Wireless sensor indoor positioning based on an improved particle filter algorithm

Pan Feng, Danyang Qin, Guangchao Xu, Ruolin Guo and Min Zhao

Abstract
Positioning by wireless sensor network is one of its main functions and has been widely used in many fields. However, when signal propagation is hindered, serious errors, non-line-of-sight errors, occur. In order to solve this problem, this article proposes an improved particle filter algorithm, which introduces the idea of residual analysis to improve reliability. The algorithm assigns weights to the particles based on the residuals and selects the appropriate particles. In addition, the non-line-of-sight error parameter $\alpha$ is introduced, and the second selection is made according to $\alpha$, which considers the influence of non-line-of-sight error. The non-line-of-sight error is greatly reduced after two selections. The simulation is performed under several different non-line-of-sight errors, and results show that the algorithm is superior to Kalman filter and particle filter.

Keywords
Indoor positioning, particle filter, wireless sensor, residual analysis, non-line-of-sight error

Date received: 8 June 2019; accepted: 6 January 2020

Handling Editor: Paolo Barsocchi

Introduction
Wireless sensor network (WSN) is a hot topic in recent years. By using sensors to collect data, it is easy to obtain various information of the monitored area. Positioning1 is one of its main functions and has been widely used in many fields. It uses two classes of nodes: fixed nodes and mobile nodes. The fixed nodes are distributed in the positioning area, the coordinates are known, and the mobile node is fixed on the positioning target.

Traditional methods have some errors in the actual situation: one is the measurement error, which is caused by system noise or measurement noise; the other is the non-line-of-sight (NLOS) error, which is the main challenge in positioning. It occurs when the signal propagation is hindered by some obstacles. In different scenarios, the NLOS error is subjected to different distribution functions. In order to minimize the impact of NLOS errors, some new algorithms have been proposed and they can be divided into two classes generally: one is to detect and select measurements which have no NLOS error. Shi et al.2 used channel visual information to improve positioning accuracy. A recurrent neural network was proposed in Choi et al.3; it proposed a recurrent neural network (RNN) model, which takes a series of channel state information (CSI) to identify the corresponding channel condition. Based on distributed filter, a new NLOS recognition algorithm was proposed in Pak et al.4; it proposes a new NLOS identification algorithm based on distributed filtering to mitigate NLOS effects, including localization failures. Momtaz et al.5 proposed a new algorithm
based on subspace method to identify and eliminate NLOS errors.

The other is weighting data with unknown propagation status. Tomic et al.\textsuperscript{6} proposed a positioning method using received signal strength (RSS) and angle of arrival (AOA). The positioning problem was transformed into the framework of a generalized trust region sub-problem in Tomic and Beko.\textsuperscript{7} Abu-Shaban et al.\textsuperscript{8} proposed a closed estimator for locating mobile stations (MSs) through three base stations (BSs) in a cellular network. There was a two-stage closed-form estimator to localize an MS by three BSs in cellular networks. It used a distance-dependent bias model to derive a range estimator as a first step. Then it used trilateration to find an estimate of the MS position. Ding and Qi\textsuperscript{9} proposed a new convex optimization model. Wang et al.\textsuperscript{10} proposed a deviceless wireless sensing system based on multi-domain features. Wang et al.\textsuperscript{11} developed a new robust optimization method source location.

However, the first method will loose a lot of measurement data, resulting in lower positioning accuracy. The second method does not consider the signal propagation state, the weight does not reflect the actual situation well, and the adaptability of the algorithm is not strong. The article proposes an improved particle filter (IMPF) algorithm to solve these problems. The main innovation of the article lies in two points: it reduces the influence of NLOS error on positioning, and it has a great influence on the positioning accuracy, but there is no related paper to discuss it. All the sensors share the generated particles, so the computational complexity of the algorithm does not increase rapidly as the number of nodes increases. Different from traditional particle filter, in the article, some reasonable particles are selected to positioning by making two selections based on residual analysis. These particles will have weights based on the residuals. The selection reduces the influence of the NLOS error and the target is localized by the particles. The simulation is performed under several different NLOS error, and results show that the algorithm is superior to Kalman filter (KF) and particle filter (PF).

**Signal model**

The propagation conditions of wireless communication systems are usually divided into two environments: line of sight (LOS) and non-line of sight (NLOS). In the case of obstacles, the wireless signal can only reach the receiving end through reflection, scattering, and diffraction, which we call NLOS communication. At this time, the wireless signal is received through multiple channels, and the multipath effect causes a series of problems such as delay in synchronization, signal attenuation, polarization change, and link instability, which are NLOS errors in Dou and Lei.\textsuperscript{12}

![Figure 1. Signal transmission model.](image-url)

As shown in Figure 1, \(N\) fixed nodes are randomly deployed in a certain scenario to locate a mobile node. The coordinate of the fixed node \(i\) is \([x_i, y_i]\) (\(i = 1, 2, \ldots, N\)) and the mobile node is \([x, y]\).

Ideally, the distance between the mobile node and the \(i\)th fixed node (\(i = 1, 2, \ldots, N\)) at time \(k\) is defined by equation (1)

\[
d_k^i = \sqrt{(x_i - x)^2 + (y_i - y)^2}
\]  

(1)

However, there are some noises in practical applications, including system noise and measurement noise. In the LOS state, the measured distance is defined by equation (2)

\[
d_k^i = \sqrt{(x_i - x)^2 + (y_i - y)^2} + \epsilon
\]  

(2)

where \(\epsilon\) is the measurement error, and it satisfies the Gaussian distribution and mean is 0. According to the existing research, estimating the measurement error distribution function as a Gaussian distribution function is an effective method.\textsuperscript{13}

In addition, the direct path between the fixed node and the mobile node may be blocked by obstacles, as shown in Figure 1, and this propagation mode is NLOS. The signal can be diffraction, reflection, or refraction, and the distance will increase as shown in equation (3)

\[
d_k^i = \sqrt{(x_i - x)^2 + (y_i - y)^2} + \epsilon + \epsilon_{\text{NLOS}}
\]  

(3)

where \(\epsilon_{\text{NLOS}}\) is the NLOS error, and it has various distributions under various conditions.
When it is a Gaussian distribution

\[
f(e_{\text{NLOS}}) = \frac{1}{\sqrt{2\pi \sigma_{\text{NLOS}}^2}} \exp\left(-\frac{(e_{\text{NLOS}} - \mu_{\text{NLOS}})^2}{2\sigma_{\text{NLOS}}^2}\right)
\]

where \(\sigma_{\text{NLOS}}\) is the standard deviation of the function and \(\mu_{\text{NLOS}}\) is the expectation of the function.

When it is uniform distribution \(e_{\text{NLOS}}(u_{\text{min}}, u_{\text{max}})\)

\[
f(e_{\text{NLOS}}) = \begin{cases} \frac{1}{u_{\text{max}} - u_{\text{min}}} & \text{if } u_{\text{min}} \leq e_{\text{NLOS}} \leq u_{\text{max}} \\ 0 & \text{else} \end{cases}
\]

where \(u_{\text{min}}\) and \(u_{\text{max}}\) are the upper and lower boundaries of the distribution, respectively. The variable is evenly distributed from \(u_{\text{min}}\) to \(u_{\text{max}}\).

When it is exponential distribution \(e_{\text{NLOS}}(\lambda)\)

\[
f(e_{\text{NLOS}}) = \begin{cases} \lambda^{-1} \exp\left(-\frac{e_{\text{NLOS}}}{\lambda}\right) & \text{if } e_{\text{NLOS}} \geq 0 \\ 0 & \text{if } e_{\text{NLOS}} < 0 \end{cases}
\]

where \(\lambda\) is rate parameter, and it is the number of times an event occurred per unit of time.

**IMPF algorithm**

**Particle filter**

The PF is developed in the late 1990s. It is another form of implementation of recursive Bayesian filtering.\(^{14}\) It uses a set of particles (also called samples) to represent the posterior distribution of some stochastic process given noisy and/or partial observations. The state-space model can be nonlinear, and the initial state and noise distributions can take any form required. PF techniques provide a well-established methodology.\(^{15-17}\)

There are many studies about it in positioning systems. A new fusion-simulated multi-feature-based improved simulated annealing PF tracking algorithm is proposed in Zhao et al.\(^{18}\) Li and Liu\(^{19}\) proposed a new PF algorithm that combines the traditional PF algorithm with the KF algorithm. Liu\(^{20}\) proposed a Gaussian and PF pre-detection tracking algorithm called the Gaussian sum particle filter–track before detect (GSPF-TBD) algorithm.

**Parameter description**

Suppose there are \(P\) particles and \(N\) fixed nodes. At time \(k\), the measured distance between the mobile node and the \(i\)th fixed node is \(\hat{d}^k_i\); the distance between the \(p\)th particle and the \(i\)th fixed node is \(\hat{d}^k_{i,p}\), the weight is given based on the distance difference, and it is \(W^k_{i,p}\).

The flow chart of the algorithm is shown in Figure 2.

**Particle weighting**

The particle weight is related to the positioning result. Unlike general PF, IMPF obtains particle weight by residual analysis rather than Gaussian function. The whole process is implemented as follows:

- Calculate the distance \(\hat{d}^k_{i,p} - \hat{d}^k_i\). Define the difference as the residual of \(p\)th particle to \(i\)th fixed node, denoted as \(\text{res}_{i,p}^k\).

\[
\text{res}_{i,p}^k = \left|\hat{d}^k_{i,p} - \hat{d}^k_i\right|
\]

- Calculate the residuals of the \(p\)th particles of each fixed node and add them together. Define it as the residual of \(p\)th particle, denoted as \(\text{res}_p^k\).

\[
\text{res}_p^k = \sum_{i=1}^{N} \left|\hat{d}^k_{i,p} - \hat{d}^k_i\right|
\]

Find the reciprocal of \(\text{res}_p^k\), and it is the weight of the \(p\)th particle

\[
W_p^k = \frac{1}{\text{res}_p^k}
\]

**Particle selection**

First selection. When all the particle weights are obtained, it will make the first selection. As the weight
indicates the possibility that the particle is suitable for mobile node, first selection chooses some particles with larger weights and defines the average weight as threshold. Given the generated \( P \) particles and their weights \( W \), first selection will choose \( S \) particles from them.

First selection chooses some particles with larger weights, which are reasonable at the overall level. However, once there are several fixed nodes that are inaccurate in determining the particles, some unreasonable particles can also pass first selection. Therefore, a second selection is necessary.

**Second selection.** The probability of NLOS errors can be estimated using particle residuals. It is assumed that the state where the residual is greater than the average contains the NLOS error. The residual ratio that is considered to contain the NLOS error is recorded as \( \lambda \). Therefore, for the LOS propagation, there are approximately \( N(1-\lambda) \) fixed nodes whose residuals are less than the residual mean. When a particle is away from the mobile node, its residual will also become larger. The particles with greater residuals will be discarded.

The particle number will be checked before positioning begins. When the propagation state is terrible, most propagation between the fixed node and the mobile node have NLOS errors, and when no particles are left after selections for the contradiction, then particles selected in first selection are used; otherwise, those that are selected in second selection are used. See Table 1 for fake code of related algorithm.

### Table 1. Algorithm second selection.

| Input: \( S \) particles with weights \( W^i \) (\( i = 1, 2, \ldots, S \)) | Output: \( Q \) particles with weights \( W^i \) (\( i = 1, 2, \ldots, Q \)) |
| --- | --- |
| 1. \( a = 0 \) | 2. for \( j = 1 : N \) |
| 3. for \( j = 1 : S \) | 4. if \( \text{rest}^i_j = \sum_{p=1}^{S} \frac{\text{rest}^i_p}{(S+1)} \) |
| 5. \( a = a + 1 \) | 6. end for |
| 7. end for | 8. \( \lambda = \frac{a}{S+1} \) |
| 9. \( t = N \times (1 - \lambda) \) | 10. for \( j = 1 : S \) |
| 11. \( b = 0 \) | 12. if \( \text{rest}^i_j = \sum_{p=1}^{S} \frac{\text{rest}^i_p}{S+1} \) |
| 13. if \( b > t \) | 14. \( b = b + 1 \) |
| 15. end for | 16. if \( b = t \) |
| 17. particle reserved | 18. else |
| 19. particle discard | 20. end for |

#### Location estimate

**Least squares method.** It is a mathematical optimization method. It finds the best function match for the data by minimizing the sum of the squares of the errors. A typical class of functional models is the linear function model. Location estimate algorithm is based on linear least squares (LLS).

Consider the overdetermined equations \( \sum_{i=1}^{m} x_i \beta_j = y_i \), \( (i = 1, 2, 3, \ldots, m) \), where \( m \) represents \( m \) equations and \( n \) represents \( n \) unknowns \( \beta \). After it is vectorized, it is \( X \beta = y \), and the equations generally have no solution; so in order to select the most appropriate \( \beta \) to make the equation as far as possible, the residual sum-of-squares function \( S(\beta) = ||X \beta - y||^2 \) is introduced. When \( \beta = \hat{\beta} \), \( S(\beta) \) is the minimum, \( \hat{\beta} = \text{argmin}(S(\beta)) \). Derivating \( S(\beta) \) and finding the extreme value, we can get \( X'^T X \hat{\beta} = X'^T y \), and if \( X'X \) is a non-singular matrix, \( \hat{\beta} = (X'^T X)^{-1} X'^T y \) is a unique solution.

**Location estimate.** Assuming that first selection chooses \( S \) particles and second selection chooses \( Q \) particles, and the retained particle number \( L = S \) when a contradiction occurs, otherwise \( L = Q \).

Before the positioning begins, the retained particle weights will be normalized, and the weight of \( p \)th particle at time \( k \) is denoted as \( NW^k_p \). At time \( k \), for the \( i \)th fixed node, the distance \( \text{dis}_i^k \) between it and the mobile node can be given by

\[
\text{dis}_i^k = \sum_{p=1}^{L} NW^k_p \hat{d}_{i,p}^k \tag{10}
\]

We can get the distance (\( \text{dis}_i \)) at time \( k \) and other times by equation (10). When all \( \text{dis}_i \) are known, as all coordinates of fixed nodes are known in advance, the ones of the mobile node can be calculated. They are obtained using the maximum likelihood, and it can be explained by the following equation. First, according to the coordinates of the mobile node and the fixed node, the distance between them is defined as equation (11)

\[
\begin{align*}
(x_1 - x)^2 + (y_1 - y)^2 &= (\text{dis}_1)^2 \\
... \\
(x_N - x)^2 + (y_N - y)^2 &= (\text{dis}_N)^2
\end{align*} \tag{11}
\]

Then by calculating equation (11), we get equation (12)

\[
\begin{align*}
2x(x_1 - x_2) + 2y(y_1 - y_2) &= (\text{dis}_2)^2 - (\text{dis}_1)^2 - (x_1^2 + y_1^2) + (x_2^2 + y_2^2) \\
... \\
2x(x_N - x_\bar{N}) + 2y(y_1 - y_{\bar{N}}) &= (\text{dis}_N)^2 - (\text{dis}_1)^2 - (x_N^2 + y_N^2) + (x_{\bar{N}}^2 + y_{\bar{N}}^2)
\end{align*} \tag{12}
\]
The above equations can be expressed by the linear equation \( AX = B \), where \( A \) and \( B \) are defined as equation (13)

\[
A = \begin{bmatrix}
(x_1 - x_2) & (y_1 - y_2) \\
(x_1 - x_3) & (y_1 - y_3) \\
\vdots & \vdots \\
(x_1 - x_N) & (y_1 - y_N) \\
(dis_2)^2 - (dis_1)^2 - (x_2^2 + y_2^2) + (x_1^2 + y_1^2) \\
(dis_3)^2 - (dis_1)^2 - (x_3^2 + y_3^2) + (x_1^2 + y_1^2) \\
\vdots \\
(dis_N)^2 - (dis_1)^2 - (x_N^2 + y_N^2) + (x_1^2 + y_1^2)
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
dis_2^2 \\
dis_3^2 \\
\vdots \\
dis_N^2
\end{bmatrix}
\]

(13)

Finally, the estimated coordinates of the mobile node is

\[
[x, y]^T = (A^T A)^{-1} A^T B
\]

(14)

**Particle replication and motion**

Once the positioning is complete, the retained particles will be utilized for next positioning. Therefore, construct replication layer to determine the particle replication process. The length of every layer is the weight of the particles. As the normalization of weights, the total length of all layers is 1. When constructing a layer, a random number satisfying uniform distribution from 0 to 1 is generated. Random numbers fall into some layer, and the corresponding particles are copied once. It can ensure that particles with larger weights have more chances to replicate while avoiding local optimization.

After the copying process is completed, the new particles will be distributed in a small range, avoiding excessive concentration of the particles. The variance of the coordinate obeys a Gaussian distribution with a mean of 0 and a variance \( \sigma^2_p \). These particles will be utilized for next positioning. The motion of the particle is given by equation (15)

\[
\left( x_p^{k+1}, y_p^{k+1} \right) = \left( x_p^k, y_p^k \right) + N \left( 0, \sigma^2_p \right)
\]

(15)

**Simulation**

The simulation runs on MATLAB, which evaluates the performance of the algorithm and compares it to the KF and the PF. The position of each Monte Carlo-operated fixed node is evenly deployed in a space of 100 \( \times \) 100 m\(^2\). Root mean square error (RMSE) as an indicator, defined by equation (16)

\[
RMSE = \left( \frac{1}{KT_0} \sum_{i=1}^{K} \sum_{k=0}^{T_0} (x(k) - \hat{x}_i(k))^2 + (y(k) - \hat{y}_i(k))^2 \right)^{1/2}
\]

(16)

**NLOS error obeys the Gaussian distribution**

Figure 3 shows the positioning results of IMPF. Even if the fixed nodes are randomly deployed, in most cases, the estimated value of the algorithm is very close to the true value. When a step has large error, such as \( x = 80 \), it can also be corrected quickly, and a reasonable positioning result can be obtained without error accumulation leading to positioning failure.
The cumulative distribution function (CDF) of the positioning error. It can be seen that the average positioning error of IMPF is about 2.5 m and that of KF and PF is about 4 m. The error of 90% in IMPF is less than 6.2 m, that is, under the given conditions, IMPF can basically control the error within 6 m, while that of PF is 10 m, KF exceeds 10 m. The positioning accuracies are obviously different. There is a cross point between the curves of PF and KF, probably due to particle distribution. Because some particles will deviate from the target, the PF curve is lower than the KF curve, but most of the particles are distributed near the target. In addition, KF is more susceptible to NLOS, which may also be the reason of the cross point.

\( \sigma_i \) is the standard deviation of the measured noise, and Figure 5 shows how it affects the positioning error. It changes from 1 to 7, which means when there is more measured noise, the RMSE of all algorithms increases. Under different \( \sigma_i \), the performance of IMPF is always better than the other algorithms. Specifically, compared with the PF and KF, the positioning accuracy of IMPF improves by 11.9% and 17.8%.

Figure 6 shows the effect of the number of fixed nodes on the positioning error. With the increase in the number of fixed nodes, IMPF algorithm has better positioning results than KF and PF. As the number of fixed nodes increases, more fixed nodes are again in LOS conditions where KF will get a better localization result than PF. When the number of fixed nodes is 5, compared with PF and KF, the positioning accuracy of IMPF improves by 15.4% and 23.3%, and the fixed nodes continue to increase, and this number will also increase.

Figure 7 shows that the positioning errors increase with NLOS error. In Figure 7(a), the average of RMSE in IMPF is 2.024 m, but it is 2.554 and 2.379 m in KF and PF, respectively. The RMSE difference of the three algorithms remains basically unchanged. In Figure 7(b), the RMSE of IMPF increases slower than that of KF, and the larger the standard deviation, the more obvious the advantage of IMPF. When the standard deviation is 9, the positioning accuracy of IMPF is 25.1% higher than KF and 15.3% higher than PF.

Figure 8 shows the effect of LOS propagation probability \( \alpha \) on positioning error. It can be seen that the positioning error decreases as the value of \( \alpha \) increases. This is because the larger the \( \alpha \), the more the fixed nodes in the LOS state, and the smaller the NLOS interference. It confirms the result in Figure 6. The downward trend of the three algorithms is roughly equivalent. Compared with PF and KF, the positioning accuracy of IMPF is improved by 17.4% and 20.1%. Combined with Figures 8 and 9, it can be seen that NLOS has a great influence on the positioning error, and the IMPF algorithm is significantly better than the popular KF and PF algorithms for NLOS.

**NLOS error obeys uniform distribution**

Figure 9 shows the CDF of the positioning error. The average positioning error of the IMPF is about 3.75 m. In the same case, the average error of KF and PF is about 5 m. IMPF achieves the lowest positioning error. The positioning error of the algorithm in Figure 10 is similar to that in Figure 6. It can be seen that the positioning error of IMPF is about 0.4 m lower than that of KF and about 0.1 m lower than that of PF.
Compared with PF and KF, the positioning accuracy of IMPF is increased by 5.7% and 10.8%, respectively. Figure 11 shows that RMSEs decline with more fixed nodes. The more fixed nodes are deployed, the less positioning errors are obtained, which is the same as Figure 7. KF and IMPF are greatly affected by the number of fixed nodes. When the number of fixed nodes increases, the positioning error will decrease rapidly. The performance of KF will exceed that of PF for the same reason in Figure 6. IMPF always maintains the lowest error throughout the change. When the number of fixed nodes is 10, the positioning accuracy of IMPF is improved by 12.7% compared with KF and 12.6% compared with PF.

The effect of the parameter $U_{\text{max}}$ on the positioning error is shown in Figure 12. As $U_{\text{max}}$ increases, the performance of algorithms decreases. IMPF maintains the best positioning result, which is about 0.2 m lower than the positioning error of PF, and the difference with KF increases as $U_{\text{max}}$ increases. With the increasing $U_{\text{max}}$, NLOS error distribution will be more uniform, which means more fixed nodes in NLOS. So the curve of PF is little lower than the curve of KF. Compared with KF and PF, its positioning accuracy increases by 10.3% and 9.0% on average.

**NLOS error obeys the exponential distribution**

In Figure 13, when $\sigma_i = 1$, the positioning performance of KF and PF is close, and the positioning error of IMPF is about 18% (about 0.36 m) lower than them.
When $s_i$ increases, the positioning error of the three algorithms increases, but IMPF always maintains the best positioning effect. Overall, IMPF is about 12.2% lower than PF and about 16.7% lower than KF in positioning error.

Figure 14 shows the effect of distribution parameter when $\lambda = 1$, IMPF and KF are equivalent, and the positioning error is about 0.3 m lower than PF. However, KF is sensitive to $\lambda$, and as $\lambda$ increases, the positioning error increases rapidly. The PF and IMPF algorithms are relatively stable. As the parameters increase, the positioning error of IMPF and PF remains at about 0.3 m, and the difference from KF increases from close to 0 to nearly 1 m. Similar to Figure 12, when $\lambda$ becomes larger, the NLOS error distribution is more uniform, so the RMSE of PF will gradually be lower than that of KF. When $\lambda = 8$, the performance of IMPF is improved by 14.0% and 24.8% when compared with PF and KF.

**Computational complexity analysis**

In addition to positioning accuracy, computational complexity is also an important factor. The simulation uses runtime to express the computational complexity. The three algorithms are coded using MATLAB R2017a in the Windows 10 PC. The running time is calculated through the MATLAB `tic` and `toc` functions for
elapsed time. The running time can reflect the complexity of the algorithm. The more complex the algorithm, the larger the calculation, and the longer the running time. As shown in Figure 15, KF obtains the lowest running time, and PF has the highest one. PF must spend some time generating and adjusting particles, and each node has its particles, causing the longest running time. In IMPF, all fixed nodes use common particles, so the computation complexity of IMPF is much lower than that of PF.

Comparison with other positioning algorithms
To further demonstrate the performance of the algorithm, we simply compare it with several other algorithms—two novel positioning algorithms: sparse pseudo-input gaussian process (SPGP) and import vector machine (IVM), and an algorithm based on PF: joint particle filter (JPF). SPGP is an NLOS mitigation method based on sparse pseudo-input gaussian process, and IVM is based on import vector machine.

All of them consider the problem of NLOS. Figure 16 shows the comparison of localization error. It can be seen that the positioning error of IMPF is better than that of JPF, which is also based on particle filtering. In addition, although IMPF is slightly worse than the other two latest algorithms, the three are very close. The average positioning errors of SPGP, IVM, and IMPF are about 0.85, 1.12, and 1.25 m, respectively.

We further compared the running time of IMPF, IVM, and SPGP, and the result is shown in Figure 17.
IMPF runs significantly less time than the other two algorithms. Combining Figures 16 and 17, the performance of IMPF is better than other algorithms.

**Conclusion**

The article proposes an IMPF algorithm, and residual is applied to the algorithm to enhance the accuracy of the positioning results. IMPF uses residuals to weight particles and select reasonable ones. Furthermore, the proposed algorithm does not make any assumptions about the distribution of NLOS errors, which means that IMPF is robust to different types of NLOS errors. Simulation results confirm the robustness of IMPF. In addition, IMPF estimates the distance from the mobile node to the fixed node, which can also be used to estimate the coordinates of the target.

The overall work of the future is to optimize the algorithm, because the algorithm is relatively simple, and the experiment only verifies our original idea. In order to get closer to the real world, we will try to add more channel noise to the simulation. At the same time, we will consider incorporating the algorithm into other new frameworks.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National High Technology Research and Development Program of China (Grant No. 2012AA120802), National Natural Science Foundation of China (Grant No. 61771186), Postdoctoral Research Project of Heilongjiang Province (Grant No. LBH-Q15121), Undergraduate University Project of Young Scientist Creative Talent of Heilongjiang Province (Grant No. UNPYSCT-2017125), and Postgraduate Innovative Research Project of Heilongjiang University (Grant No. YJSCX2019-059HLJU).

**ORCID iD**

Danyang Qin https://orcid.org/0000-0001-5829-6121

**References**

1. Vaghefi RM, Buehrer RM and Wymeersch H. Collaborative sensor network localization: algorithms and practical issues. *P IEEE* 2018; 106(6): 1089–1111.
2. Shi X, Chew YH, Yuen C, et al. A novel mobile target localization algorithm via HMM-based channel sight condition identification. *Peer Peer Netw Appl* 2016; 10(3): 1–15.
3. Choi JS, Lee WH, Lee JH, et al. Deep learning based NLOS identification with commodity WLAN devices. *IEEE T Veh Technol* 2018; 67(4): 3295–3303.
4. Pak JM, Ahn CK, Shi P, et al. Distributed hybrid particle/FIR filtering for mitigating NLOS effects in TOA based localization using wireless sensor networks. *IEEE T Ind Electron* 2016; 64(6): 5182–5191.
5. Monttaz AA, Behnia F, Amiri R, et al. NLOS identification in range-based source localization: statistical approach. *IEEE Sens J* 2018; 18: 3745–3751.
6. Tomic S, Beko M, Dinis R, et al. On target localization using combined RSS and AoA measurements. *Sensors* 2018; 18(4): 1266.
7. Tomic S and Beko M. A bisection-based approach for exact target localization in NLOS environments. *Signal Process* 2018; 143: 328–335.
8. Abu-Shaban Z, Zhou X and Abhayapala TD. A novel TOA-based mobile localization technique under mixed LOS/NLOS conditions for cellular networks. *IEEE T Veh Technol* 2016; 65(11): 8841–8853.
9. Ding C and Qi HD. Convex Euclidean distance embedding for collaborative position localization with NLOS mitigation. *Comput Optim Appl* 2018; 66(1): 1–32.
10. Wang J, Zhang L, Gao Q, et al. Device-free wireless sensing in complex scenarios using spatial structural information. *IEEE T Wirel Commun* 2018; 17(4): 2432–2442.
11. Wang G, So AMC and Li Y. Robust convex approximation methods for TDOA-based localization under NLOS conditions. *IEEE T Signal Proces* 2016; 64(13): 3281–3296.
12. Dou Z and Lei X. Principles of CDMA wireless communication. Beijing, China: Tsinghua University Press Co., Ltd, 2004.
13. Barsocchi P, Cimino MG, Ferro E, et al. Monitoring elderly behavior via indoor position-based stigmergy. *Pervasive Mob Comput* 2015; 23: 26–42.
14. Arulampalam MS, Maskell S, Gordon N, et al. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE T Signal Proces* 2002; 50(2): 174–188.
15. Del Moral P. Non-linear filtering: interacting particle resolution. *Markov Process Relat* 1996; 2(4): 555–581.
16. Del Moral P. Measure-valued processes and interacting particle systems. Application to nonlinear filtering problems. *Ann Appl Probab* 1998; 8(2): 438–495.
17. Laubenfels R. Feynman-Kac formulae: genealogical and interacting particle systems with applications. *J Am Stat Assoc* 2005; 100: 1460.
18. Zhao Y and Guo Q. Vision target tracking algorithm based on multi-features annealing particle filter. *Softw Eng* 2015; 7: 63–64.
19. Li Y and Liu M. Application of an improved particle filter algorithm in moving target tracking. *Ship Electron Eng* 2019; 39(10): 41–44.
20. Liu Z. Tracking algorithm for weak small target detection based on Gaussian and particle filter. *Comput Digit Eng* 2019; 47(9): 2190–2195.
21. Yang X. NLOS mitigation for UWB localization based on sparse pseudo-input Gaussian process. *IEEE Sens J* 2018; 18(10): 4311–4316.
22. Yang X, Zhao F and Chen T. NLOS identification for UWB localization based on import vector machine. *AEU: Int J Electron C* 2018; 87: 128–133.
23. Huang Y, Zhang S and Jing Y. An indoor mobile localization strategy based on particle filter in NLOS environment. In: *Proceeding of 2016 Chinese control and decision conference (CCDC)*, Yinchuan, China, 28–30 May 2016. New York: IEEE.