Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **4**, July **2018**

Node Tacking Location Algorithm for Wireless Sensor Networks Based on Improved Particle Filter

Chun-Ming Wu, Xue Yang, Hao-Quan Gong
College of Information Engineering
Northeast Electric Power University
No.169, Changchun Rd., Chuanying, 132012, Jilin, China
466389144@qq.com; 627053170@qq.com 1814864076@qq.com

Received May, 2017; revised January, 2018

Abstract. *To solve the degeneracy phenomenon and the sample impoverishment problem of basic particle filter(PF) algorithm, and to improve the target tracking accuracy of the wireless sensor network, this paper put forward a target tracking method for WSN based on improved particle filter method. By using the maximum a posteriori probabilistic estimation of ensemble kalman filter to generate the proposal distribution function of each particle at each moment of particle filter, and the latest observation information is integrated into the proposal distribution function, so the proposal distribution can approximate the true posterior distribution more accurately and greatly improve the target tracking accuracy, replacing re-sampling stage by introducing the artificial fish-swarm algorithm to optimize the particle distribution, which makes prior particles move towards the high likelihood region and finds the optimal position, increases the number of effective particles, enhances the diversity of the particles and improves the problem of particle depletion. The simulation results show that the improved algorithm is better than the traditional target tracking algorithm in tracking accuracy, stability and reliability.*
Keywords: Please write down the keywords of your paper here, such as, Watermarking, Video compression, ...

1. **Introduction.** As a new technology of human life in the new century, wireless sensor network (WSN) is favored by domestic and foreign scholars, and the research results are numerous. However, it is of great practical significance to study the target tracking algorithm in wireless sensor networks in many applications such as enemy invasions and force mobilization, traffic control and so on, which need dynamic real-time access to the target location and other information in wireless sensor networks [1, 2, 3].

When tracking in wireless sensor networks, the current position and motion trajectory of one or more targets are determined through the interaction data between nodes. At the same time, it is necessary to make an accurate prediction of the target state of motion at the next moment. In order to reduce the energy consumption, the monitoring node needs to wake up early and dynamically join the target tracking process according to the prediction result. During the tracking process, the sensor node accurately estimates the current state of the target through the filtering algorithm according to the monitoring data. The particle filter method can deal with the state estimation of moving targets better, but the traditional particle filter algorithm has two shortcomings in the state estimation of the target [4, 5, 6]:

(1) does not take into account the current observations: In the implementation of particle filter, the prior probability density function is usually selected as the proposal

distribution function, however, this method does not incorporate the latest observation information into it, which causes large deviation from the actual value, resulting in a large variance of the right value, thus affecting the filtering accuracy. In the literature [7, 8], we propose to use EKF and UKF to generate the proposal distribution, and the algorithm accuracy is improved. However, when the EKF algorithm is run, high-order truncation of nonlinear systems will produce strong errors. If the non-linearity of the system is strong, the estimation error will increase. The UKF is used to obtain the proposal distribution. Although the system is up-to-date, the state estimation performance of the particle filter is greatly improved. However, since the number of samples of the Sigma point is related to the system dimension, the complexity of the algorithm is higher when the dimension is increased.

(2) particle depletion phenomenon: in the re-sampling stage, by copying large weight particles and removing small weight particles, which makes the sampling results contain many repeated points, reducing the diversity of particles to a certain extent. In the literature[9], genetic algorithm is used to select the seeds, such as selection, crossover and mutation, to obtain more fine particles and improve the particle degradation problem. In the literature[10], the ant colony thought is introduced into the particle filter algorithm, in which instead of the re-sampling step. According to the biological mechanism of the ant looking for food, the sample particle transfer process is equivalent to the route of the ants, thus avoiding the problem of sample depletion and improving the target tracking accuracy.

In this paper, an improved particle filter algorithm is proposed, in which is based on the ensemble kalman method generating the proposal distribution function of particle filter; at the same time, the artificial fish-swarm algorithm is used to optimize the particle distribution, so that the particles are concentrated in the high likelihood domain, increasing the diversity of the particles and suppressing the phenomenon of particle dilution, thus increasing the reliability and stability of the target tracking.

2. Target Tracking Algorithm for Wireless Sensor Network.

2.1. Target Tracking Architecture. In the WSN, the target tracking system is shown in Figure 1. The sensor nodes deployed in the monitoring area, As the network is initialized, all nodes in the network are located in real time in order to keep track of the targets to be monitored.

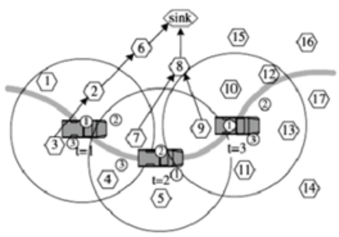

FIGURE 1. The general architecture of target tracking in WSN

When the target moves, the nearby node will detect the target and get the corresponding observation value. The node sends the observed value to the convergence node through the multi-hop mode. The convergence node obtains the estimated value of the target's current position by the filtering algorithm, and predicts the next time target location to achieve the dynamic tracking of the target.

2.2. **Target Tracking Problem Description.** For the target tracking problem in WSN, the moving process of the target (state transition equation) is usually described by the first-order Markovian equation of state:

$$x_k = f(x_{k-1}) + w_k \tag{1}$$

Where x_k represents the target state vector (such as the position, velocity, acceleration, etc.) of the target; $f(x_{k-1})$ is the state transition function of the target state from x_{k-1} to x_k; w_k is the process noise.

The general form of observation equation at time K is:

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

Where z_k is the observation vector obtained in the node at time x_k, $h(x_k)$ is the observation function, and v_k is the observation noise.

The problem to be solved by the target tracking is to solve the current state of the target based on the state before the target and the resulting observation vector.

2.3. **Particle Filter Algorithm to Achieve Target Tracking.** Particle filtering is based on a set of weighted random samples to approximate the probability density function, the sample mean instead of the integral operation, in order to obtain the state of the minimum variance estimation process. The principle is that the samples of N equal weights are randomly sampled from the prior probability distribution of the state at the time of K-1; When the measured value arrives, the particle weights are updated according to the likelihood function to obtain the set $\{x_{k-1}^i, \tilde{w}_{k-1}^i\}$; Then the re-sampling process is performed, the large weight particles are copied, the small weight particles are discarded, and the weight of the particles after re-sampling is set to $1/N$; finally, through the state transition model of the system predicted the state of the particle at time K, and using the observations to update the particle weights in time to obtain the weighted particle set $\{x_k^i, \tilde{w}_k^i\}$ at time K. The specific steps of a standard particle filter algorithm to estimate the target state vector at time k are as follows:

(1)Initialization:The initial set of primitives is sampled in the known prior probability density $p(x_0)$,and each particle in the set is equal $\{x_0^i, 1/N; i = 1, 2, \cdots, N\}$;

(2)Importance sampling;

a) Randomly extract N particles from the proposal distribution function;

$$x_k^i \sim q(x_k^i | x_{k-1}^i, z_k) = p(x_k^i | x_{k-1}^i), i = 1, 2, \cdots, N \tag{3}$$

b) Update the weight of the particle and normalize the weight;

$$w_k^i = w_{k-1}^i p(z_k | x_k^i) \tag{4}$$

$$\tilde{w}_k^i = w_k^i / \sum_{i=1}^{N} w_k^i \tag{5}$$

(3) Re-sampling: Calculate the number of valid samples $\hat{N}_{eff}$;

$$\hat{N}_{eff} = \frac{N}{\sum_{i=1}^{N} (W_k^i)^2} \tag{6}$$

If $\hat{N}_{eff} < N_{thr}$,N_{thr} is the set threshold, then re-sampling according to $p(\tilde{x}_k^j = x_k^i) = \tilde{w}_k^i$, get a new support particle set $\{\tilde{x}_k^j, 1/N; j = 1, 2, \cdots, N\}$;

(4)State estimation: system state and variance;

$$\begin{cases} \hat{x}_k = E(x_k|z_k) \approx \sum_{i=1}^{N} x_k^i \cdot \tilde{w}_k^i \\ P_k = \sum_{i=1}^{N} \tilde{w}_k^i (\hat{x}_k - x_k^i)(\hat{x}_k - x_k^i)^T \end{cases} \tag{7}$$

(5)Let $k = k + 1$, return to the second step.

3. **Improved Particle Filter Target Tracking Algorithm.** In order to overcome the shortcomings of particle filter, the ensemble kalman filter method will be used to generate the proposal distribution. Through the sampling set approximation of the real statistical value, the estimation accuracy is improved, and the calculation amount is effectively controlled. At the same time, in order to solve the problem of particle diversity loss caused by sample depletion in the process of particle filtration, the artificial fish-swarm algorithm is introduced into the particle filter. By optimizing the distribution of particles, the particles are concentrated in the high likelihood domain, increasing the number of effective particles, so that the diversity of particles is enhanced and the phenomenon of particle depletion is improved.

3.1. **Ensemble Kalman Filter(Enkf).** The Enkf is based on the Monte Carlo kalman filter method, which uses the form of a set to approximate the posterior probability density distribution of the nonlinear function without the need for linearization of the system, so that can be very good solve the nonlinear problem, meanwhile the collection of sampling points can be arbitrarily set, in dealing with practical problems with great advantages[12, 13]. The principle is to define the state of the system as a set of background collection, using the kalman method and the latest observation data to update the background set of each sampling sample, getting the analysis set which used to accurately predict the current state of the target.

Set the background set for the k moment:$X_k^b = \{x_{k,i}^b, i = 1, 2, \cdots, n\}$ (n is the number of sets of samples), it is passed from the analysis set of the previous moment:$X_{k-1}^a = \{x_{k-1,i}^a, i = 1, 2, \cdots, n\}$. The sampling mean and variance of the background set are:

$$\hat{x}_k^b = \frac{1}{n} \sum_{i=1}^{n} x_{k,i}^b \tag{8}$$

$$\hat{P}_k^b = \frac{1}{n-1} \sum_{i=1}^{n} (x_{k,i}^b - \hat{x_k^b})(x_{k,i}^b - \hat{x_k^b})^T \tag{9}$$

In practical applications, (9) can be replaced by the following formula:

$$\begin{cases} \hat{P}_{xh}^k = \frac{1}{n-1} \sum_{i=1}^{n} (x_{k,i}^b - \hat{x_k^b})[h(x_{k,i}^b) - h(\hat{x_k^b})]^T \\ \hat{P}_{hh}^k = \frac{1}{n-1} \sum_{i=1}^{n} [h(x_{k,i}^b) - h(\hat{x_k^b})][h(x_{k,i}^b) - h(\hat{x_k^b})]^T \end{cases} \tag{10}$$

The kalman gain is calculated as:

$$K_k = \hat{P}_{xh}^k (\hat{P}_{hh}^k + R_k)^{-1} \tag{11}$$

where R_k represents the observed error covariance matrix at time k.

In the background set into the latest observation data node, k time to get the analysis of the collection:

$$x^a_{k,i} = x^b_{k,i} + K_k[z_{k,i} - h(x^b_{k,i})], i = 1, 2, \cdots, n \tag{12}$$

Where $z_{k,i}$ is the Gaussian distribution with mean z_k and variance R_k.

The sampling mean and variance of the analysis set are:

$$\hat{x}^a_k = \frac{1}{n}\sum_{i=1}^{n} x^a_{k,i} \tag{13}$$

$$\hat{P}^a_k = \frac{1}{n-1}\sum_{i=1}^{n}(x^a_{k,i} - x^a_k)(x^a_{k,i} - x^a_k)^T \tag{14}$$

The ensemble kalman recursively estimates the posterior probability density by propagating the gaussian estimate of the posterior distribution at the same time combined with the latest observations at each moment:

$$p(x_k|Z_{1:k}) \approx p_N(x_k|Z_{1:k}) = N(\hat{x}_k, \hat{P}_K) \tag{15}$$

When performing particle filtering, using a separate Enkf for each particle to generate and pass the gaussian proposal distribution:

$$q(x^{(i)}_k | x^{(i)}_{1:k-1}, z_{1:k}) = N(\hat{x}^{(i)}_k, \hat{P}^{(i)}_k) \tag{16}$$

At time k-1, the mean and variance of the proposal distribution of each particle are updated using Enkf and the most recent observations:

$$\begin{cases} \hat{x}^b_k = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} x^b_k \\ \hat{P}^b_k = \dfrac{1}{n-1}\displaystyle\sum_{i=1}^{n}(x^b_{k,i} - x^{\hat{b}}_k)(x^b_{k,i} - x^{\hat{b}}_k)^T \end{cases} \tag{17}$$

Because Enkf can obtain the maximum posteriori probability estimate of the state, the latest observation information is integrated into the proposal distribution, which makes the sampling sample closer to the real sample, while avoiding the linearization of the nonlinear system, effectively controlling the computation and improving the algorithm performance.

3.2. **Artificial Fish-Swarm Algorithm(AFSA)..** The basic idea of AFSA is that the largest number of fish in a water is the most nutritious place in the water, and the global optimization is achieved by simulating the behavior of fish in nature.

The artificial fish is regarded as the particle filter algorithm in each particle, scattered around the target being tracked, because the AFSA to find food on the mathematical problem is equivalent to seeking the optimal solution, and the particle filter algorithm to find the most similar goal is equivalent and the AFSA has the characteristics of fast convergence and good real-time. Therefore, the AFSA is introduced into the particle filter algorithm, in which is used to optimize the distribution of the particles, so that the particles accumulate in the food within the optimal solution range.The pattern of artificial fish-swarm algorithm:

1. foraging behavior

In the n-dimensional target search space D, there are n artificial fish, and the state of the i-th artificial fish is expressed as:

$$X^{(i)} = (x^{(i)}_1, x^{(i)}_2, \cdots, x^{(i)}_d) \in D, i = 1, 2, \cdots, n \tag{18}$$

Set the food concentration of the current location of the artificial fish to be expressed as:

$$Y = f(x) \tag{19}$$

Where Y is the objective function value;

The distance between artificial fish is expressed as individual:

$$d_{i,j} = \|X_i - X_j\| \tag{20}$$

Set the current state of artificial fish to X_i, in its perception range ($d_{i,j} < v$) randomly select a state X_j, v said the artificial fish perception distance; If $Y_i < Y_j$, then move forward in that direction:

$$Y_i < Y_j, X_{inest} = X_i + r \cdot s \cdot \frac{X_j - X_i}{\| X_j - X_i\|} \tag{21}$$

Otherwise, move one step at random:

$$X_{inext} = X_i + r \cdot s \tag{22}$$

Where r is the random number between $(0, 1)$ and s represents the maximum step size of the move.

(2)clustering behavior

Artificial fish are naturally clustered in the process of swimming, and when the number of other cluster partners n_f and the center position X_C are explored in their perceptual range ($d_{i,j} < v$), If $Y_c/n_f > \delta Y_i$, indicates that the partner center has a lot of food and is not too crowded, while move to the partner heart:

$$Y_c/n_f > \delta Y_i, X_{inest} = X_i + r \cdot s \cdot \frac{X_c - X_i}{\| X_c - X_i\|} \tag{23}$$

Otherwise, the implementation of foraging behavior.

(3)rear-end behavior

Assuming $\hat{z}_{k|k-1}$ in the scope of v there are other artificial fish, select the largest Y value of the artificial fish Y_j, when $Y_i < Y_j$, the particle by comparing the objective function of constantly updated themselves to the real state close.

$$\hat{x}^{i_m}_{k|k-1} = \hat{x}^{i_{m-1}}_{k|k-1} + r \cdot s \frac{\hat{x}^{j}_{k|k-1} - \hat{x}^{i_{m-1}}_{k|k-1}}{\|\hat{x}^{j}_{k|k-1} - \hat{x}^{i_{m-1}}_{k|k-1}\|} \tag{24}$$

Where $\hat{x}^{i_m}_{k|k-1}$ is the iteration value of $\hat{x}^{i}_{k|k-1}$ after m-th .

3.3. **Improved Target Tracking Algorithm Steps.** In order to optimize the PF sampling process, the AFSA is incorporated into the PF algorithm. The latest observations are introduced into the sampling process and the objective function is defined as :

$$Y = \left[(2\pi)\sigma_v^2\right]^{-\frac{1}{2}} e^{-\frac{1}{2\sigma_v^2}\left[(z_k - \hat{z}^i_{k|k-1})^2\right]} \tag{25}$$

Where z_k is the latest observed value and $\hat{z}_{k|k-1}$ is the predicted value.

The algorithm steps are as follows:

Step1:At k=0, the initial sampling set is obtained in the prior probability distribution $p(x_0)$, and the weight of each particle is $1/N$. At the same time, the background set $X_k^{(i),b} = \{x_{k,j}^{(i),b}, j = 1, 2, \cdots, n\}$ and the analysis set $X_k^{(i),a} = \{x_{k,j}^{(i),a}, j = 1, 2, \cdots, n\}$ are defined for each particle of the initial sample, and the background set is initialized with a set of state Gaussian samples.

Step2: Use Enkf to update the particles to produce the importance of each particle density function:

(1) Calculate the mean and variance of the samples of each particle background at time k, and obtain $\hat{x}_k^{(i),b}$, $\hat{P}_{xh}^{(i)k}$, $\hat{P}_{hh}^{(i)k}$;

(2)The background set of each particle at time k is updated to the analysis set to obtain $X_0^{(i),a} = \{x_{0,j}^{(i),a}, j = 1, 2, \cdots, n\}$; and at the same time calculate the mean and variance of each particle analysis set as a gaussian approximation of the particle density function:

$$\hat{x}_k^{(i)} \sim q(x_k^{(i)}|x_{k-1}^{(i)}, z_k) = N(\hat{x}_k^{(i),a}, \hat{P}_k^{(i),a})$$

Step3: Re-sample from the acquired importance distribution to get a new set of particles.

Step4:Calculate the weight of each particle from equation (4) and normalize it with equation(5)

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

Step5: Using the objective function (25), a set of optimal solutions is calculated as the current initial particle set.

Step6: Perform the artificial fish swarm algorithm.

The particles are optimizedeach particle executes n iterations and adaptively selects a move strategy. First, n times the foraging action is performed. If a better position can not be found within a specified number of times, artificial fish automatically perform clustering behavior, followed by rear-end behavior, and ultimately get closer to the real state of the particle set $\{\tilde{x}_k^i, i = 1, 2, \cdots, n\}$.

Step7: Update the weight of the particle and normalize the weight:

$$w_k^i = w_{k-1}^i p(z_k|x_k^i);$$

$$w_k^i = w_k^i / \sum_{i=1}^{N} w_k^i$$

Step8:Re-sampling: If $\hat{N}_{eff} = \frac{N}{\sum_{i=1}^{N} (W_k^i)^2} < N_{thr}$, N_{thr} is the set threshold, get a new support particle set $\{\tilde{x}_k^j, 1/N; j = 1, 2, \cdots, N\}$.

Step 9: State and variance estimates.

Step10: Determine whether the algorithm is finished. If so, exit the algorithm. Otherwise, let $k = k + 1$ return to Step3 to estimate the next time of the dynamic target.

4. **Simulation Results and Analysis.** In order to compare the proposed algorithm and the existing filtering proposed algorithm to estimate the performance of nonlinear systems, a univariate non-static growth model (UNGM model) is used to simulate the experiment. This model is a typical verification model that compares the performance of various PF algorithms, and its state posterior distribution has bimodal characteristics and the system is non-linear.

The equation of state is:

$$x(t) = 0.5x(t-1) + \frac{25x(t-1)}{1 + [x(t-1)]^2} + 8\cos[1.2(t-1)] + w(t) \tag{26}$$

The measurement equation is:

$$z(t) = \frac{x(t)^2}{20} + v(t) \tag{27}$$

Where $x(t)$ is the state quantity; $z(t)$ is the observation quantity; $w(t)$ is the process noise; $v(t)$ is the observation noise.

In this paper, root mean square error (RMSE) is used as the index of performance evaluation. The average mean square error is defined as:

$$RMSE = \left[\frac{1}{T} \sum_{t=1}^{T} (x_t - \hat{x}_t)^2 \right]^{\frac{1}{2}} \tag{28}$$

Where T is the simulation time; $x(t)$ is the target state vector; $\hat{x}(t)$ is the estimated state vector.

1)Let the number of particles N=100, the process noise $w_k \sim (0, 1)$, no observation noise. Simulation results shown in Figure 2.

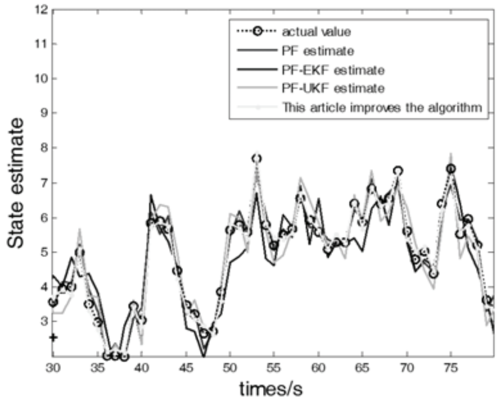

FIGURE 2. Comparison of the filter algorithm state(no observation noise)

2)Let the number of particles N = 50, the process noise $w_k \sim (0, 1)$, measurement noise $v_k \sim (0, 1)$, are subject to zero mean gaussian distribution. Simulation results shown in Figure 3, tracking error curve shown in Figure 4.

3)Let the number of particles N = 100, the process noise $w_k \sim (0, 1)$. measurement noise $v_k \sim (0, 1)$, are subject to zero mean Gaussian distribution. Simulation results shown in Figure 5, tracking error curve shown in Figure 6.

Figure 3 Comparison of different algorithms' Figure 4 numerical compariso of various tracking filtering results in N =50 Algorithms for N = 50

Figure 5 Comparison of different algorithms' Figure 6 numerical comparison of various tracking filtering results in N = 100 Algorithms for N = 100

Table 1 shows the performance comparison of various filtering algorithms for different particle numbers. It can be seen from the table that the error of PF-EKF, PF-UKF and the improved algorithm is obviously smaller than that of PF algorithm, and the error of the improved algorithm is the smallest. At the same time, the improved algorithm reduces the calculation time. In addition, as the number of particles increases, the performance of the algorithm is also improved. However, since the PF-EKF, PF-UKF and the improved algorithm are sensitive to the number of iterations, the algorithm increases the computational complexity of the algorithm by increasing the number of particles. Therefore, the number of particle samples should be properly selected.

Through the simulation results and Table 1 can be seen that PF-EKF and PF-UKF and the improved algorithm are both the proposal distribution function of particle filter with maximum a posteriori estimation. By combining the latest observations with the prior probability density function closer to the true posterior probability distribution, performance, accuracy are better than PF algorithm. Because PF-EKF directly generates high-order truncation errors for the system, the PF-EKF algorithm performs poorly when the system nonlinearity is strong and the error will further increase. The

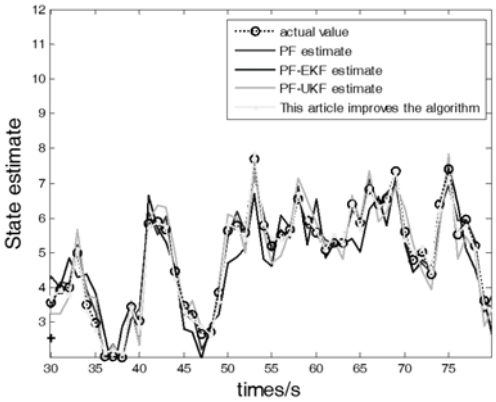

FIGURE 3. Comparison of different algorithms' tracking filtering results in N =50

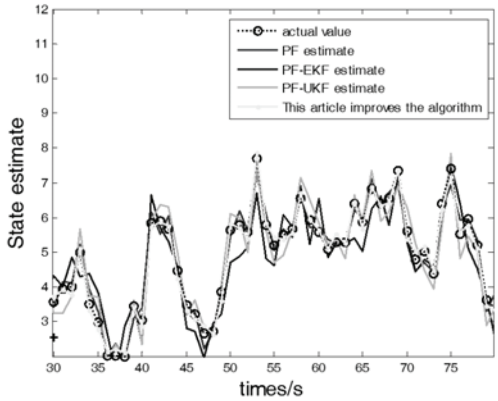

FIGURE 4. numerical compariso of various Algorithms for N = 50

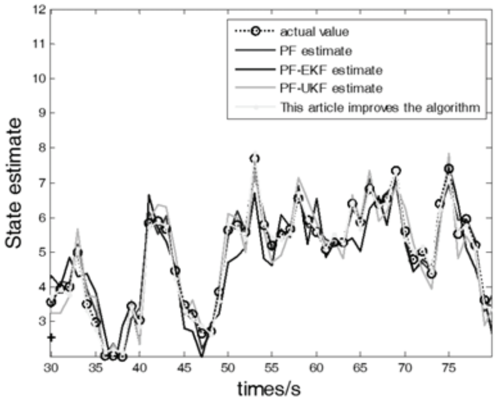

FIGURE 5. Comparison of different algorithms' tracking filtering results in N = 100

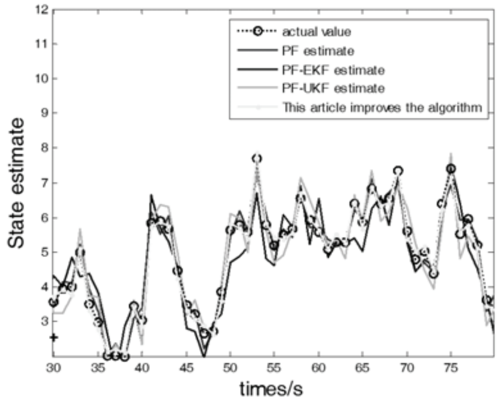

FIGURE 6. numerical comparison of various Algorithms for N = 100

PF-UKF and the improved algorithm are approximated by the non-linear distribution of the sampling method, which avoids the error caused by the linearization process and further improves the estimation accuracy. In the practical application, the target tracking is a high-dimensional nonlinear system. The PF-UKF algorithm produces a large number of sampling points when generating the proposed distribution function by using no-trace transform. When the dimension is increased, the computational complexity is complicated and the algorithm complexity is high. When the proposed improved algorithm is used to generate the proposal distribution function, the actual statistical value is estimated by analyzing the set, and the number of sampling points can be set flexibly to balance the estimation accuracy and the computational cost. At the same time, artificial fish-swarm algorithm is introduced in the re-sampling stage to move particles to high likelihood domain, which increases the number of effective particles, weakens the sample depletion problem and improves the estimation performance of the algorithm.

5. **Conclusion.** In this paper, the ensemble kalman filter and artificial fish-swarm algorithm are introduced into the particle filter algorithm. An improved particle filter algorithm is proposed to estimate the target motion state accurately. The algorithm obtains the proposal distribution function by using Enkf and the latest current observed data obtained by combining nodes. Then, the artificial fish-swarm algorithm is used to

TABLE 1. Comparison of Filtering Algorithm Performance

Parameter	Filtering algorithm	RMSE mean	RMSE variance	Run time
			0.0576	
			0.0287	
			0.0195	
			0.0129	
	PF			0.9133
N=50	PF-EKF		0.0493	1.6746
	PF-UKF		0.0206	3.7524
	improve algorithm		0.0147	2.1138
			0.01030.5282	
	PF		0.3398	1.1689
N=100	PF-EKF		0.2302	2.3024
	PF-UKF		0.0682	5.3147
	improve algorithm		2.6425	
			0.4765	
			0.3108	
			0.1824	
			0.0519	

optimize the particle distribution in the re-sampling stage so that the particles are close to the real state, which enhances the diversity of the particles and improves the accuracy of the target tracking. Finally, this paper verifies the effectiveness of the improved algorithm by matlab simulation. It can be seen that the algorithm proposed in this paper is much more accurate than the traditional target tracking algorithm and has high robustness. At the same time, the improved algorithm optimizes the particle distribution state, particle diversity has been enhanced to improve the phenomenon of particle depletion, so that the tracking accuracy and reliability of the target to be further enhanced.

REFERENCES

[1] T. T. Ding, M. F. Gao.Target Tracking Algorithm for Wireless Sensor Networks with Improved Particle Filter, *Journal of Sensors and Microsystems*, vol. 7. no. 2, pp. 140-1422016.

[2] M. Orton, W. Fitzgerald, A Bayesian Approach to Tracking Multiple Targets Using Sensor Array and Particle Filters, *Journal of IEEE Transactions on Signal Processing*, 2012, 50(2), pp. 216-223.

[3] M. Gan, Y. Cheng, Y. Wang, Hierarchical Particle Filter Tracking Algorithm Based on Multi-Feature Fusion, *Journal of Journal of Systems Engineering and Electronics*, 1, no. 7), pp. 51-62, 2016.

[4] T. W. Sung, C. S. Yang, A Voronoi-Based Sensor Handover Protocol for Target Tracking in Distributed Visual Sensor Networks, *International Journal of Distributed Sensor Networks*, vol. AID 586210, March 2014, pp. 1-14, 2014.

[5] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S.K. Pang, S. Godsill, Overview of Bayesian sequential Momte Carlo methods for group and extended object tracking, *Journal of Digital Signal Processing*, vol. 25, no. 8), pp. 1-16, 2014.

[6] H. X. Xie, Z. Y. Xie, X. F. Tu. Research on Multi-target Tracking Algorithm Based on Improved Particle Filter, *Journal of Computer Engineering and Design*, 2014, 35, no. 6), pp. 2142-2146.

[7] Y. J. Chen, G. J. Horng, and S. T. Cheng, A Distributed Cross-Layer Compromise Detection Mechanism for Wireless Sensor Networks, *Journal of Journal of Network Intelligence*, vol. 2, No. 1, pp. 147-161, Feb 2017.

[8] T. T. Nguyen, J. S. Pan, S. C. Chu, J. F. Roddick, and T. K. Dao, Optimization Localization in Wireless Sensor Network Based on Multi-Objective Firefly Algorithm, *Journal of Network Intelligence*, vol. 1, No. 4, pp. 130-138, Dec. 2016.

[9] D. Meana-Llorián, Cristian González García, Vicente García-Díaz, B. Cristina Pelayo G-Bustelo, and Nestor Garcia-Fernandez. SenseQ, pp. Replying questions of Social Networks users by using a Wireless Sensor Network based on sensor relationships, *Journal of Data Science and Pattern Recognition*, vol. 1, no. 1), pp. 1-12, 2017.

[10] W. Liu, W. J. Zhao, C. Li. Particle Filter Theory Framework and Its Application in Target Tracking, *Journal of Automation and instrumentation*,vol. 3, no. 3, pp. 190-191, 2016.

[11] Q. H. Ning, Y. B. Zhang, L. Liu, etc. Target Tracking Optimization Algorithm for Extended Kalman Filtering, *Journal of Detection and Control*, 2016, , no. 1, pp. 90-94.

[12] K. Gyorgy, A. Kelemen, L. David, Unscented Kalman Filters and Particle Filter Methods for Nonlinear State Estimation, *Journal of Pro-cedia Technology*, 2014, 12, no. 1), pp. 65-74.

[13] X. Y. Song, Z. H. Chen, X. Y. Sun. A Target Tracking Algorithm for Wireless Sensor Networks with Improved Particle Filter, *Journal of Automation and instrumentation*, 2016, 2, no. 6, pp. 170-172.

[14] E. S. Wang, T. Pang, P. P. Qu. An Improved Particle Swarm Optimization Algorithm for Particle Filter Based on Chaos, *Journal of Beijing University of Aeronautics and Astronautics*, 2016, 5, no. 11, pp. 885-890.

[15] S. Bourdarie, D. Lazaro, P. Nieminen, et al. In-depth Analysis of SREM Count Rate Measurements for Future Use in EnKF Salammb. Data Assimilation Tool , *Journal of IEEE ransactions on Nuclear Science* 2014, 61, no. 4), pp. 1679-1686.

[16] B. Fu. Research on the Optimization of Wireless Sensor Network Based on Artificial Fish School Algorithm, *Journal of Computer system application*, 2015, 12, no. 6, pp. 223-227.

[17] B. Gao, D. Hang, G. Z. Zhang. Distribution Network Reconfiguration Based on Artificial Fish Group and Particle Swarm Hybrid Algorithm, *Journal of Northeast Dianli University*, 2012, , no. 6, pp. 10-13.