filter (PF) which employs the Unscented Kalman Filter (UKF) to generate the importance proposal distribution, and adopts the Hybrid Differential Evolution (HDE) algorithm based on Simulated Annealing (SA) algorithm as the resampling scheme. Firstly, the Importance Distribution (ID) which contains the newest measurements is constructed by the UKF. In addition, the UKF generates proposal distributions that match the true posterior more closely. Secondly, to solve the particle degeneracy and impoverishment phenomenon, the sampling particles are resampled by the HDE algorithm. The mutation and crossover steps of the Differential Evolution (DE) algorithm are executed to generate the trial vectors. Then the selection step is replaced by the Metropolis criterion of the SA algorithm. The proposed algorithm combines the advantages of the SA algorithm with the DE algorithm. It not only has superior estimation performance, but also the convergence speed is fast. Simulation results demonstrate that the proposed algorithm outperforms the standard PF, the Auxiliary Particle Filter (APF), the Regularized Particle Filter (RPF) and the Particle Filter based on Differential Evolution (PFDE).

Index Terms **─** Hybrid differential evolution, nonlinear filtering, particle filter, radar tracking, simulated annealing algorithm.

I. INTRODUCTION

As we know, most of the radar tracking filtering algorithms are linear filtering, such as Kalman filter. However, these linear filters are optimal only under the condition that the system is linear or the noise is Gaussian. Actually, the system model and the measurement equation of the target are nonlinear, and the noise is non-Gaussian. In that situation, if we still use linear filter to track targets, the tracking performance will be reduced, or even worse, the targets will be lost. Therefore, it is necessary to employ the nonlinear filtering to solve the nonlinear problem of the radar tracking.

Nonlinear filtering is a very active topic in signal processing and control theory. There is a vast literature on this subject; see [1-5] for excellent references among others. Although the equations of the optimal nonlinear filter have been developed since the middle of the 1960s, the involved integrals are still intractable. Hence, many suboptimal nonlinear filters have been proposed.

The simplest way to solve the problem of non-Gaussian, nonlinear filtering is the Extended Kalman Filter (EKF) [6]. It linearizes the state transition and the measurement equations through Taylor series expansions. However, the series approximations in the EKF algorithm could cause large errors of the nonlinear functions and probability distributions. So, this filter would result in divergency. The convergence of the EKF is studied in [7]. Later on, the UKF is proposed by Julier and Uhlmann, which uses several so-called sigma points to recursively calculate the mean and covariance used in the Kalman Filter [8]. The UKF could obtain more accurate results than the EKF, but it couldn't adapt to general non-Gaussian distributions. Essentially, the EKF and the UKF have the same principle; both of them use the Gaussian distribution to approximate the true posterior distribution.

Submitted On: February 9, 2014
Accepted On: July 25, 2014

A different approach to the nonlinear filtering problem is the Bayesian approximation, which is detailedly described in [9]. This kind of filter is based on the principle of constructing the posterior probability density of the state by the prior knowledge and the observation; and, the key point is to obtain a good approximation of the posterior density.

Another popular method for nonlinear filtering is Monte Carlo methods, also known as Particle Filter (PF). Up to now, the PF is the most successful nonlinear filter, which goes back to the 1950s, but it hadn't been used in practical applications until 1993 when Gorden proposed the Sequential Importance Resampling (SIR) algorithm [10]. The PF utilizes some random particles with associated weights to approximate the true posterior density function. The PF has been used successfully in many domains; however, its performance depends heavily on the choice of the importance distribution function and the resampling algorithms. To improve the performance of the PF, choosing a good proposal distribution or modifying the resampling scheme are often adopted. For example, Pitt and Shephard [11] introduced the Auxiliary Particle Filter (APF), which uses an auxiliary variable to select the particles. In [12], the Regularized Particle Filter (RPF) is put forward, which resamples from a continuous approximation of the posterior density to reduce the particle impoverishment problem. In [13], the EKF Gaussian approximation is used as the importance distribution for a PF. In [14], the EKF proposal is replaced by UKF proposal, and the Unscented Particle Filter (UPF) is proposed. We will propose a new method in this paper which uses the HDE based on SA algorithm as resampling schemes for the PF, which regards the resampling process as an optimization problem. We refer to it as Hybrid Differential Evolution Particle Filter (HDEPF). In the HDEPF, the importance distribution is generated by the UKF.

The remainder of this paper is organized as follows. At first, the problem statement and the principle of the basic PF are introduced in Section II. In Section III, we proposed the HDE algorithm and HDEPF. Then in Section IV, we discussed some experimental results. At last, conclusions and pointers for future research are presented in Section V.

II. PROBLEM STATEMENT AND THE BASIC PARTICLE FILTER

Many nonlinear filtering problems can be written in the form of the Dynamic State Space (DSS) model as follows:

$$x_k = f(x_{k-1}, u_{k-1}), \tag{1}$$

$$z_k = h(x_k, v_k), \tag{2}$$

where x_k and z_k are the state variable and observation at time k, respectively. k is the time index. $f(\cdot)$ and $h(\cdot)$ are some known functions, system noise u_{k-1} and observation noise v_k are random variables at time of given distributions. u_{k-1} and v_k are independent of past and current states. v_k is independent of system noise u_{k-1}. The objective of filtering is to recursively estimate the posterior density $p(x_k \mid z_{1:k})$ of the state x_k based on all available measurements $z_{1:k} = \{z_1, z_2 \cdots, z_k\}$. A recursive update of the posterior density as new observations arrive is given by the recursive Bayesian filter defined by:

$$p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{1:k-1}) dx, \tag{3}$$

$$p(x_k \mid z_{1:k}) = \frac{p(z_k \mid x_k) p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})}, \tag{4}$$

where the conditional density $p(z_k \mid z_{1:k-1})$ can be calculated by:

$$p(z_k \mid z_{1:k-1}) = \int p(z_k \mid x_k) p(x_k \mid z_{1:k-1}) dx_k. \tag{5}$$

It can be seen that the integrals are intractable. So, the PF uses Monte Carlo methods to translate the integrals problems into the cumulative process of limited particles probability transition. The PF uses the transition density $p(x_k \mid x_{k-1})$ as the importance distribution function $q(x_k^i \mid x_{k-1}^i, z_k)$ to generate particles. Then the posterior density $p(x_k \mid z_{1:k})$ at time k can be described as:

$$p(x_{0:k} \mid z_{1:k}) \propto p(z_k \mid x_k) p(x_k \mid x_{k-1}) p(x_{0:k-1} \mid z_{1:k-1}). \tag{6}$$

Accordingly, the weights of the particles are called importance weight. We define the unnormalized weights as:

$$\begin{aligned} w_k^i &= \frac{p(x_{0:k}^i \mid z_{1:k})}{q(x_{0:k}^i \mid z_{1:k})} \\ &= w_{k-1}^i \frac{p(z_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)}. \end{aligned} \tag{7}$$

Then we normalize the weights and begin the resampling process. The aim of resampling is to eliminate samples with low importance weights and copy samples with high importance weights. After the resampling process, the weights can be defined by $w_k^i = 1/N$. The popular resampling algorithm is the multinomial resampling.

Last the posterior density $p(x_k \mid z_{1:k})$ can be calculated by:

$$p(x_k \mid z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i), \quad (8)$$

where $\delta(\cdot)$ is the Dirac delta function, x_k^i is the i th particle with the normalized weight w_k^i. N is the particle number.

III. HYBRID DIFFERENTIAL EVOLUTION PARTICLE FILTER

A. Hybrid differential evolution algorithm

DE algorithm, proposed by Price and Storn [15], is a population-based stochastic algorithm for global optimization, which has earned a reputation as a very effective global optimizer. DE algorithm has the following advantages over the traditional genetic algorithm: more efficient memory utilization, lower computational complexity, and it is much more easy to use. However, DE algorithm has insurmountable shortcomings. It has slower convergence rate in latter periods, even failing to local extremes [16]. Then Hybrid Differential Evolution Algorithm based on SA algorithm is proposed by [17]. The new algorithm utilizes the search capability of the SA algorithm to enhance the convergence capability of the DE in latter periods and improve the robustness of the DE algorithm. The HDE algorithm uses the Metropolis criterion of the SA algorithm to replace the section step of the DE algorithm. So it relies on the initial population generation, mutation, recombination and the new selection to probe search space through iterative progress until the terminate criteria are met.

Detailed steps are presented accordingly in the subsequent sections.

Step 1: Creating initial population

The first step of HDE is to create the initial population samples (the number of generations is $g = 0$) in n dimension space as follows:

$$x_{ij}(0) = x_{ij}^L + rand_{ij}(0,1)(x_{ij}^U - x_{ij}^L), \quad (9)$$

where $i = 1, 2, \ldots NP; j = 1, 2, \ldots n,$ NP is the population size. x_{ij}^U and x_{ij}^L denote the upper and lower limit of the j th variable in the population, respectively. $rand_{ij}(0,1)$ represents a uniformly distributed random value within $[0,1]$.

Step 2: Mutation operation

The function of mutation in HDE is to maintain the diversity of population. A typical HDE mutation samples formulation is:

$$h_{ij}(g) = x_{ij}(g) + F \cdot (x_{r1j}(g) - x_{r2j}(g)), \quad (10)$$

where g represents the g th generation, $h_{ij}(g)$ are the mutated vector samples. $r1 \neq r2 \neq i$, and $r1$, $r2$ are randomly selected integers within NP, $r1, r2 \in \{1, 2, \ldots, NP\}$. F is scaling factor.

Step 3: Crossover operation

The basic crossover process is a discrete recombination, which employs a crossover constant $CR \in [0,1]$ to determine whether the new generated individual samples need to be recombined. The expression of the crossover process is given in (11):

$$v_{ij}(g) = \begin{cases} h_{ij}(g) & rand(0,1) \leq CR \\ x_{ij}(g) & rand(0,1) > CR \end{cases}, \quad (11)$$

where $v_{ij}(g)$ are the trial vector samples.

Step 4: New selection operation

The HDE algorithm adopts the Metropolis criterion of the SA algorithm to select the trial vector samples.

Step 5: Cool-down operation

In this step, the cool-down operation of the SA algorithm is executed. We define $T = T \cdot \rho$, $\rho \in (0,1)$ as an annealing parameter.

When the new population is propagated, Step 2 to Step 5 is repeated until the pre-specified temperature $T0$ is reached.

B. Hybrid differential evolution particle filter

In the presented algorithm, the particles from using a UKF for importance distribution are regarded as the initial population of the HDE algorithm, and the corresponding weights are treated as the fitness functions of the target vectors, respectively. The HDE resampling scheme recombines the particles by using an iterative process of mutation, crossover and the simulated annealing operator. Then a new set of diverse particles are propagated.

The new filter proposed in this paper is called HDEPF, and the steps follows.

In the first step, we can obtain Gaussian importance distribution with the mean $\tilde{x}_k^i$ and the variance $\tilde{P}_k^i$ by using UKF filter for all particles $x_k^i (i=1,2,\ldots N)$, where N is the particle number. Then the sampling particles can be gained by sampling the importance distribution, $\hat{x}_k^i \sim N(\tilde{x}_k^i, \tilde{P}_k^i)$, and their weights w_k^i can be calculated and normalized by equation (7).

In the second step, the sampling particles $\hat{x}_k^i$ are regarded as the initial population of the HDE algorithm, and the corresponding weights w_k^i are regarded as the fitness functions $f(x_{ij})$. Then a mutation step and a crossover step are executed. Finally, the trial vector samples $v_{ij}(g)$ can be obtain by equation (12), and the corresponding fitness function is calculated as $f(v_{ij}(g)) = \dfrac{p(z_k \mid v_{ij}(g))p(v_{ij}(g) \mid x_{k-1}^i)}{q(v_{ij}(g) \mid x_{k-1}^i, z_k)}$ and is normalized as $f(v_{ij}(g)) = f(v_{ij}(g)) / \sum_i^{NP} v_{ij}(g)$, where NP is population size and $NP = N$.

In the third step, the new selection operator is proceeding. We construct the following parameter about fitness function values, $\Delta f = f(v_{ij}(g)) - f(x_{ij}(g))$, then we decide the trial vector samples by the Metropolis criterion.

In the fourth step, the cool-down operation of the HDE algorithm is executed.

In the fifth step, the process is repeated until the optimum is found or a pre-specified temperature T is reached.

And at last, we obtain the optimal particles $\{\dot{x}_k^i, \dot{w}_k^i : i = 1,2\ldots N\}$ and estimate the system state $x_k = \sum_{i=1}^N \dot{x}_k^i \dot{w}_k^i$.

IV. EXPERIMENT RESULTS AND DISCUSSIONS

To compare the performance of the proposed filters to those of the PF, the RPF, the PFDE and the APF, where the PFDE combines particle filter with differential evolution [19]. We choose the same model as Merwe, et al. used in his experiments [14]. This model is very representative due to its strong nonlinearity. And it has been used before in many publications [18-21]:

$$x_{k+1} = 1 + \sin(w\pi k) + \phi_1 x_k + v_k, \qquad (12)$$

$$z_k = \begin{cases} \phi_2 x_k^2 + n_k & k \le 30 \\ \phi_3 x_k - 2 + n_k & k > 30 \end{cases}, \qquad (13)$$

where v_k is a Gamma(3,2) random variable modeling the process noise, $w = 4e-2$, $\phi_1 = \phi_3 = 0.5$, and $\phi_2 = 0.2$ are scalar parameters. The observation noise n_k is drawn from a Gaussian distribution $N(0, 0.00001)$. Different filters are used to estimate the state sequence x_k for $k = 1,2,\ldots T$, the total observation time is $T = 50$. The UKF parameters were set to $\alpha = 1$, $\beta = 0$ and $\kappa = 2$ [14]. In the proposed filters, $F = 0.9$, $Cr = 0.6$ and the maximum number of generations is $G = 20$, the annealing initial temperature $T0 = 100$, and the annealing parameter $\rho = 0.9$ [17,19]. All of the particle filters used $N = 10$ particles and systematic resampling. The experiment was repeated $M = 200$ Monte Carlo simulations to demonstrate the performance of the proposed algorithm.

To measure the performance of the algorithms, we introduce the Root Mean Squared Error ($RMSE$) and its mean $\overline{RMSE}$, RMSE for M simulations with observation time $RMSE'$ are shown as follows:

$$RMSE = \sqrt{\frac{1}{T}\sum_{k=1}^{T}\left(x_k - \hat{x}_k\right)^2}, \qquad (14)$$

$$\overline{RMSE} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{T}\sum_{k=1}^{T}(x_k^m - \hat{x}_k^m)^2\right]}, \qquad (15)$$

$$RMSE' = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(x_k^m - \hat{x}_k^m)^2}, \qquad (16)$$

where x_k^m is the true value of target state and $\hat{x}_k^m$ is defined as the estimation of target state.

A. Simulation results of the estimation

Figure 1 compares the estimates of the different filters generated from a single run of the state estimates. From Fig. 1, we can see that the tracking trajectory of the HDEPF is much closer to the true trajectory than other filters.

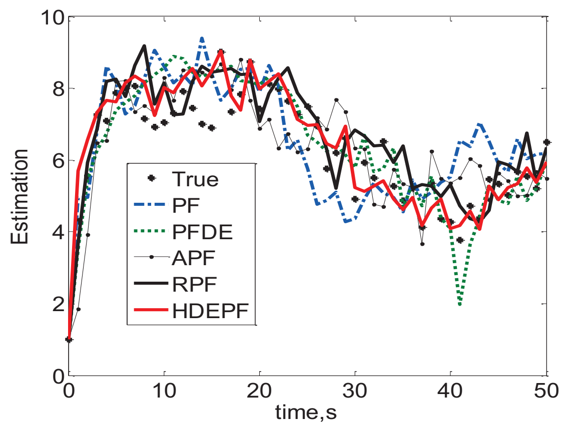

Fig. 1. Plot of estimates generated by the different filters for a simulation.

For clearly, the estimation RMSE with observation time and the estimation RMSE with simulation number of different filters are shown in Figs. 2 and 3, respectively. We can see that both two kinds of the RMSE curves of the proposed method are lower than other algorithms. Moreover, it can be found that the RMSE of HDEPF is higher than other methods before 2s or so in Fig. 2. That is because the UKF and HDE algorithms both need initialized process, but this time is shorter, as a whole, the result of the experiment proves that the new algorithm has good optimization effect. Meanwhile, Fig. 3 also illustrates this point.

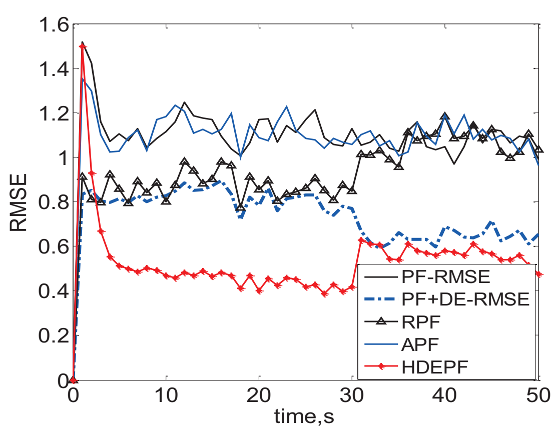

Fig. 2. RMSE' of PF, APF, RPF, PFDE and HDEPF with observation time for 200 MC simulations, where N=10.

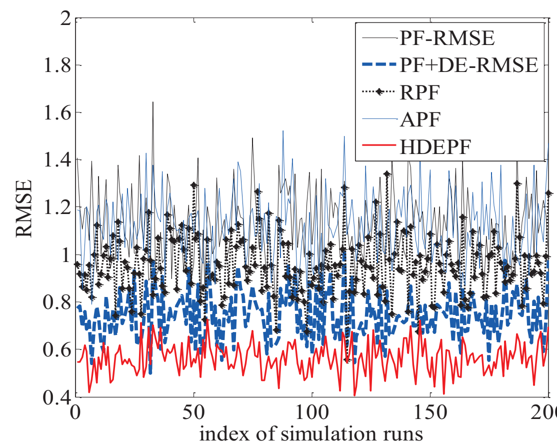

Fig. 3. RMSE of PF, APF, RPF, PFDE and HDEPF with simulation number, where N=10, T=50.

Table 1 displays and summarizes the performance of the five filters, where the means and variances of the state estimates are shown. It can be clearly seen that the Mean RMSE of the HDEPF is lower than others, as well as, the Variance RMSE is obviously low. From Figs. 1, 2, 3 and Table 1, we can realize that the estimation accuracy of the proposed algorithm (HDEPF) is much higher than other filters.

Table 1: Mean and variance of RMSE of PF, APF, RPF, PFDE and HDEPF for 200 MC simulations with N=10

Algorithm	Mean (RMSE)	Variance (RMSE)
PF	1.1132	0.0178
PFDE	0.7422	0.0103
APF	1.1049	0.0247
RPF	0.9543	0.0213
HDEPF	0.5574	0.0047

It is shown in Fig. 2 that the proposed algorithm has better estimation accuracy when $k \le 30$. According to equation (13), we can find that the observation function is a quadratic function in the first 30 seconds, and it becomes a linear function after 20 seconds. Before 30 seconds, the nonlinear degree of system is greater than later time. From Table 2 we can see that the RMSE of the proposed algorithm has decreased by around 50% over the PF when $k \le 30$. This shows that the proposed algorithm has better performance for nonlinear filtering than the PF algorithm.

Table 2: RMSE mean of PF and HDEPF for 200 MC of different observation time periods

Algorithm	0<k<30	30<k<50
PF	1.1232	0.7131
HDEPF	0.5654	0.3895
The percentage of improvements	49.66%	45.38%

B. Analysis of the runtime of the algorithms

Table 3 compares the run time for a simulation of different algorithms, where $N=10$ except the bracket, there $N=100$. It can be seen that the proposed algorithm has much more run time than others, except RPF and PFDE; because the proposed algorithm not only has the UKF filter but also combines the HDE, it wastes much time. The HDE uses the search capability of the SA algorithm, which improves its convergence speed. So the proposed algorithm has less run time than the PFDE algorithm.

Table 3: Comparison of the run time

Algorithm	Run Time
PF	0.031027 (0.137454/N=100)
PFDE	0.339256
APF	0.072424
RPF	0.372693
HDEPF	0.112712

Then we increase the particle number of the PF, here $N=100$, and kept the other particle numbers. It can be calculated that the RMSE mean of the PF with $N=100$ is 0.87414. From Table 1, we can see that the estimation accuracy of the PF with $N=100$ is also not as good as the proposed algorithm. However, Table 3 shows that the run time of the PF with $N=100$ is as much as the proposed algorithm. This represent that the estimation performance of the proposed algorithm is higher than the PF algorithm with the same run time. In addition, the proposed algorithm uses only 10 particles to reach the precision of the PF with $N=100$, it shows that the proposed algorithm has better efficiency. In the future, it will be interesting to investigate how to choose the parameters of the new algorithm.

V. CONCLUSION

In this paper, a new particle filter algorithm was developed for nonlinear filtering. Firstly, we use the experience of the UPF algorithm for reference to generate the importance proposal distribution though the UKF. Since the generated distribution matches the true posterior more closely. Secondly, the Hybrid Differential Evolution (HDE) based on SA is employed as the resampling scheme and is the major new contribution of this paper. The proposed resampling algorithm can effectively reduce the particle degeneracy and impoverishment problem, and improves the state estimation accuracy. In addition, the convergence performance of the HDE is better than the DE by utilized search capability of the SA [17]. Therefore, the proposed algorithm yields a better performance than the particle filter which based on the DE. Moreover, it has less run time. The numerical simulations were conducted to attest that the proposed algorithm has better estimation performance and higher particle utilization than the previous method. Future works will concentrate on the nonlinear no Gaussian radar target tracking using the proposed algorithm.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (Grant No. 61172159) and the Fundamental Research Funds for the Central Universities (HEUCFT1101).

REFERENCES

[1] H. Tanizaki, "Nonlinear filter: estimation and applications (2nd ed.)," *Springer Verlag*, New York, 1996.

[2] M. West and J. Harriaon, "Bayesian forecasting and dynamic models (2nd ed.)," *Springer Verlag*, New York, 1997.

[3] J. Liang and X. Y. Peng, "Particle estimation algorithm using correlation of observation for nonlinear system state," *Electronics Letters*, vol. 44, no. 8, pp. 553-555, April 2008.

[4] J. Liang and X. Y. Peng, "Improved particle filter for nonlinear system state," *Electronics Letters*, vol. 44, no. 21, pp. 1275-1277, October 2008.

[5] A. Doucet, N. de Freitas, and N. Gordon (Eds.), "Sequential monte carlo methods in practice," *Springer Verlag*, New York, 2001.

[6] B. D. Anderson and J. B. Moore, "Optimal filtering," *Prentice-Hall*, New Jersey, 1979.

[7] K. Reif, "Stochastic stability of the discrete-time extended Kalman filter," *IEEE Trans. Automatic Contro.*, vol. 44, no. 4, pp. 714-728, 1999.

[8] S. J. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of

probability distributions," *Dept. of Engineering Science, University of Oxford*, 1996.
[9] Z. Chen, "Bayesian filtering: from kalman filters and beyond," *Communication Research Laboratory, McMaster University*, 2003.
[10] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEEE Proc. F.*, vol. 140, no. 2, pp. 107-113, 1993.
[11] M. Pitt and N. Shephard, "Filtering via simulation: auxiliary particle filter," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590-559, 1999.
[12] C. Musso, N. Oudjane, and F. Legland, "Improving regularised particle filter," *In Sequential Monte Carlo Methods in Practice, Springer-Verlag*, pp. 247-271, 2001.
[13] J. F. G. De Freitas, M. Niranjan, A. H. Gee, and A. Doucet, "Sequential monte carlo methods to train neural network models," *Neural Comput.*, vol. 12, no. 4, pp. 955-993, 2000.
[14] R. Merwe, A. Doucet, N. Freitas, and E. Wan, "The unscented particle filter," *Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department*, pp. 1-40, 2000.
[15] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous space," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.
[16] S. Rahnamayan and H. R. Tizhoosh, "Opposition-based differential evolution," *IEEE Trans. On Evolutionary Computation*, vol. 12, pp. 64-78, 2008.
[17] H. ZhongBo and X. ShengWu, "Study of hybrid differential evolution based on simulated annealing," *Computer Engineering and Design*, vol. 28, no. 9, pp. 1989-1992, May 2007.
[18] Y. Wang, F. Sun, and Y. Zhang, "Central difference particle filter applied to transfer alignment for SINS on missiles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 1, pp. 375-387, 2012.
[19] H. Li, J. Wang, and H. Su, "Improved particle filter based on differential evolution," *Electron. Lett.*, vol. 47, no. 19, pp. 1078-1079, 2011.
[20] J. Y. Zuo and Y. N. Jia, "Adaptive iterated particle filter," *Electron. Lett.*, vol. 49, no. 12, pp. 742-744, June 2013.
[21] C. Qi, "An efficient two-stage sampling method in particle filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 3, pp. 2666-2672, July 2012.

Chaozhu Zhang was born in 1970. He received his B.S. degree in Electronics and Information Engineering from Harbin Institute of Technology in 1993, his M.S. degree in Communications and Information Systems and his Ph.D. degree in Signal and Information Processing from Harbin Engineering University in 2002 and 2006, respectively. He is a Professor for Harbin Engineering University, China. He is a member of IEEE, academician of Chinese Aerospace Society and Heilongjiang Biomedical Engineering Society. His researches include signal processing applications in radar and communications and image processing.

Lin Li was born in 1987. She received her B.S. degree in Electronics and Information Engineering from Anhui Agricultural University in 2009. Now she is a Ph.D. candidate in Harbin Engineering University, China. Her research interests include radar signal processing, nonlinear filtering and target tracking.