IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 7, 2020, accepted July 5, 2020, date of publication July 10, 2020, date of current version July 22, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3008373*

# Target Positioning Based on Particle Centroid Drift in Large-Scale WSNs

**ZHENGWAN ZHANG[1], CHUNJIONG ZHANG[2], MINGYONG LI[3], (Member, IEEE), AND TAO XIE[4]**

[1]College of Online and Continuous Education, Southwest University, Chongqing 400715, China
[2]Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
[3]School of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China
[4]Institute of Education, Southwest University, Chongqing 400715, China

Corresponding author: Mingyong Li (limingyong@cqnu.edu.cn)

**ABSTRACT** The localization problem of target nodes remains unresolved, especially in large-scale and complex environments. In this paper, we propose a particle centroid drift (PCD) algorithm to reduce the distance errors between nodes and obtain the particle aggregation region by using the drift vector. First, we use the particle quality prediction function to obtain the particles in a high-likelihood region. The high-quality particles have high probability in the calculation, which can increase the number of effective particles and enable avoiding particle degradation. Then, the centroid drift vector is used to make the particle distribution similar to the actual reference distribution. Experiments are conducted on state-space models: the local movement where 55% nodes are moving and the globe movement where 100% nodes are moving. The results show that the proposed algorithm has low estimation errors, a good tracking effect and an acceptable time complexity.

**INDEX TERMS** Centroid drift, node positioning, particle filter, wireless sensor networks.

## I. INTRODUCTION

There has been fruitful progress in the wireless sensor networks(WSNs) domain during the last decade, but the localization and tracking of target nodes still remain unresolved, especially in large-scale and complex environments, such as those in ecological monitoring and a military battlefield [1], [2].

Compared to the traditional static network, dynamic nodes are more difficult to monitor due to frequent node movement [3]. Range-based positioning algorithms, e.g., Time of Arrival(TOA) algorithm [4], Time Difference of Arrival(TDOA) algorithm [5], Angle of Arrival(AOA) algorithm [6], and Received Signal Strength(RSSI) algorithm [7], try to obtain the measurement distance and utilize the geometric information to locate unknown nodes based on RF signals. However, the attenuation and refraction in the signal propagation process between nodes result in poor positioning accuracy and do not meet the actual application requirements [7]–[9]. Analogously, range-free positioning algorithms that rely on the connectivity between nodes are also difficult to use in a large-scale complex environment [5]–[7] because a high node density and a large communication overhead are required. Typical examples of range-free algorithms include Amorphous [8], Distance Vector-Hop(DV-Hop) [9], Multi Dimentional Scaling Map(MDS-MAP) [10], Distance Vector-short(DV-short) [11] and Approximate Point in Triangulation Test(APTT) [12].

Moreover, many scholars have proposed using the connectiveness between nodes to determine a target's current movement position and predict its next position. Bayesian-estimation-based filtering algorithms such as the Particle Filter(PF) [13], Extended Kalman Filter(EKF) [14], Extended Kalman Particle Filter(EKPF) [15], Unscented Kalman Filter(UKF) [16] and Hybrid Iterated Kalman Particle Filter(HIKPF) [17] are representative of this scenario. Specifically, the PF can carry out the position estimation of target nodes in nonlinear and non-Gaussian WSNs [18]–[20], but it depends on the initial state of the sensor nodes and cannot accurately measure the Euclidean distance between nodes [21], [22]. In addition, the particle degradation problem of the PF, in which the weights of particles increase after a series of Bayesian iterations, has significant influences on the positioning accuracy.

To address these problems, we propose a particle centroid drift (PCD) algorithm to reduce the distance errors between

The associate editor coordinating the review of this manuscript and approving it for publication was Lorenzo Mucchi.

nodes without depending on the Euclidean distance. We calculate the prior distribution parameters of the PF and obtain the particle aggregation region by using the drift vector of the particle center. To do this, we first use the particle quality prediction (PQP) function [23] to obtain the particles in the high-likelihood region. The high-quality particles participate in the subsequent calculation with high probability, which can increase the number of effective particles and enable avoiding the degradation of particles. Then, the centroid drift vector is used to drift the predicted particles to make the particle distribution similar to the actual reference distribution.

In summary, this study has the following main contributions:

- We propose to use the motion of the particle's centroid to increase the positioning accuracy. The target positioning and tracking are transformed into a Bayesian estimation problem and the centroid drift vectorization is included in particle motion function.
- We combine the centroid algorithm and the particle distribution function so that high-quality particles can be obtained for subsequent position estimation. The moment equation is adopted to avoid the complex matrix calculation which turns out to be effective in terms of tracking and convergence.
- A series of experimental evidences are provided based on two different state-space models. Compared to the public benchmark algorithms, results the PCD produced have shown better performance in terms of estimation errors and tracking effects.

The remainder of this article is organized as follows. Section II summarizes the related work. Section III provides details of the proposed algorithm. Section IV validates our approach through experimental simulations. Section 5 draws some conclusions.

## II. RELATED WORK

Many filtering algorithms can improve the estimation accuracy in dynamic networks. The classic filter algorithms in terms of WSN positioning include the Wiener filter, least square filter, Kalman filter and its extensions (e.g., EKF and UKF), the PF, etc. [22], [24].

Typically, the PF algorithm is used to handle nonlinear and non-Gaussian filtering problems when the target nodes are deployed in large-scale and complex environments. Pak *et al.* [25] proposed a distributed and PF-based range-free algorithm to alleviate the transmission interruption of indoor wireless signals. Ahmed *et al.* [13] studied the accuracy in tracking moving objects and calculated the time of object motion. Suggestions for selecting appropriate time parameters were put forward to provide a reasonable trade-off between accuracy and complexity. Sve čko *et al.* [11] presented the use of the PF in multiple hops between nodes for distance estimation and connected it with the source node using the radio frequency of the received signal strength indicator (RSSI). An improved PF for multitarget tracking was proposed in [26], which was grounded on the drift theory of the stochastic differential equation. The drift theory was used to design a Monte Carlo procedure in a Markov chain, causing the particle samples to be similar to observations. The particle swarm optimization (PSO) proposed in [12] measures the distance between nodes, and the measurement error was contained in the observation function of the motion noise. As a result, the positions of objects were determined by a weighted aggregation and by maximizing the PSO. Hou *et al.* [27] proposed an improved cubature PF algorithm that uses the least squares to estimate the prior states of target nodes. The cubature Kalman filter and Gauss-Newton iterative methods were used to measure the posterior state. This work provided the suggested distribution for the PF and improved the positioning accuracy. Vázquez and Míguez [28] introduced a distributed PF algorithm based on the posterior probability and combined it with Bayesian estimation. Experimentally, this method is superior to the distributed PF in terms of the positioning accuracy and robustness.

In summary, most existing works combined the range-based and range-free methods and used additional auxiliary filters to improve the positioning accuracy. However, they did not consider the particle degradation problem and ignored the influence of various intricate environmental factors (such as the number of deployed nodes, the mobility, and the path loss factor) and PF parameters (such as the number of generated particles, the noise variation and the particle population mobility) on the positioning accuracy of the target node. In addition, they were limited by their computational complexity and convergence speed, hindering them from being applicable in a large-scale and complex environment.

## III. PROPOSED ALGORITHM

The target node periodically broadcasts beacon packets to neighboring nodes, and each beacon packet contains its identification number and location coordinates. When the number of beacon packets exceeds a specified threshold, the node is determined to be the centroid of a polygon. Correspondingly, the centroids of its neighboring nodes are viewed as its estimation.

Although the traditional PF algorithm can perform node position estimation in non-linear and non-Gaussian WSNs, it has the particle degradation problem which means particles in the higher likelihood region have bigger weights and particles that do not intersect with the prior distribution region have weights close to 0. This problem can be well addressed by predicting the posterior distribution of high-quality particle populations. PQP algorithm can be applied in this respect.

Because nodes in large-scale WSNs are movable, we must consider the distances between them when they are moving. However, frequent calculation of the distances leads to high time complexity and more importantly influences the positioning accuracy significantly. The centroid algorithm is range-free and does not depend on Euclidean distance, which

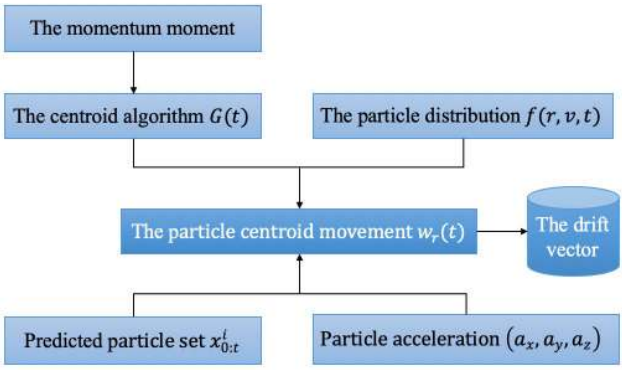



**FIGURE 1.** Algorithm framework.

can estimate the node positions by using observable moving nodes nearby.

When the nodes move to complex boundaries where the coverages of nodes are coupled and crossed, the centroid solution is very time-consuming. In order to reduce the complexity of the algorithm, we apply the momentum equation to represent the acceleration and velocity of the particles to avoid complex matrix calculations.

Fig. 1 illustrates the framework of the proposed algorithm. The centroid algorithm is to calculate the centroid of the particle population [29] by applying the momentum moment theorem [14], [25], [30]. Combining the centroid with the average particle velocity, the scalar representation is transformed into the momentum. Considering the real conditions, we accommodate the particle acceleration in the framework. The high-quality particle set is obtained according to the nonlinear replication algorithm [23], and the predicted particle set $\left\{x_{0:t}^i : i = 1, 2, \ldots, N\right\}$ is obtained by the PQP algorithm. The weighted centroid drift and the centroid of the predicted particle set are calculated when the centroid velocity solution is applied to the predicted particle set. Finally, we obtain the centroid drift vector by calculating the centroid motion equation, the centroid of the predicted particle set and the weighted centroid drift.

### A. PARTICLE CENTROID MOVEMENT

Considering the increasing complexity of a solution when the node movement is involved in the determination of the centroid resolution, we apply the momentum moment theorem (moment equation) [14], [25], [30] to optimize the complex matrix inversion calculation and training process.

To determine the motion state of a particle, we define its position in structural space and the velocity space. According to the rules of a six-dimensional phase space [30], $(r, v) = (x, y, z, v_x, v_y, v_z)$ is used to represent the phase space, where $x, y, z$ are the directions. Then, the average $w_v$ of all particle velocities in the x direction can be denoted by $f(r, v, t)$, which contains the particle velocity distribution and time, as shown in equation (1) [31]. Similarly, the motion equations in the y and z directions can be obtained in the same way, and we omit them for simplicity. The motion equation of the particle population centroid $G(t)$ can be expressed by the drift velocity and the function $f(r, v, t)$ containing the particle velocity distribution and time, as shown in (2):

$$w_r(t) = \frac{d}{dt}G(t) = \frac{d}{dt}\frac{\int_{r,v} xf(r, v, t)\, drdv}{\int_{r,v} f(r, v, t)\, drdv} \tag{1}$$

$$\omega_v(t) = \frac{\int_{r,v} vxf(r, v, t)drdv}{\int_{r,v} f(r, v, t)drdv} \tag{2}$$

where the centroid $G(t)$ at moment $t$ is obtained by the centroid algorithm [29]. The centroid drift velocity of the particle population can be obtained by CV/CT analysis of the average particle velocity [20], [26].

We use the differential equation to transform the equation (1), i.e.

$$w_r(t) = \frac{\int_{r,v} x\left(\partial/\partial t\right) f(r, v, t)drdv}{\int_{r,v} f(r, v, t)drdv} - G(t)\frac{\int_{r,v}\left(\partial/\partial t\right) f(r, v, t)drdv}{\int_{r,v} f(r, v, t)drdv} \tag{3}$$

The proof of equation(3) is obvious by using the differential equation. The partial differential term at time $t$ in equation (3) represents the movement of particles in the phase space.

According to the Boltzmann equation [32], we have:

$$\partial f(r, v, t)/\partial t = \{v\cdot(\partial/\partial r) + \alpha\cdot(\partial/\partial v) + (\partial/\partial t)_{coll}\} f(r, v, t) \tag{4}$$

where $(\frac{\partial}{\partial t})_{coll}$ is the change in the particle distribution function caused by the signal attenuation between nodes and $a = (a_x, a_y, a_z) = (\frac{e}{m}, 0, 0)$ is the acceleration factor. The drift speed of the particle population centroid is obtained by:

$$w_r(t) = w_v(t) + \frac{\int_{r,v}(x - G(t))(\frac{\partial}{\partial t})_{coll} f(r, v, t)\, drdv}{\int_{r,v} f(r, v, t)\, drdv} \tag{5}$$

Considering the y and z directions of the phase space, the actual drift velocity of the particle population centroid can be represented as $w_r(t, x, y, z)$. Obviously, equation (5) is a continuous equation with respect to the particle density [33]. In addition to the average velocity of particles, the entire particle population has another velocity component (i.e., deflection trend), which is caused by the deflection of the particle position outside the range of the particle centroid.

As the number of particles changes, the particle centroid will move forward, and the centroid velocity will exceed the particle velocity. Because faster particles have heavier weight, it is possible to predict the movement directions. However, we are bound to calculate the particle weight and the spatial distribution, which is very time consuming.

To reduce the complexity of the algorithm, we convert the velocity scalar into a vector to obtain $w_r$ and use the moment equation of the Boltzmann equation to address the factors under complex and real conditions.

*Proof 1:* Taking (4) into (3), we get:

$$\omega_r(t) = \frac{\int_{r,v} x\left(\partial/\partial t\right) f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv} - G(t)\frac{\int_{r,v}\{v\cdot(\partial/\partial r)+\alpha\cdot(\partial/\partial v)+(\partial/\partial t)_{coll}\}f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv}$$

Because $\alpha=\left(\alpha_x,\alpha_y,\alpha_z\right)=(e/m,0,0)$, the partial derivative of $\alpha=0$, i.e. $\alpha(\partial/\partial v)$=0, Then, we have:

$$\begin{aligned}\omega_r(t) = \omega_r(t) &= \frac{\int_{r,v} x\left(\partial/\partial t\right) f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv} - G(t)\frac{\int_{r,v}\{v\cdot(\partial/\partial r)+(\partial/\partial t)_{coll}\}f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv}\\ &= \frac{\int_{r,v} x\{v\cdot(\partial/\partial r)+(\partial/\partial t)_{coll}\}f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv} - G(t)\frac{\int_{r,v}\{v\cdot(\partial/\partial r)+(\partial/\partial t)_{coll}\}f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv}\\ &= \frac{\int_{r,v} xv\cdot(\partial/\partial r)f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv} + \frac{\int_{r,v} x(\partial/\partial t)_{coll}f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv} - G(t)\frac{\int_{r,v}\{v\cdot(\partial/\partial r)+(\partial/\partial t)_{coll}\}f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv}\end{aligned}$$

When the particle distribution function changes, $G(t)\frac{\int_{r,v} v\cdot(\partial/\partial r)f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv}=0$ and the equation can be rewritten as:

$$\begin{aligned}\omega_r(t) &= \frac{\int_{r,v} vxf(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv} + \frac{\int_{r,v} x(\partial/\partial t)_{coll}f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv} - \frac{\int_{r,v} G(t)(\partial/\partial t)_{coll}f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv}\\ &= \frac{\int_{r,v} vxf(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv} + \frac{\int_{r,v}\{x-G(t)\}(\partial/\partial t)_{coll}f(r,v,t)drdv}{\int_{r,v} f(r,v,t)drdv}\end{aligned}$$

Therefore, the equation (5) is hold.

### B. MOMENT EQUATION SOLUTION

The momentum moment equation refers to the relationship among the momentum of the external force on a moving node, the momentum of the speed, and the momentum against the rate of time change [14], [25], [30]. Therefore, we obtain the population centroid containing the trend and speed of the centroid movement.

The centroid $G(t)$ of the particle population is represented as the quotient of $Ne$ and $M_x$ [34] in equation (6).

$$G(t) = \frac{M_x(t)}{Ne(t)} \tag{6}$$

$$Ne(t) = \int_{r,v} f(r,v,t)\,drdv = \int_v f(v,t)\,dv \tag{7}$$

$$M_x(t) = \int_{r,v} xf(r,v,t)\,drdv = \int_v M_x(v,t)\,dv \tag{8}$$

$Ne$ and $M_x$ are, respectively, the first-order moment related to the movement along the $x$ direction and the zero-order moment of the particle distribution in equation [35].

The partial derivatives of $Ne$ and $M_x$ with respect to $t$ are obtained by (9) and (10), respectively.

$$\frac{\partial f(v,t)}{\partial t} = -ax\frac{\partial}{\partial xv}f(v,t) + (\frac{\partial}{\partial t})_{coll}f(v,t) \tag{9}$$

$$\frac{\partial M_x(v,t)}{\partial t} = vxf(v,t) - ax(\frac{\partial}{\partial xv})M_x(v,t) + (\frac{\partial}{\partial t})_{coll}M_x(v,t) \tag{10}$$

To reduce the number of calculation parameters, the integrated equation can be rewritten as equation (11):

$$(\frac{\partial}{\partial t})_{coll} = \sum_k \left\{-Nq_k(v)|v| + \int_{v'} Nq_k(v')|v'|P_k(v',v)\,dv'\right\} \tag{11}$$

where $N$ is the particle density, $q$ is the coupling area of particles, $P_k(v',v)$ is the Gaussian distribution of particles, with the velocity $v'$ becoming $v$ when the movement direction changes, and $k$ is a coupling type, representing the motion state under different scenes.

### C. PARTICLE CENTROID DRIFT

According to [23], applying the PF to the target node may significantly affect the positioning accuracy. Particles in the high-likelihood region have large weights, and those that do not intersect with the prior distribution region have weights that are approximately equal to zero. As a result, the particle degradation problem occurs. When the observed likelihood distribution does not intersect with the prior distribution, the PF algorithm will degrade and affect the system stability.

The posterior distribution of particles is the optimal distribution of the PF, but the integral of the particle weight cannot be solved [16], [26]. Scholars frequently use a transfer probability function as a reference distribution for the particle prediction. The limitation is that observations at the current time in the reference distribution are not available. To address this problem, the PQP algorithm was proposed in [23], which presents a predictor to obtain a particle swarm of high quality and modify the reference distribution function by calculating nonlinear variations ranging between particles and specify flags.

In this work, we use PQP to calculate the predicted particle set $\left\{x_{0:t}^i : i=1,2,\ldots,N\right\}$ and use the centroid drift vector in the particle set to track the target particle. The centroid drift vector is calculated by predicting the difference between the centroid of the particle set and the weighted centroid.

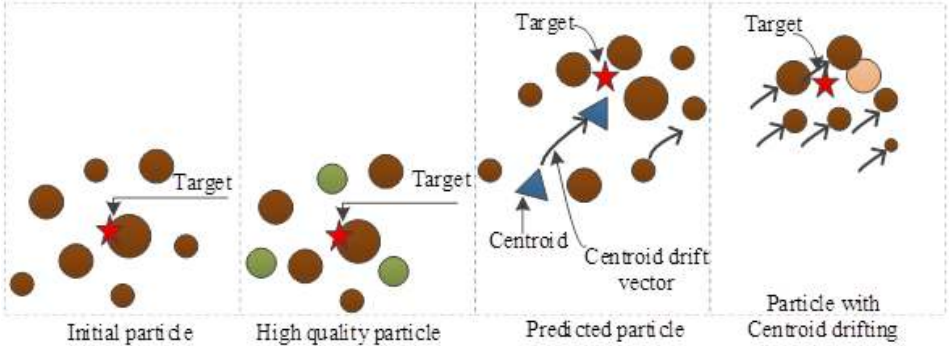



**FIGURE 2.** An example of particle centroid drift.

The weighted particle centroid $\overline{w_r(t)}$ is computed by (12):

$$\overline{w_r(t)} = \sum_{i=1}^{N} w_r(t) x_{0:t}^i + \frac{dNe(t)}{dt} + \frac{dM_x(t)}{dt} \tag{12}$$

where $w_r(t)$ is the centroid motion equation. Then, the centroid drift vector is as given in equation (13):

$$(w_r(t))_{shift} = (\overline{w_r(t-1,x,y,z)} - w_r(t-1,x,y,z), \overline{w_r(t,x,y,z)} - w_r(t,x,y,z)) \tag{13}$$

The particles drift with the guidance of the particle population centroid drift vector. Then, we have:

$$\overline{x_{0:t}^i} = \begin{cases} x_{0:t}^i + (w_r(t))_{shift}, & \text{if } rand\ N < min(1, \dfrac{\overline{w_r(t)}}{w_r(t)}) \\ w_r(t), & \text{else} \end{cases} \tag{14}$$

where *randN* is a random value in [0, 1]. Given a weighted particle set $\left\{\overline{x_{0:t}^i} | i = 1, \ldots, N\right\}$, its weighted centroid is the unbiased estimation of the current state. Then, the particle is resampled to update the particle weight.

Overall, the implementation of the proposed algorithm is as follows:

**Step 1**: Predict the particle filter distribution based on the transfer function $p(x_t^i | x_{t-1}^i)$.

**Step 2**: Calculate the centroid drift vector using equations (13) and (14) and $(w_r(t))_{shift} = \overline{w_r(t)} - w_r(t)$.

**Step 3**: Estimate the location of the target node according to the particle selection probability $p(x_t) = \frac{1}{(2\pi)^{(\frac{m}{2})}} exp[-\frac{1}{2}(z_t - \mu_k)^T s_k^{-1}((z_t - \mu_k)]$.

**Step 4**: Resample the particles and update the particle weights; then, return to Step 3.

Step 1 is performed only once, while steps 3 and 4 are performed repeatedly. Step 2 is carried out to update the set after particle drift and obtain the posterior distribution $p(x_t|z_t)$ of the target particle.

An example of this algorithm is illustrated in Fig. 2. The circle represents the particle, the size of the circle represents the weight of the particle, and the triangle represents the particle population centroid. The algorithm predicts the particle distribution first and then moves the particles to the high-likelihood region according to the centroid drift vector. The observation of the current moment determines the position of the particle by the transfer function, causing the particle distribution to be more similar to the real reference distribution. To be more specific, the algorithm performs the following stages. First, the PCD searches the degraded particles (i.e. green circles) and removes them from the initial particle population using step 1. We obtain a set of high-quality particles. When the high-quality particles are moving, more high-quality particles are produced by a non-linear replication mechanism in PQP algorithm. Then, we obtain the centroids of particles (i.e. blue triangles) and corresponding drift vectors (i.e. black arrows) by using step 2. The step 3 predicts the particle distribution and then the particles move to the high-likelihood region according to the centroid drift vector. Finally, the step 4 updates particle weights (i.e. the pink circle) by a resampling process so that the position of the target node is optimal.

## IV. EXPERIMENT

In this section, a series of experiments on target node tracking are conducted. We compare public benchmark algorithms (PF [13], EKF [14], EKPF [15] and HIKPF [17]) and the proposed PCD algorithm in terms of the estimation errors, tracking effects and performance.

### A. ESTIMATION ERRORS

To analyze the estimation errors, we use two state-space models: the local movement where 55% nodes are moving and the globe movement where 100% nodes are moving.

The state-space model of local movement is:

$$x_k = 1 + \sin(0.04\pi(k-1)) + 0.5x_{k-1} + v_{k-1} \tag{15}$$

$$z_k = \begin{cases} 0.23x_k^2 + \mu_k, & k \le 30 \\ 0.51x_k - 2 + \mu_k, & k > 30 \end{cases} \tag{16}$$

The state-space model of globe movement is:

$$x_k = 1 + \sin(0.04\pi(k-1)) - \frac{\sin(x_k)}{4} + 0.5x_{k-1} + v_{k-1} \tag{17}$$

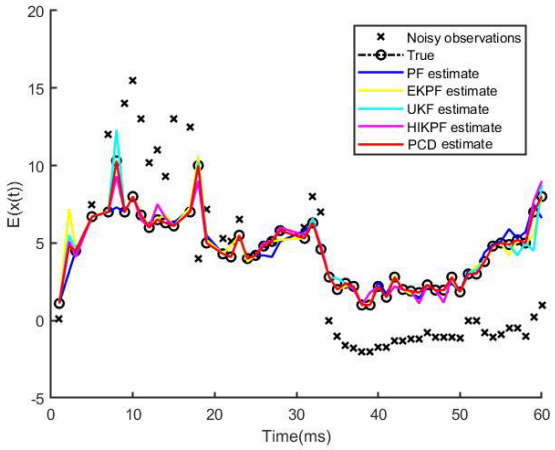



**FIGURE 3.** Expectation curves (local movement).

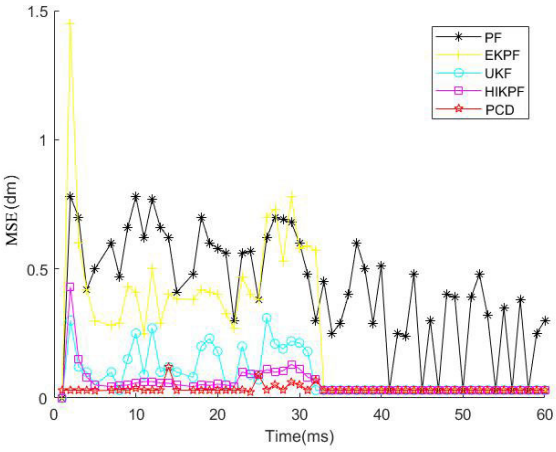



**FIGURE 4.** MSE curves (local movement).

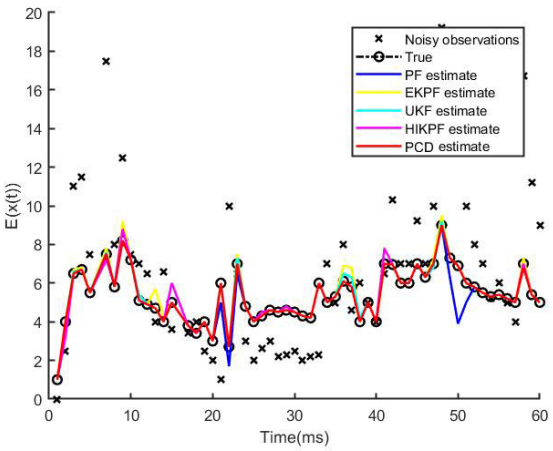



**FIGURE 5.** Expectation curves (globe movement).

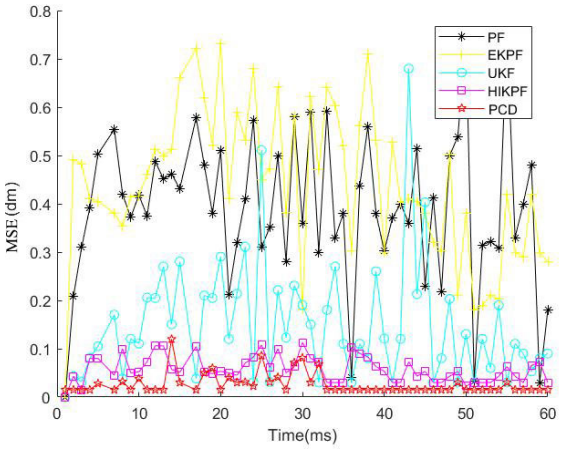



**FIGURE 6.** MSE curves (globe movement).

$$z_k = 0.23x_k^2 + 0.51x_k + \frac{\sin(x_k)}{5} - 2 + \mu_k \tag{18}$$

where $v_k$ is subject to the Gamma distribution $\Gamma(\alpha, 2)$ and the measurement noise $\mu_k$ is subject to the Gaussian distribution $N(0, 0.001)$. Here, the mean of 0 is commonly used when data are drawn from the standard normal distribution and the variance of 0.001 is used for the simulation of data dispersion in real-world environments by assuming a more centralized distribution than the standard normal distribution. The number of particles $N = 200$, and the observation time $t = 60$. We run 100 independent experiments. The mean of the particle set is calculated as:

$$\hat{x} = \frac{1}{N}\sum_{j=1}^{N} x_k^j \tag{19}$$

We adopt the mean square error (MSE) for the noise measurement:

$$mse = \frac{1}{t}\sum_{i=1}^{t}(\hat{x}_k^i - x_k^i)^2 \tag{20}$$

It is worth noting that we conduct experiments by selecting parameters in large-scale and complex environments. Basically, an environment which is large has more than 100 WSN nodes according to literature [9], [22]. Therefore, the number of nodes in this experiment is set between 100 and 1000 to observe the performance of the algorithms. Besides, the network delay parameter and complex node speed can also simulate large-scale scenarios which are usually used in dynamic mountain monitoring and pollution source tracking.

Fig. 3 and Fig. 4 are drawn using the 55% local movement model. Fig. 3 illustrates the expectation values between the different algorithms. They show that the estimates of the PF, UKF and EKPF tend to deviate from the actual position and that the estimates of the HIKPF and PCD are basically consistent with the real positions. Generally, the MSE of PCD is lower than that of others. Fig. 4 shows changes in the MSEs with time. The PCD has the smallest MSE, followed by the HIKPF, UKF, EKPF and PF, which means that the PCD is not likely to be influenced by the running time. It is much more stable and optimal with respect to errors.

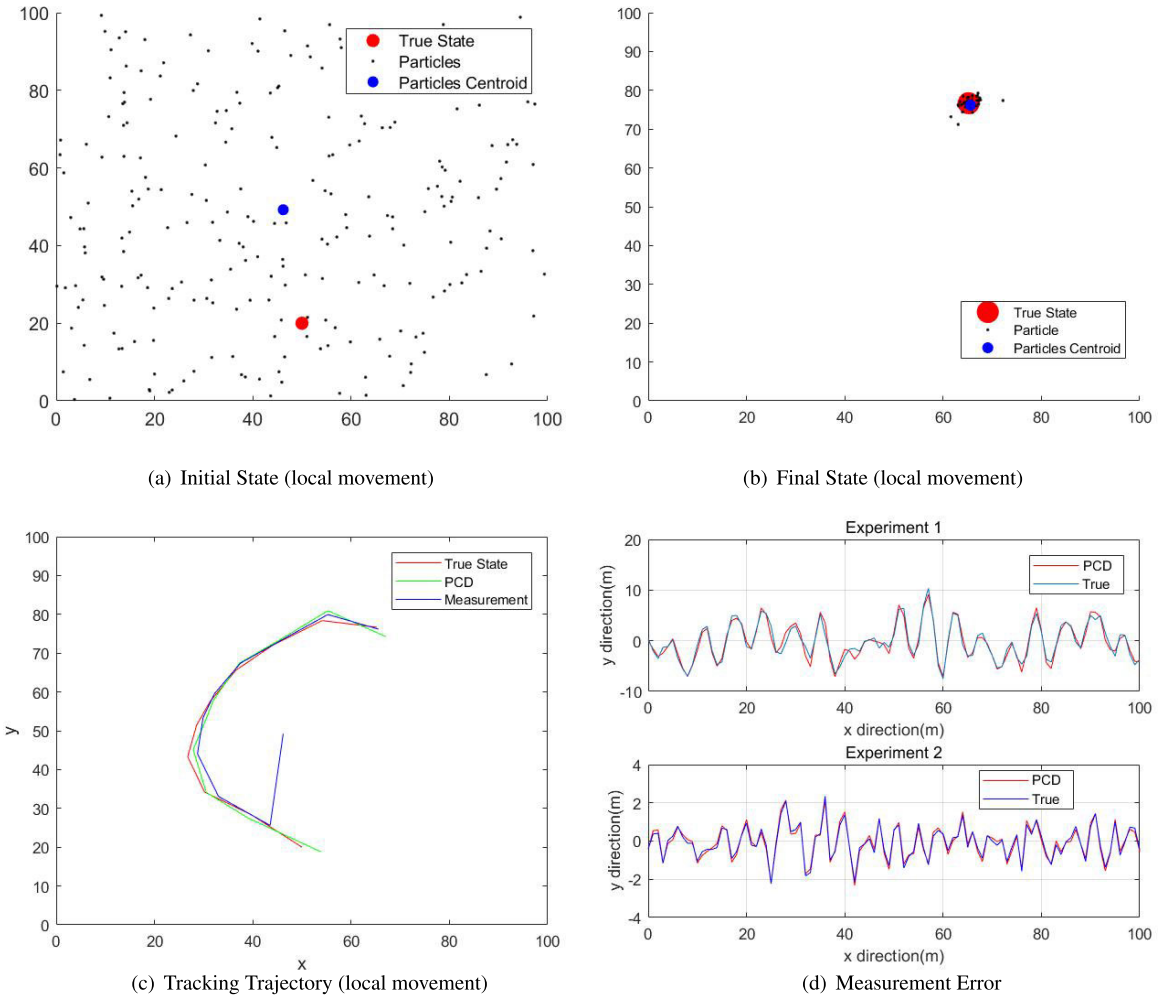



**FIGURE 7.** Tracking effect of PCD.

Fig. 5 and Fig. 6 are drawn using the globe movement model. Fig. 5 shows the differences between the estimated expectation of the positioning state and the real state. The EKPF has the worst adaptability to the localization of target nodes in the WSN. The positioning mean square error curve generated by each filter is shown in Fig. 6. It also demonstrates the superiority of the PCD algorithm in positioning estimation.

### B. TRACKING EFFECT

To analyze the tracking effect, we deployed 500 nodes randomly in a rectangular area of $100 * 100\ m$. The target node moves at a speed of 0.5 $m/s$, pausing for 2 seconds after every interval of 20 seconds [36], [37]. Again, we use the state-space models presented in Section IV-A.

The initial position state of the centroid shown in Fig. 8(a) is different from the true state. However, they overlap at the final state in Fig. 8(b). Fig. 8(c) presents the movement trajectory of the target particles, showing that the result of the PCD algorithm is close to the true movement state and is better than the measurement trajectory. Fig. 8(d) shows the acceptable measurement error between the PCD and the real state.

### C. ALGORITHM PERFORMANCE

In this subsection, we evaluate the algorithm performance by changing the parameters.

Fig. 8 shows the impact of motion noise on the MSE. An increase in motion noise leads to a larger MSE. Overall, the PCD outperforms the other algorithms in terms of the average MSE.

Fig. 9 shows the impact of observation noise on the MSE. The MSE increases as the observation noise increases. The PF has higher but flatter MSE values than those of the other algorithms. The EKPF, UKF and HIKPF have almost comparable values in the curves, which increase dramatically as the observation noise increases. The PCD proposed in this study has the lowest MSE on average.

Fig. 10 shows the influence of nodes on the MSE. Obviously, the MSEs decrease as the number of nodes increases. The PCD is superior to the others, especially to the PF, likely because the PF algorithm has the particle degradation

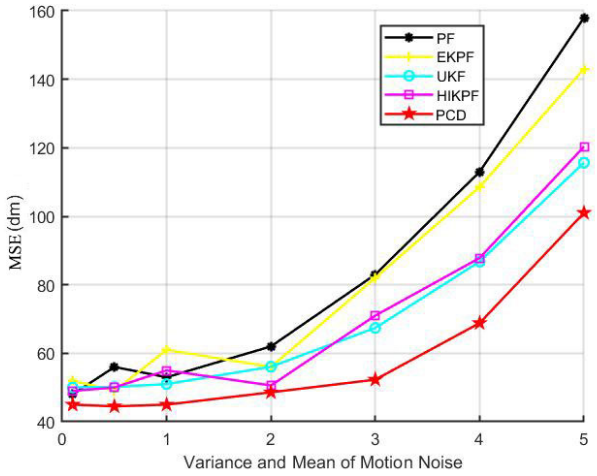



**FIGURE 8.** Impact of motion noise on the MSE (local movement).

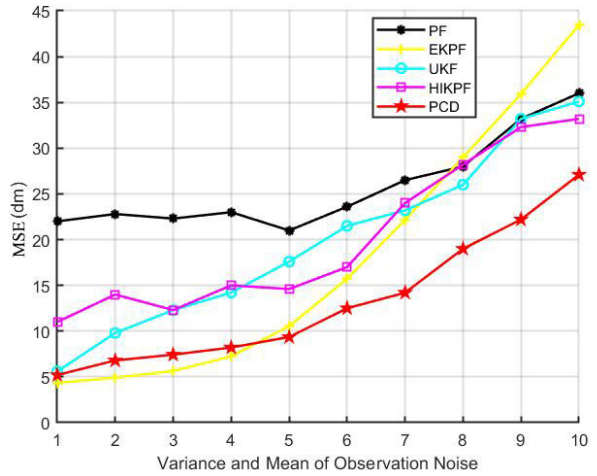



**FIGURE 9.** Impact of observation noise on the MSE (local movement).

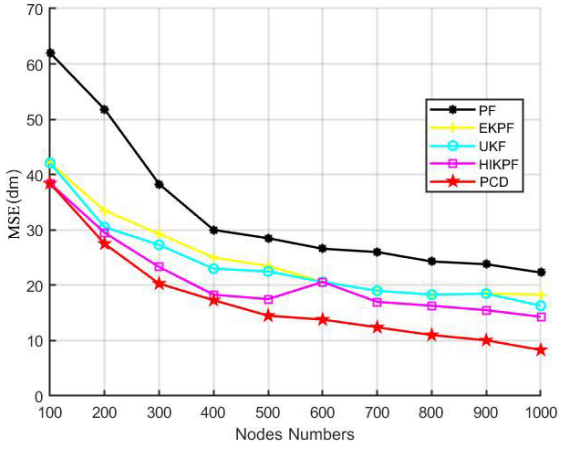



**FIGURE 10.** Impact of the node number on the MSE (local movement).

phenomenon, leading to a dramatic increase in errors. When the observation noise is small, the UKF and PQP significantly outperform the PF. As the observation noise increases,

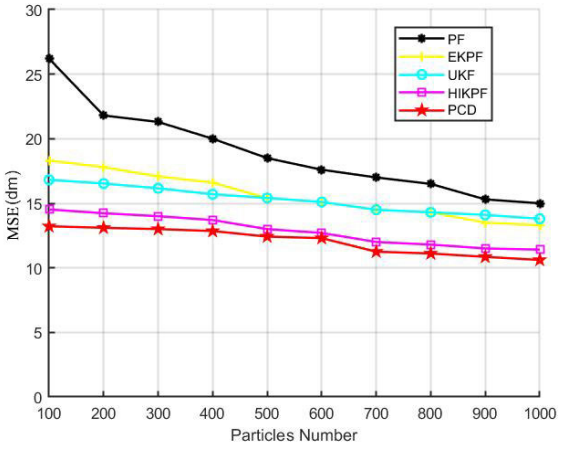



**FIGURE 11.** Impact of the particle number on the MSE (local movement).

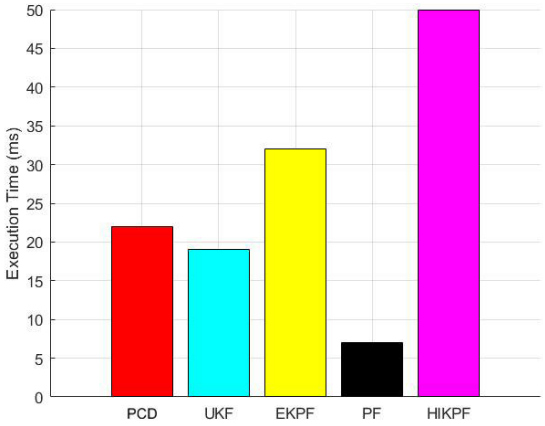



**FIGURE 12.** Execution time across algorithms (local movement).

the quality prediction function can effectively reduce the degradation of the PF.

Fig. 11 shows the influence of the particle number on the MSE. The PCD achieves the lowest MSEs on average as the number of particles increases. The EKPF and UKF algorithms have similar performances. When the number of particles is greater than 100, the performance essentially becomes stable. The PF improves its performance gradually with the increase in particle number, but it still performs poorer than the others. The PCD can achieve the specified accuracy given a small number of particles.

### D. TIME COMPLEXITY

The time complexity tests the required time for an algorithm to produce an estimate. Herein, we consider the time spent to carry out sampling, importance sampling, and resampling rather than the signal communication. Fig. 12 shows the average time complexity based on duplicate configurations. The PF requires the least amount of time, while the HIKPF requires the most. The PCD requires nearly the same amount of time as required by the UKF and much less than that of the EKPF. The time complexity of PCD is not optimal

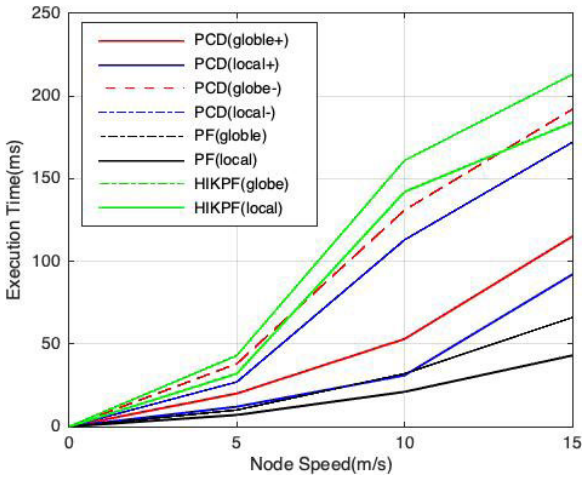



**FIGURE 13.** Execution time versus node speed. "+" denotes the algorithm with momentum moment, and "−" denotes the algorithm without momentum moment.

because it requires calculation of the particle quality, matrix multiplication, the weights in PQP and the centroid drift vector in the particle population. In addition, the state model can also influence the processing time.

Simulations of the execution time versus node speed are shown in Fig. 13. As the speed increases, the execution time increases. Totally speaking, the PCD algorithm runs more fast than other algorithms. This can be attributed to the use of momentum moment which simplifies the complex matrix calculation. To further verify it empirically, we run PCD algorithm with and without momentum moment. It shows PCD algorithm with momentum moment has less execution time than that without momentum moment. The time of PCD algorithm is mainly consumed by sampling and resampling of high-quality particles, which has O(nlogn) time complexity.

### E. OVERALL MOBILITY NETWORK

In this subsection, we evaluate the performance of algorithms in highly dynamic environments, such as those in water pollution source tracing and air quality monitoring. We use the random path mobility model [27], [28], [33], [36], which provides random node movement in many mobile WSN applications. The model includes random pauses along the $x$, $y$, and $z$ directions and speed changes [27], [36].

The network delay parameter is set to 3 s according to [13], [25], [36], [38]. The simulation time is 300 s. We consider the RF radius as a fixed value and evaluate the performances of various algorithms at a node density of 100 and speeds of 5, 10 and 15 m/s. These speeds are selected based on some practical applications, such as indoor positioning (low speed), dynamic mountain monitoring (medium speed) and pollution source tracking (high speed). The average of the 30 simulation tests is obtained.

Table 1 provides the number of target nodes tracked by the algorithms at different node speeds. As the moving speed increases, the number of target nodes decreases.

**TABLE 1.** Tracked nodes at different speeds.

| Velocity | Algorithms | | | | |
|---|---|---|---|---|---|
| | PF | UKF | EKPF | HIKPF | PCD |
| 5 m/s | 23 | 26 | 30 | 34 | **48** |
| 10 m/s | 16 | 19 | 25 | 31 | **41** |
| 15 m/s | 12 | 14 | 23 | 26 | **32** |

**TABLE 2.** Positioning error at different speeds (unit: m).

| Algorithm | Velocity | | |
|---|---|---|---|
| | 5 m/s | 10 m/s | 15 m/s |
| PF | 0.041927 | 0.079802 | 0.114526 |
| UKF | 0.026121 | 0.051457 | 0.092357 |
| EKPF | 0.025819 | 0.058313 | 0.093875 |
| HIKPF | 0.020825 | 0.048831 | 0.077243 |
| PCD | **0.018159** | **0.032357** | **0.048724** |

At the same speed level, the PF tracks less target nodes than the other algorithms, while the PCD tracks the most, e.g., we obtain 48 nodes at 5 m/s and 32 at 15 m/s.

Table 2 shows the positioning errors when the node speed changes. It is clear that the higher the moving speed is, the faster the centroid drift speed. Overall, the PCD has higher positioning accuracies than those of the other algorithms.

### F. DISCUSSION

In the experiments, we simulated the large-scale WSN environments by setting corresponding parameters. From the results presented, it is fairly clear that the accuracies of PF, UKF and EKPF algorithms are not satisfactory and the accuracies of HIKPF and PCD algorithms are basically equivalent. The superiority of PCD can be attributed to the following reasons. First, The PCD algorithm is designed for large-scale WSNs where the movement of nodes is considered. The movement leads to centroid drift and different speeds have significant impacts on the positioning accuracy. The centroid algorithm included in PCD is key to improve the accuracy of target positions which makes error changes in the independent experiment stable and convergent. Second, PCD algorithm addresses the problem of particle degradation, which filters and replicates high-quality particles while other algorithms do not do this. Third, execution time of PCD algorithm is mainly spent on the sampling and resampling process. It uses the moment function to avoid complex matrix calculation. Therefore, the time complexity is obviously less than HIKPF and EKPF algorithms. Although the PCD algorithm runs more slowly than UKF and PF algorithms, it has greater improvement of accuracy.

## V. CONCLUSIONS

In this paper, we proposed a particle centroid drift method to measure the centroid drift state of a particle population. The simulation results demonstrate that the algorithm is robust and accurate. The calculation time and energy

consumption are acceptable. Particle centroid drift can be applied to large-scale and complex environments.

## ACKNOWLEDGMENT

*(Zhengwan Zhang and Chunjiong Zhang contributed equally to this work.)*

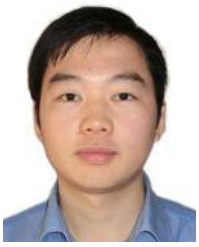
## REFERENCES

[1] C.-F. Cheng and C.-W. Wang, "The target-barrier coverage problem in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 5, pp. 1216–1232, May 2018.
[2] J. Kashniyal, S. Verma, and K. P. Singh, "A new patch and stitch algorithm for localization in wireless sensor networks," *Wireless Netw.*, vol. 25, no. 6, pp. 3251–3264, Aug. 2019.
[3] Y. Xiong, N. Wu, and H. Wang, "On the performance limits of cooperative localization in wireless sensor networks with strong sensor position uncertainty," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1613–1616, Jul. 2017.
[4] M. Ke, Y. Xu, A. Anpalagan, D. Liu, and Y. Zhang, "Distributed TOA-based positioning in wireless sensor networks: A potential game approach," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 316–319, Feb. 2018.
[5] W. Meng, L. Xie, and W. Xiao, "Optimal TDOA sensor-pair placement with uncertainty in source location," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9260–9271, Nov. 2016.
[6] K. Yu and Y. J. Guo, "Statistical NLOS identification based on AOA, TOA, and signal strength," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 274–286, Jan. 2009.
[7] N. A. M. Maung and M. Kawai, "Experimental evaluations of RSS threshold-based optimised DV-HOP localisation for wireless ad-hoc networks," *Electron. Lett.*, vol. 50, no. 17, pp. 1246–1248, Aug. 2014.
[8] S. S. Kumar, M. N. Kumar, and V. S. Sheeba, "Obstacle based range-free localization-error estimation for WSN," *Int. J. Comput. Sci. Issues*, vol. 8, no. 5, p. 31, 2011.
[9] T. L. T. Nguyen, F. Septier, H. Rajaona, G. W. Peters, I. Nevat, and Y. Delignon, "A Bayesian perspective on multiple source localization in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1684–1699, Apr. 2016.
[10] A. Javanmard and A. Montanari, "Localization from incomplete noisy distance measurements," *Found. Comput. Math.*, vol. 13, no. 3, pp. 297–345, 2011.
[11] J. Sve ko, M. Malajner, and D. Gleich, "Distance estimation using RSSI and particle filter," *ISA Trans.*, vol. 55, pp. 275–285, Mar. 2015.
[12] A. Keshavarz-Mohammadiyan and H. Khaloozadeh, "PSO-PF target tracking in range-based wireless sensor networks with distance-dependent measurement noise," in *Proc. 23rd Iranian Conf. Electr. Eng.*, May 2015, pp. 911–915.
[13] N. Ahmed, M. Rutten, T. Bessell, S. S. Kanhere, N. Gordon, and S. Jha, "Detection and tracking using Particle-Filter-Based wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 9, pp. 1332–1345, Sep. 2010.
[14] L. Zhi-Yu, W. Da-Ming, W. Jian-Hui, and W. Yue, "A tracking algorithm based on orthogonal cubature Kalman filter with TDOA and FDOA," *ACTA Phys. SINICA*, vol. 64, no. 15, 2015, Art. no. 150502.
[15] É. L. Souza, R. W. Pazzi, and E. F. Nakamura, "A prediction-based clustering algorithm for tracking targets in quantized areas for wireless sensor networks," *Wireless Netw.*, vol. 21, no. 7, pp. 2263–2278, Oct. 2015.
[16] R. Costanzi, F. Fanelli, E. Meli, A. Ridolfi, A. Caiti, and B. Allotta, "UKF-based navigation system for AUVs: Online experimental validation," *IEEE J. Ocean. Eng.*, vol. 44, no. 3, pp. 633–641, Jul. 2019.
[17] A. M. Nagy, A. Ahmed, and H. H. Zayed, "Hybrid iterated Kalman particle filter for object tracking problems," in *Proc. VISAPP*, vol. 2. New York, NY, USA: Citeseer, 2013, pp. 375–381.
[18] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, 2001.
[19] T. Xie, C. Zhang, Z. Zhang, and K. Yang, "Utilizing active sensor nodes in smart environments for optimal communication coverage," *IEEE Access*, vol. 7, pp. 11338–11348, 2019.
[20] A. Herrero García, "Localización de nodos dentro de una red inalámbrica de sensores," B.S. thesis, Univ. Politècnica de Catalunya, Barcelona, Spain, 2009.
[21] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Pers. Commun.*, vol. 7, no. 5, pp. 28–34, 2000.
[22] C.-H. Ou and K.-F. Ssu, "Sensor position determination with flying anchors in three-dimensional wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 9, pp. 1084–1097, Sep. 2008.
[23] C. Zhang, T. Xie, K. Yang, H. Ma, Y. Xie, Y. Xu, and P. Luo, "Positioning optimisation based on particle quality prediction in wireless sensor networks," *IET Netw.*, vol. 8, no. 2, pp. 107–113, 2018.
[24] I. Mabrouki and A. Belghith, "E-SeRLoc: An enhanced serloc localization algorithm with reduced computational complexity," in *Proc. 9th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jul. 2013, pp. 153–158.
[25] J. M. Pak, C. K. Ahn, P. Shi, Y. S. Shmaliy, and M. T. Lim, "Distributed hybrid particle/FIR filtering for mitigating NLOS effects in TOA-based localization using wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 5182–5191, Jun. 2017.
[26] V. Maroulas and P. Stinis, "Improved particle filters for multi-target tracking," *J. Comput. Phys.*, vol. 231, no. 2, pp. 602–611, Jan. 2012.
[27] J. J. Hou, M. Q. Zhu, and L. Ying, "An improved localization algorithm in wireless sensor networks inder complex background noise," *Trans. Beijing Inst. Technol*, vol. 36, no. 5, pp. 535–540, 2016.
[28] M. A. Vázquez and J. Míguez, "A robust scheme for distributed particle filtering in wireless sensors networks," *Signal Process.*, vol. 131, pp. 190–201, Feb. 2017.
[29] S. Phoemphon, C. So-In, and N. Leelathakul, "Fuzzy weighted centroid localization with virtual node approximation in wireless sensor networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4728–4752, Dec. 2018.
[30] G. W. Rathjens, N. K. Freeman, W. D. Gwinn, and K. S. Pitzer, "Infrared absorption spectra, structure and thermodynamic properties of Cyclobutane1," *J. Amer. Chem. Soc.*, vol. 75, no. 22, pp. 5634–5642, Nov. 1953.
[31] H. Sugawara, T. Yahata, A. Oda, and Y. Sakai, "The drift velocity vector of electron swarms in crossed electric and magnetic fields," *J. Phys. D, Appl. Phys.*, vol. 33, no. 10, pp. 1191–1196, May 2000.
[32] M. Bostan, "On the Boltzmann equation for charged particle beams under the effect of strong magnetic fields," *Discrete Continuous Dyn. Syst.-B*, vol. 20, no. 2, pp. 339–371, 2015.
[33] D. B. Licea, D. McLernon, and M. Ghogho, "Mobile robot path planners with memory for mobility diversity algorithms," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 419–431, Apr. 2017.
[34] M. M. Tsukamoto, L.-Y. Cheng, and F. K. Motezuki, "Fluid interface detection technique based on neighborhood particles centroid deviation (NPCD) for particle methods," *Int. J. Numer. Methods Fluids*, vol. 82, no. 3, pp. 148–168, Sep. 2016.
[35] A. J. Ballard, R. Das, S. Martiniani, D. Mehta, L. Sagun, J. D. Stevenson, and D. J. Wales, "Energy landscapes for machine learning," *Phys. Chem. Chem. Phys.*, vol. 19, no. 20, pp. 12585–12603, 2017.
[36] A. Rezaei, J. Gao, J. M. Phillips, and C. D. Tóth, "Improved bounds on information dissemination by manhattan random waypoint model," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.* New York, NY, USA: ACM, Nov. 2018, pp. 139–148.
[37] C. M. Albertsen, "Generalizing the first-difference correlated random walk for marine animal movement data," *Sci. Rep.*, vol. 9, no. 1, p. 4017, Dec. 2019.
[38] A. Oka and L. Lampe, "Distributed target tracking using signal strength measurements by a wireless sensor network," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1006–1015, Sep. 2010.

**ZHENGWAN ZHANG** received the B.E. degree in computer applications from Southwest Normal University, in 2004, and the M.S. degree in education technology from Southwest University, in 2009. His research interests include big education data, artificial intelligence, and wireless sensor networks.

**CHUNJIONG ZHANG** received the B.S. degree in electronic and information engineering from Shangqiu University, in 2015, and the M.S. degree in circuits and systems from Chongqing Three Gorges University, in 2018. He is currently pursuing the Ph.D. degree in electronic and information with Tongji University, China. His research interests include distributed artificial intelligence and wireless sensor networks.

**MINGYONG LI** (Member, IEEE) received the M.S. degree from Chongqing Normal University, Chongqing, China. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Donghua University. He is also an Associate Professor with the School of Computer and Information Science, Chongqing Normal University. His current research interests include deep learning, large-scale data retrieval, and image processing.



**TAO XIE** received the B.S. and M.S. degrees in educational technology from Southwest University, in 2007 and 2010, respectively, and the Ph.D. degree from the Department of Computer Science and Technology, Xi'an Jiaotong University, in 2017. He is currently an Associate Professor with Southwest University. His research interests include mobile Internet, data mining, and artificial intelligence.