

Research Article

A Multifilter Location Optimization Algorithm Based on Neural Network in LOS/NLOS Mixed Environment

Zhentian Bian , Long Cheng , and Yan Wang 

Department of Computer and Communication Engineering, Northeastern University, Qinhuangdao 066004, China

Correspondence should be addressed to Long Cheng; chenglong@neuq.edu.cn

Received 19 May 2021; Revised 5 October 2021; Accepted 13 October 2021; Published 13 November 2021

Academic Editor: Roshan Thotagamuge

Copyright © 2021 Zhentian Bian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

While the modern communication system, embedded system, and sensor technology have been widely used at the moment, the wireless sensor network (WSN) composed of microdistributed sensors is favored due to its relatively excellent communication interaction, real-time computing, and sensing capabilities. Because GPS positioning technology cannot meet the needs of indoor positioning, positioning based on WSN has become the better option for indoor localization. In the field of WSN indoor positioning, how to cope with the impact of NLOS error on positioning is still a big problem to be solved. In order to mitigate the influence of NLOS errors, a Neural Network Modified Multiple Filter Localization (NNMML) algorithm is proposed in this paper. In this algorithm, LOS and NLOS cases are distinguished firstly. Then, KF and UKF are applied in the LOS case and the NLOS case, respectively, and appropriate grouping processing is carried out for NLOS data. Finally, the positioning results after multiple filtering are corrected by neural network. The simulation results illustrate that the location accuracy of NNMML algorithm is better than that of KF, EKF, UKF, and the version without neural network correction. It also shows that NNMML is suitable for the situation with large NLOS error.

1. Introduction

Within the framework of modern Internet of Things (IoT) technology, the wireless sensor network (WSN) plays a very key role. The technology relies on a network of sensors built together. The network collects the required information through various sensors, processes the data through embedded and information distribution technology, and then transmits the data to the top device. In the practical application level, the positioning technology based on this network has relative advantages compared with other positioning means. Due to the practical application demand at the present stage, satellite positioning has significant advantages in the precision demand of outdoor positioning, but it has obvious disadvantages in indoor positioning. Therefore, WSN positioning [1], which is light and small in size, inexpensive in price, low in energy consumption, and topologically strong, will be a cost-effective choice in the indoor positioning field with higher positioning accuracy requirements.

There are two kinds of nodes in WSN positioning: one is the mobile node, whose coordinates and motion information

are unknown; the other is the beacon node, whose coordinates are known. By transmitting signals, the distance from the mobile node can be obtained by means of time of arrival (TOA) [2, 3], time difference of arrival (TDOA) [4], angle of arrival (AOA) [5], RSSI [6, 7], and other possible ways, so as to achieve the positioning effect. In the actual signal transmission process, there is no ideal channel in the theoretical derivation. Even if the signal travels in a straight line, there is still a certain error, which is called the line-of-sight (LOS) error. Moreover, considering that possible obstacles may block the LOS path traveling along a straight line, the signal is compelled to travel along a non-line-of-sight path in the form of refraction or reflection, and the resulting error is known as a non-line-of-sight (NLOS) error [8, 9]. Therefore, it is worth studying how to effectively suppress the non-line-of-sight (NLOS) error of TOA, TDOA, AOA, and other measured values in the actual algorithm research.

The non-line-of-sight error different from the line-of-sight error of normal distribution is a kind of positive error of indefinite form that makes the measured value greater than the actual value. If the measurement value with such

error is not properly handled, the positioning effect will be significantly worse. Up to now, although there have been a variety of ways to judge and solve NLOS errors such as those listed in Related Work [10–25], how to reduce and weaken the impact of NLOS error on positioning is still a big problem to be solved.

In this paper, a Neural Network Modified Multiple Filter Localization (NNMML) algorithm is proposed. Our proposed algorithm has the following improvements:

- (1) The location accuracy of the proposed algorithm is better than that of the KF algorithm in the case of LOS and the UKF algorithm in the case of NLOS by means of multiple filtering after FCM grouping
- (2) Through parameter correction after each round of Kalman filter, the error caused by cycle parameter deviation is alleviated
- (3) Compared with the traditional hybrid algorithm, the follow-up neural network correction reduces the possibility that the trajectory in the positioning results does not conform to the real situation and reduces the error of most sampling moments
- (4) Because trajectory estimation is carried out not only by the least square method, but also by the BP neural network training method, the possibility of misjudgment is reduced

2. Related Work

At present, many pioneers have proposed a number of methods to try to identify and reduce the error caused by NLOS, but there is still no relative optimal solution. In [10], an interacting multimodel based on Kalman Filtering was proposed by Liao and Chen. This model has good extensibility under LOS/NLOS transition conditions and opens up a new idea for later comers. In [11], Chen proposed a method that parameters are provided by both time of arrival and received signal strength. The algorithm in [11] performs extremely well under the given special LOS/NLOS environment. However, its universality is not strong, and it is difficult to apply it in the unknown environment of NLOS. Then, in [12], J. Svec̃ko. estimated the distance as the state of the random system and realized the particle filter. In their particle filter algorithm, they used RSSI acquisition to calculate the important weight and resampled the weighted particles to ensure their reasonable distribution and density.

From [13, 14], the robust extended Kalman filter proposed by Hammes et al. is based on nonhorizon detection and probabilistic correlation coefficient adjustment, in which different distance subgroups are additively constructed. His improvement not only surpasses the classical extended Kalman filter in the non-LOS condition but also maintains the effect of the classical extended Kalman filter in the LOS condition so as to improve overall robustness. In [15], Fang et al. used the adaptive Kalman filter to modify noise parameters and improved the robustness compared

with the classical method, and combined with various basic algorithms, the accuracy was significantly improved, while in [16], Cheng et al. proposed a triple filtering algorithm which used FCM to divide the errors in the NOLS environment into soft NLOS and hard NLOS and then integrated them with the filtering results under LOS. This enhances the robustness of his algorithm in the mixed environment. In [17], the Raccoon Optimization Algorithm-based Accurate Positioning Scheme (ROA-APS) was proposed to strengthen the local search process involved in the estimation of the NLOS node. The authors propose a robust positioning method which tackles this problem by detecting the NLOS through a decision tree in [18]. It plays a good role in the mixed environment where the non-line-of-sight error is relatively small. The authors in [19] proposed an idea that applied a data clustering method of unsupervised machine learning to classify the NLOS signals and exclude them. Its accuracy can be improved to extremely close to pure LOS environment, but the adaptability to high NLOS error is poor.

In literature [20], attention was paid to the selection of a suitable hybrid TDOA/RTT/DOA location base station in a hybrid LOS/NLOS environment. The authors of [20] jumped out of the framework of optimizing the algorithm and successfully extracted more accurate measurement data. Tian and his partners proposed a distance and angle probability model so as to identify the NLOS propagation in [21]. This model can work well in a more specific NLOS environment. In [22], deep learning (DL) was applied to the NLOS identification. A localization method using a robust extended Kalman filter and track-quality-based (REKF-TQ) fusion algorithm was proposed to mitigate the effect of NLOS errors in [23]. Compared with the Kalman Filtering based on IMM, this fusion algorithm has much higher accuracy. The study in [24] developed a coarse NLOS detection algorithm based on discrete power levels to efficiently achieve the coarse NLOS mitigation, which automatically discards most unreliable dynamic anchors, while in [25], to obviously alleviate NLOS effects, the authors proposed a polynomial fitting-based adjusted Kalman filter (PF-AKF) method. The method employed polynomial fitting to accomplish both NLOS identification and distance prediction, and it was really inspirational. In addition, in my previous conference paper [26], I have made some preliminary explorations on this topic.

Up to now, even if more and more new methods have been proposed, how to reduce and weaken the influence of NLOS error on positioning is still a big problem that is worth of further study.

3. Personal Method

3.1. Signal Model. In the plane of the node to be tested, n ($n \geq 3$) positions are randomly selected to place beacon nodes, and their coordinates, as known quantities, are, respectively, noted as follows:

$$(X_i, Y_i) \quad i = 0, 1, \dots, n - 1, n. \quad (1)$$

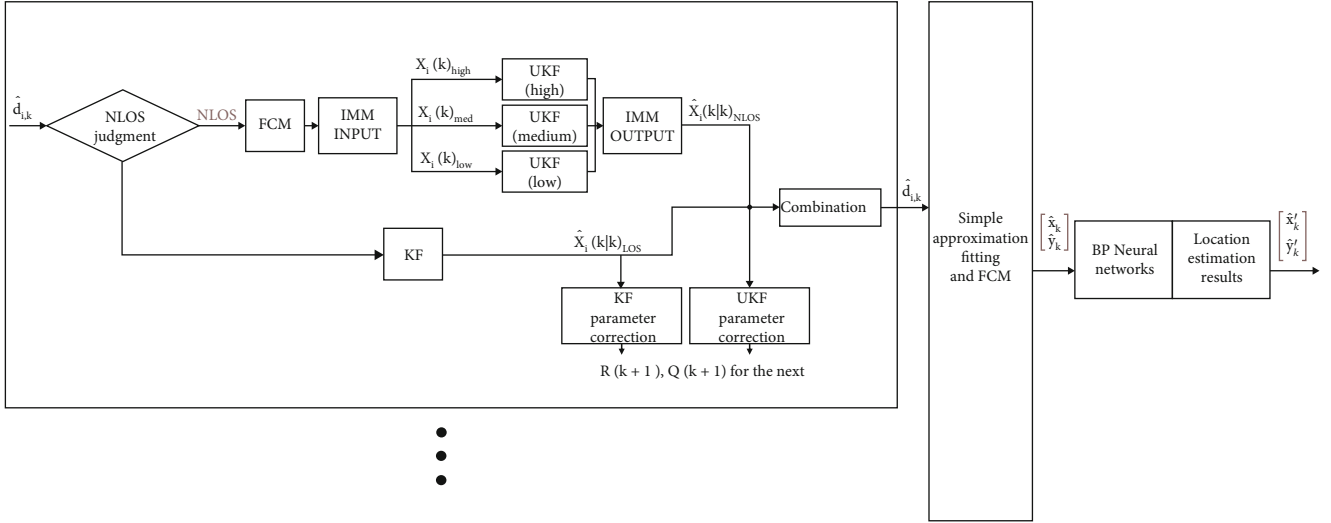


FIGURE 1: Algorithm flow chart.

Set the number of observations to be prepared, fix the real action track of the mobile node (which does not know the specific coordinates when measuring), and mark as (X_k, Y_k) at time k . Then, the accurate distance between it and the beacon node is

$$D_{ik} = \sqrt{(X_i - X_k)^2 + (Y_i - Y_k)^2}, \quad (2)$$

where in D_{ik} , k represents the k th moment and i represents the i th beacon node.

In the LOS environment, due to the nonideal channel, the actual measured distance through the TOA/TDOA/RSSI model is

$$D_{ik}(\text{LOS}) = D_{ik} + n_{\text{LOS}}, \quad (3)$$

where n_{LOS} is a Gaussian distribution which has a mean of 0 and a standard deviation σ_{LOS} , noted as $n_{\text{LOS}} \sim N(0, \sigma_{\text{LOS}}^2)$. It owns the following distribution function.

$$f(n_{\text{LOS}}) = \frac{1}{\sqrt{2\pi} \sigma_{\text{LOS}}} \exp\left(-\frac{n_{\text{LOS}}^2}{2\sigma_{\text{LOS}}^2}\right). \quad (4)$$

In the NLOS environment, due to the existence of possible obstacles, which may block the LOS path traveling along a straight line and compel the signals to travel along a non-line-of-sight path in the form of refraction or reflection, the actual measured distance becomes more complex just as [27, 28]

$$D_{ik}(\text{NLOS}) = D_{ik} + n_{\text{LOS}} + n_{\text{NLOS}}, \quad (5)$$

where n_{NLOS} has many possibilities, and here it is briefly summarized as one of Gaussian distribution, uniform distribution, and Poisson distribution.

If n_{NLOS} satisfies the Gaussian distribution, it will obey the Gaussian distribution owning the mean of μ_{NLOS} and

the standard deviation of σ_{NLOS} , noted as $n_{\text{NLOS}} \sim N(\mu_{\text{NLOS}}, \sigma_{\text{NLOS}}^2)$. And its distribution function is as follows:

$$f(n_{\text{NLOS}}) = \frac{1}{\sqrt{2\pi} \sigma_{\text{NLOS}}} \exp\left(-\frac{(n_{\text{NLOS}} - \mu_{\text{NLOS}})^2}{2\sigma_{\text{NLOS}}^2}\right). \quad (6)$$

When n_{NLOS} satisfies a uniform distribution, its minimum and maximum values are a and b , respectively, i.e., $n_{\text{NLOS}} \sim U(a, b)$, whose distribution function satisfies as follows:

$$f(n_{\text{NLOS}}) = \begin{cases} \frac{1}{b-a}, & a \leq n_{\text{NLOS}} \leq b, \\ 0, & \text{else.} \end{cases} \quad (7)$$

When n_{NLOS} satisfies an exponential distribution, its rate parameter is expressed as λ , i.e., $n_{\text{NLOS}} \sim E(\lambda)$, whose distribution function satisfies as follows:

$$f(n_{\text{NLOS}}) = \begin{cases} 1 - \exp(-\lambda n_{\text{NLOS}}), & n_{\text{NLOS}} \geq 0, \\ 0, & n_{\text{NLOS}} < 0. \end{cases} \quad (8)$$

3.2. General Concept. In this paper, a new algorithm named as the Neural Network Modified Multiple Filter Localization (NNMML) will be put forward. Its algorithm flow is illustrated in Figure 1. At first, the NLOS judgment is carried out towards the estimated distance \hat{d}_{ik} . The residual calculation method is adopted here, and the mean value of distance residual is obtained by calculating maximum likelihood estimation coordinates. This method has high practicability and high confidence interval, which can maintain an error rate of less than 5% in a high LOS/NLOS mixed environment.

After grouping, we perform targeted filtering on the data. For the LOS case, the updated results are worked out by traditional Kalman Filtering. In the linear environment, the effect of the traditional Kalman filter is always quite satisfactory. For the NLOS case, due to the large difference of

NLOS errors, the measurement data will be classified by NLOS classification based on FCM. High NLOS measurements, medium ones, and soft ones will be processed after FCM. Then, the NLOS measurements after filtering can be obtained by the interacting multimodel (IMM).

After we get the updated results, the noise parameters such as the covariance matrix are adjusted adaptively by using corresponding formulae for KF and UKF. The filter itself is constantly used to judge whether the dynamic of the system has changed and updates the noise parameters so as to be used in the next filtering. This adaptive approach can improve the robustness of the algorithm in complex environments.

Then, the distance data of multiple nodes were integrated to obtain the preliminary positioning results by the least square method, and the curve was simply fitted according to multiple moment data. FCM was used to remove the preliminary positioning results seriously deviating from the overall trend. Finally, the BP neural network is used to take the preliminary positioning results that fit the overall trend as the training set. Specifically, we use gradient descent to complete the back propagation of network parameters. Eventually, the output after training fills the gap of the positioning results, and we get the full result after correction.

3.3. NNMLL Algorithm

3.3.1. NLOS Judgment Based on Residual Calculation. Under the condition that the error value brought by NLOS is unknown, N estimated distance values $\{\hat{d}_{i,k} | i = 1, 2, \dots, N\}$ from N ($N \geq 3$) beacon nodes can be obtained at a certain measurement time k . According to the permutation and combination, we can have group M distance estimation.

Through the basic Newtonian least square method, the m -pair maximum likelihood estimation coordinates of the mobile node $\{(\hat{X}_{k,m}, \hat{Y}_{k,m}) | m = 1, 2, 3 \dots M\}$ calculated from the above m -set data can be obtained. Then, the distance between each group $(\hat{X}_{k,m}, \hat{Y}_{k,m})$ and the coordinate of beacon node (X_i, Y_i) is figured out, and the average residual difference is calculated with the estimated distance as follows:

$$\epsilon_{k,i} = \frac{\sum_{m=1}^M \sqrt{\left[d_{i,k} - \left(\sqrt{(X \wedge_{k,m} - X_i)^2 + (Y \wedge_{k,m} - Y_i)^2} \right)^2 \right]^2}{M}. \quad (9)$$

The calculated residual difference between the i th beacon node and the mobile node that we achieve at time k will make a comparison with the measurement noise standard deviation σ to confirm whether it belongs to the NLOS category or not.

$$\omega_{k,i} = \begin{cases} 0, & \epsilon_{k,i} \leq \sigma, \\ 1, & \epsilon_{k,i} > \sigma. \end{cases} \quad (10)$$

As is shown in formula (10), if $\omega_{k,i}$ is zero, it is judged in the LOS case, otherwise in the NOLS environment.

3.3.2. Kalman Filtering. Since the KF algorithm still has satisfactory accuracy in the linear environment, Kalman Filtering is still adopted in the LOS environment.

Firstly, we define the basic state parameters between the i th beacon node and the mobile node at time k as $X_i(k)$:

$$X_i(k) = \begin{bmatrix} \hat{d}_{i,k} \\ \dot{d}_{i,k} \end{bmatrix}, \quad (11)$$

where $\hat{d}_{i,k}$ represents the distance between the i th beacon node and the mobile node and $\dot{d}_{i,k}$ denotes the speed of mobile node.

Thus, state equation can be worked out as follows:

$$X_i(k) = FX_i(k-1) + Cw(k), \quad (12)$$

where coefficient matrices are, respectively,

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}. \quad (13)$$

Δt is the time interval between two measurements and $w(k)$ represents the process noise vector.

Moreover, measurement equation is obtained as follows:

$$Z_i(k) = HX_i(k) + v(k), \quad (14)$$

where column vector

$$H = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad (15)$$

and $v(k)$ delegates the measurement noise vector.

From the above state equation and measurement equation, the following iterative formula of the KF algorithm can be derived:

$$\begin{aligned} \hat{X}_i(k|k-1) &= F\hat{X}_i(k-1|k-1), \\ P_i(k|k-1) &= FP_i(k-1|k-1)F^T + R(k-1), \end{aligned} \quad (16)$$

where $R(k)$ represents covariance matrix of the measured noise whose initial value is set as $\sigma_{\text{LOS}}^2 CC^T$.

$$E_i(k) = Z_i(k) - H\hat{X}_i(k|k-1), \quad (17)$$

$$S_i(k) = HP_i(k|k-1)H^T + Q(k-1), \quad (18)$$

where $Q(k)$ represents the covariance matrix of the observed noise which is independent of $R(k)$.

From following formula (19), the incremental Kalman gain $K_i(k)$ in the iteration is ciphered out:

$$K_i(k) = P_i(k|k-1)H^T(S_i(k))^{-1}. \quad (19)$$

The updated state parameters and covariance required by the round of iteration can be obtained from the Kalman gain $K_i(k)$ as follows:

$$\begin{aligned}\widehat{X}_i(k|k) &= \widehat{X}_i(k|k-1) + E_i(k)K_i(k), \\ P_i(k|k) &= P_i(k|k-1)(I - K_i(k)H),\end{aligned}\quad (20)$$

where I is the identity matrix.

3.3.3. Unscented Kalman Filtering. Unscented Kalman Filtering (UKF) is an improved algorithm of Kalman Filtering. Although its advantages are not obvious in the LOS environment with linear signal propagation, it can significantly correct the nonlinear error and improve the positioning accuracy in the nonlinear environment. So UKF is adopted in the NLOS environment in this paper.

(1) *Initialization.* Its state equation and measurement equation can be derived from the same formulae in KF. Therefore, those formulae will not be repeated and elaborated in this part.

For the convenience of subsequent calculation, X_i is defined as the basic state parameters between the i th beacon node and the mobile node. \bar{x}_i and P_i are, respectively, its mathematical expectation and its covariance matrix.

(2) *Calculate the Sigma Points and Their Weights.* Such a state, as an N -dimensional random variate, owns a total of $2n + 1$ sigma points, which are obtained by formula as follows:

$$X_{i,j}(k-1) = \begin{cases} \bar{x}_i(k-1) & j=0, \\ \bar{x}_i(k-1) + \sqrt{(n+\lambda)(P_i(k-1))_j} & j=1, 2, 3, 4, \dots, n, \\ \bar{x}_i(k-1) - \sqrt{(n+\lambda)(P_i(k-1))_{j-n}} & j=n+1, \dots, 2n, \end{cases}\quad (21)$$

where $(P_i(k-1))_j$ signified the j th column in the covariance matrix P_i .

The sigma point weights follow the following law $\omega_0^m = \lambda/(n+\lambda)$, $\omega_j^m = 1/2(n+\lambda)$, $j=1, 2, 3, \dots, 2n$.

And the weights of its variances satisfy $\omega_0^c = \lambda/(n+\lambda) + (1-\alpha^2 + \beta)$, $\omega_j^c = 1/2(n+\lambda)$, $j=1, 2, 3, \dots, 2n$, where λ follows the expression $\lambda = \alpha^2(n+k) - n$, α is determined by the dispersion degrees of the above sigma points, and the optimal value of β that can be obtained in Gaussian distribution is two. In addition, in this circumstance, k is set as zero.

(3) *State Prediction.* From the state equation and measurement equation, the following state prediction of the UKF algorithm can be derived:

$$\begin{aligned}\widehat{X}_i(k|k-1) &= \sum_{j=0}^{2n} \omega_j^m X_{i,j}(k-1), \\ P_i(k|k-1) &= \sum_{j=0}^{2n} \omega_j^c [X_{i,j}(k-1) - \widehat{X}_i(k|k-1)] \\ &\quad \cdot [X_{i,j}(k-1) - X \wedge_i(k|k-1)]^T + Q(k), \\ \widehat{Z}_i(k|k-1) &= \sum_{j=0}^{2n} \omega_j^m Z_{i,j}(k-1),\end{aligned}\quad (22)$$

where $R(k)$ and $Q(k)$ stand for covariance matrix of measurement noise and covariance matrix of observation noise, respectively.

(4) *State Update.* On the basis of the formulae of Part 3, the updated variance and covariance matrix can be achieved as follows:

$$\begin{aligned}P_{xx_i}(k|k-1) &= \sum_{j=0}^{2n} \omega_j^c [X_{i,j}(k-1) - \widehat{X}_i(k|k-1)] \\ &\quad \cdot [X_{i,j}(k-1) - X \wedge_i(k|k-1)]^T + Q(k), \\ P_{zz_i}(k|k-1) &= \sum_{j=0}^{2n} \omega_j^c [Z_{i,j}(k-1) - \widehat{Z}_i(k|k-1)] \\ &\quad \cdot [Z_{i,j}(k-1) - Z \wedge_i(k|k-1)]^T + R(k), \\ P_{xz_i}(k|k-1) &= \sum_{j=0}^{2n} \omega_j^c [X_{i,j}(k-1) - \widehat{X}_i(k|k-1)] \\ &\quad \cdot [Z_{i,j}(k-1) - Z \wedge_i(k|k-1)]^T.\end{aligned}\quad (23)$$

In the wake, the Kalman gain of UKF can be figured out:

$$K_i(k) = P_{xz_i}(k|k-1)P_{zz_i}^{-1}(k|k-1).\quad (24)$$

In light of the above Kalman gain, the updated state parameter and its covariance will be done as follows:

$$\begin{aligned}\widehat{X}_i(k) &= \widehat{X}_i(k|k-1) + K_i(k)(Z_i(k) - \widehat{Z}_i(k|k-1)), \\ P_i(k|k) &= P_i(k|k-1) - K_i(k)P_{zz_i}(k|k-1)K_i^T(k) \\ &= P_i(k|k-1) - K_i(k)P_{xz_i}(k|k-1).\end{aligned}\quad (25)$$

At this point, one of its iterations ends.

3.3.4. Fuzzy C-Means Clustering. In practice, the distribution of NLOS is more complex, which will lead to great uncertainty in the error parameters of NLOS. In order to alleviate this problem, the FCM method is adopted for the NLOS

group after NLOS judgment, and then, filtering is carried out after classification.

At time k , the distance between the M beacon nodes and the mobile node is denoted as $\tilde{d}(k)$, $\tilde{d}(k) = [\tilde{d}_1(k), \dots, \tilde{d}_M(k)]$.

Let each element be divided into c groups; $[c_1, c_2, \dots, c_c]$ is the cluster center matrix, and μ_{ij} represents the membership degree of the j th distance element to the i th group, that is, the degree to which it belongs to this group. In this paper, c is set to 3.

Then, the objective function and constraint conditions of FCM are obtained as follows:

$$J(U, c_1, c_2, \dots, c_c) = \sum_{i=1}^c \sum_{j=1}^M \mu_{ij}^m d_{ij}^2$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^c \mu_{ij} = 1, & \forall j = 1, 2, \dots, M, \\ \mu_{ij} \in [0, 1], \end{cases} \quad (26)$$

where d_{ij} represents the Euclidean distance between the i th cluster center and the j th distance element and M indicates the fuzzy weight factor.

In order to make $J(U, c_1, c_2, \dots, c_c)$ reach the minimum value, the updated objective function is obtained by the Lagrange multiplier method:

$$L(U, c_1, c_2, \dots, c_c, \lambda_1, \lambda_2, \dots, \lambda_M)$$

$$= J(U, c_1, c_2, \dots, c_c) + \sum_{j=1}^M \lambda_j \left(\sum_{i=1}^c \mu_{ij} - 1 \right), \quad (27)$$

where λ is the Lagrange multiplier.

The clustering center and the membership degree of its corresponding elements can be obtained through partial derivative calculation:

$$c_i = \frac{\sum_{j=1}^M \mu_{ij}^m \tilde{d}_i(k)}{\sum_{j=1}^M \mu_{ij}^m}, \quad i \in [1, c], \quad (28)$$

$$\mu_{ij} = \frac{1}{\sum_{k=1}^c (d_{ij}/d_{kj})^{2/(m-1)}}. \quad (29)$$

By membership degree, M distance elements are divided into the group with the largest membership degree. Therefore, when $c = 3$ here, they can be divided into high NLOS measurements, medium ones, and low ones.

3.3.5. Interactive Multimodel Algorithm. The Interacting Multimodel (IMM) algorithm is an algorithm that is based on the Bayesian theory, through multiple filters to achieve the purpose of model adaptation. There are four steps in the IMM algorithm: interactive input, filter use, probability update, and interactive output. In this paper, the algorithm is used to carry out the grouping calculation under NLOS cases in order to achieve a more robust state estimation.

Owing to NLOS cases, the state equation and measurement equation satisfy formula (22) in the UKF without further elaboration.

(1) *Interactive Input.* In the first step, the mixing probability can be figured out, whose values are the ratio of the initial probability after the weighted transfer probability to its normalized coefficient:

$$\mu_{i,j}(k-1 | k-1) = \frac{p_{ij} \mu_i(k-1)}{\sum_i p_{ij} \mu_i(k-1)}, \quad (30)$$

where p_{ij} represents the transition probability which obeys the Markov Transition Probability Matrix

$$p = \begin{pmatrix} p_{11} & \dots & p_{1g} \\ \vdots & \ddots & \vdots \\ p_{g1} & \dots & p_{gg} \end{pmatrix}, \quad (31)$$

and parameter g in the matrix before is the dimension number.

Then, the measurement estimates can be weighted through the mixing probabilities. At the same time, its covariance is recalculated:

$$\tilde{X}_j(k-1 | k-1) = \sum_i \hat{X}_i(k-1 | k-1) \mu_{i,j}(k-1 | k-1), \quad (32)$$

$$\tilde{P}_j(k-1 | k-1) = \sum_i \left[P_i(k-1 | k-1) + (\hat{X}_i(k-1 | k-1) - \tilde{X}_{ij}(k-1 | k-1)) * (\hat{X}_i(k-1 | k-1) - \tilde{X}_{ij}(k-1 | k-1))^T \right] \mu_{i,j}(k-1 | k-1). \quad (33)$$

(2) *Filter Use.* The state parameters and covariance obtained by Equations (32) and (33) were used as the input of the filter, and UKF was selected as the filter in the NLOS environment. By the formulae in Section 3.3.3, the corresponding results can be calculated: $\hat{X}_j(k | k)$, $P_j(k | k)$, $E_j(k)$, $S_j(k)$.

(3) *Probability Update.* In this part, the probability is redistributed for the next iteration, and the updated mixing probability is as follows:

$$\mu_j(k) = \frac{L_j(k) \sum_i p_{ij} \mu_i(k-1)}{\sum_j (L_j(k) \sum_i p_{ij} \mu_i(k-1))}, \quad (34)$$

where the $L_j(k)$ above represents the maximum likelihood function on measurement equation, and it is a function of residual $E_j(k)$ as the dependent variable, which is following formula (35).

$$L_j(k) = \frac{1}{2\pi\sqrt{S_j(k)}} \exp\left(-\frac{E_j^2(k)}{2S_j(k)}\right) = N(E_j(k); 0, S_j(k)). \quad (35)$$

(4) *Interactive Output.* The mixing probability obtained in (34) is used as the weight to finally get the state parameters and covariance.

$$\begin{aligned} \hat{X}(k|k) &= \sum_j \mu_j(k) \hat{X}_j(k|k), \\ E_j(k) &= \hat{X}_j(k|k) - \hat{X}(k|k), \\ P(k|k) &= \sum_j \mu_j(k) \left[P_j(k|k) + E_j(k)E_j^T(k) \right]. \end{aligned} \quad (36)$$

3.3.6. *Parameter Correction.* In order to improve the overall adaptive performance of the algorithm, NNMML carries out an additional error correction for the observation noise covariance matrix $Q(k)$ and the measurement noise covariance matrix $R(k)$ after each filtering iteration, so as to reduce the error caused by the initial covariance value in the iteration process and improve the accuracy of filtering. The relevant parameters and formulae will be briefly described below.

Let the correction constant be b , which does not change in a given location. Then, we can reach the correction weight by the following formula:

$$\omega_k^* = \frac{1-b}{1-b^{k+1}}. \quad (37)$$

The modified formulae for KF and UKF filtering methods are slightly different, which will be briefly described below.

(1) *Parameter Correction Formulae Applied to KF.* The modification of the measurement noise covariance matrix $R(k)$ in the iteration is as follows:

$$R(k+1) = \omega_k^* [E_i(k)E_i^T(k) - HP_i(k|k-1)H^T] + (1-\omega_k^*)R(k). \quad (38)$$

The modification of the observation noise covariance matrix $Q(k)$ in the iteration is as follows:

$$Q(k+1) = \omega_k^* [K_i(k)E_i(k)E_i^T(k)K_i^T(k) + P_i(k|k) - FP_i(k-1|k-1)F^T] + (1-\omega_k^*)Q(k), \quad (39)$$

where matrices H and F are the same as described in KF filtering that

$$\begin{aligned} H &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\ F &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}. \end{aligned} \quad (40)$$

(2) *Parameter Correction Formulae Applied to UKF.* For the sake of simplifying the formal expression of the UKF revision, we define it for the time being that

$$E_i(k) = Z_i(k) - \hat{Z}_i(k|k-1). \quad (41)$$

The modification of the measurement noise covariance matrix $R(k)$ in the iteration is as follows:

$$R(k+1) = \omega_k^* [E_i(k)E_i^T(k) - (Pzz_i(k|k-1) - R(k))] + (1-\omega_k^*)R(k). \quad (42)$$

The modification of the observation noise covariance matrix $Q(k)$ in the iteration is as follows:

$$\begin{aligned} Q(k+1) &= \omega_k^* [K_i(k)E_i(k)E_i^T(k)K_i^T(k) + P_i(k|k) \\ &\quad - (Pxx_i(k|k-1) - Q(k))] + (1-\omega_k^*)Q(k). \end{aligned} \quad (43)$$

For fear of the loss of positive quality in the iteration of the above noise covariance matrix, the positive quality should be tested after correcting the parameters in each round. If the positive nature is lost, this round of parameter correction will be abandoned.

3.3.7. *Preliminary Location Estimation.* Through the above subalgorithm, the $\hat{X}_i(k|k)$, which combines the results of the two filtering methods, can be obtained; namely, we get the revised distance $\hat{d}_{i,k}$ between i th beacon node (coordinates (X_i, Y_i)) and mobile node at every measured time.

If the coordinates of the mobile node at the time k are defined as (x_k, y_k) , an underdetermined system of equations of it is able to be listed as follows:

$$\begin{cases} (x_k - X_1)^2 + (y_k - Y_1)^2 = \hat{d}_{1,k}^2, \\ (x_k - X_2)^2 + (y_k - Y_2)^2 = \hat{d}_{2,k}^2, \\ (x_k - X_3)^2 + (y_k - Y_3)^2 = \hat{d}_{3,k}^2, \\ \vdots \\ \vdots \\ (x_k - X_n)^2 + (y_k - Y_n)^2 = \hat{d}_{n,k}^2. \end{cases} \quad (44)$$

By simple transposition and matrix operations, it is capable of being rewritten as the following matrix equation:

$$[A][X] = [b], \quad (45)$$

where

$$[X] = \begin{bmatrix} x_k \\ y_k \end{bmatrix}, \quad (46)$$

and the matrices $[A]$ and $[b]$ satisfy as follows:

$$[A] = \begin{bmatrix} 2(X_1 - X_n) & 2(Y_1 - Y_n) \\ 2(X_2 - X_n) & 2(Y_2 - Y_n) \\ \vdots & \vdots \\ 2(X_{n-1} - X_n) & 2(Y_{n-1} - Y_n) \end{bmatrix}, \quad (47)$$

$$[b] = \begin{bmatrix} X_1^2 - X_n^2 + Y_1^2 - Y_n^2 + \tilde{d}_{1,k}^2 - \tilde{d}_{n,k}^2 \\ X_2^2 - X_n^2 + Y_2^2 - Y_n^2 + \tilde{d}_{2,k}^2 - \tilde{d}_{n,k}^2 \\ \vdots \\ X_{n-1}^2 - X_n^2 + Y_{n-1}^2 - Y_n^2 + \tilde{d}_{n-1,k}^2 - \tilde{d}_{n,k}^2 \end{bmatrix}.$$

Because of the high matrix dimensions, this system is overdetermined. So, we use the Gauss-Newton least square method to arrive at the answer.

$$[\hat{X}] = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} = (A^T A)^{-1} A^T b. \quad (48)$$

3.3.8. Trajectory Correction Based on BP Neural Network. By analogy, the approximate position of the mobile node at all times can be obtained preliminarily.

After the above coordinate calculation, the approximate positions of mobile nodes at all times are obtained. However, in practical application, there are still a few moments when the positioning trajectory does not conform to the overall motion trend. Therefore, in order to alleviate such errors, this paper adopts the BP neural network to correct the overall trajectory.

We define the approximate trajectory of the mobile node as

$$\begin{cases} x(k), \\ y(k), \end{cases} \quad (49)$$

namely, the trajectory for a parameter equation about the measuring interval k . For example, the coefficients of the parametric equation $x(k)$ can be obtained by the following least square method:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & 2 & \cdots & N-1 & N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 2^p & \cdots & (N-1)^p & N^p \end{bmatrix} * \text{coefficient} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_N \end{bmatrix}, \quad (50)$$

where N is the total moving steps and P is the order of the parametric equation. In general, P is set as 3.

Therefore, the Euclidean distance from the estimated position of the mobile node to its approximate motion trajectory at each time is as follows:

$$d_e(k) = \sqrt{(x_{\wedge k} - x(k))^2 + (y_{\wedge k} - y(k))^2}. \quad (51)$$

N Euclidean distances are divided into two groups through fuzzy C -means clustering (FCM) described above; that is, $c = 2$ is set. Thus, the data of group A which conforms to the trajectory and group B which deviates from the trajectory can be obtained. The specific process of FCM is similar to the previous one and will not be repeated here.

BP (back propagation) neural network, a multilayer negative feedback network, uses the adaptive mapping ability of the network to carry out back propagation. It can realize arbitrary nonlinear operation from input to output. From the perspective of hierarchical analysis of the network structure, its structure generally consists of the input layer, output layer, and implication layer. The BP neural network compares the error between the network output and the expected output by solving the network weight corresponding to the optimization of the target result, constructs the corresponding error function, uses the error gradient descent method to solve, corrects the network weight, and returns to the input layer to recalculate until the error is small.

In this paper, the BP neural network takes group A data which is in line with the motion trajectory as the training set and establishes the mapping relationship of $(k, x(k), y(k))$ in group A data, so as to train and output the modified mobile node position at the time k corresponding to group B data.

The following formula (52) represents the information transfer relationship between the input layer and the output layer, where n_1 and n_2 are the number of nodes in the input layer and the implication layer, respectively. Among the input layer, x_i is the input value of the i th node in the input layer; φ_{ij} is the weighting constant from the i th input layer node to the j th implication layer node. Among the implication layer, C_j is the additional offset of the j th node; $f(\cdot)$ is the transfer function, and its common ones are logsig, tansig, and pirelin. Here, we use $\text{logsig}(x) = 1/(1 + e^{-x})$; φ_{jk} is the weighting constant between the j th implication layer node and the k th output layer node; among the output layer, x'_k is the output value which belongs to the k th node.

$$x'_k = \sum_{j=1}^{n_2} \varphi_{jk} f \left(\sum_{i=1}^{n_1} \varphi_{ij} x_i + C_j \right). \quad (52)$$

Back propagation is to calculate the output error of each layer through the output layer and adjust the weight and additional offset of each layer according to the error gradient descent method. The overall error objective function is as follows:

$$Q = \frac{1}{2} \sum_{j=1}^P \sum_{i=1}^{n_3} (E_i - x_i^j)^2, \quad P < N, \quad (53)$$

where P , n_3 , and E_i are the number of samples, the number of nodes in the output layer, and the output expectation, respectively.

Here, we take the error function of y_j , the output result of the j th implication layer, as an example to introduce the gradient descent method:

$$\begin{aligned} P_j &= \sum_{i=1}^{n_1} (\varphi_{ij} x_i + C_j), \\ y_j &= f \left(\sum_{i=1}^{n_1} \varphi_{ij} x_i + C_j \right), \\ Q_j(\varphi_{ij}, C_j) &= \frac{1}{2} \sum_{j=1}^{n_3} \left((E_j - y_j)^2 \right). \end{aligned} \quad (54)$$

We choose the transfer function as follows, and it is clear that $\text{logsig}(x)$ is a special and initial state of the following function:

$$f(x) = \frac{\alpha}{1 + e^{-x/\beta}}. \quad (55)$$

This transfer function satisfies the following formula:

$$f'(x) = \frac{f(x) * [\alpha - f(x)]}{\alpha\beta}. \quad (56)$$

Then, the partial derivatives of the functions Q_j with respect to φ_{ij} and C_j are performed:

$$\begin{aligned} \frac{\partial Q_j(\varphi_{ij}, C_j)}{\partial \varphi_{ij}} &= \frac{\partial}{\partial \varphi_{ij}} \frac{1}{2} \sum_{j=1}^{n_3} \left((E_j - y_j)^2 \right) \\ &= \frac{\partial y_j}{\partial \varphi_{ij}} (E_j - y_j) = \frac{\partial P_j}{\partial \varphi_{ij}} f'(P_j) (E_j - y_j) \\ &= x_i \frac{f(P_j) [\alpha - f(P_j)]}{\alpha\beta} (E_j - y_j) \\ &= x_i \phi_{ij} \frac{\partial Q_j(\varphi_{ij}, C_j)}{\partial C_j} = \phi_{ij}, \end{aligned} \quad (57)$$

where $\phi_{ij} = (f(P_j) [\alpha - f(P_j)] / \alpha\beta) (E_j - y_j)$.

We start from the last layer to proceed layer by layer, pass the error forward by weight to get the error of the previous layer, and repeat successively. Meanwhile, the values of φ_{ij} and C_j are updated according to the gradient descent method to minimize the error:

$$\begin{aligned} \varphi_{ij} &= \varphi_{ij} - \eta_1 x_i \phi_{ij}, \\ C_j &= C_j - \eta_2 \phi_{ij}, \end{aligned} \quad (58)$$

TABLE 1: List of the parameters.

Parameter	Symbol	Values
Total moving steps	N	100
Number of beacon nodes	n	5
Standard of LOS error	σ_{LOS}	1
Mean of NLOS error	μ_{NLOS}	
Standard of NLOS error	σ_{NLOS}	
NLOS threshold	S	0.7
Rate parameter of NLOS error	λ_{NLOS}	
Maximum number of iterations		3000
Training target		10^{-7}
Number of nodes in implication layer	n_2	5

where η_1, η_2 are the learning rates and use the default value 0.05. The higher the learning rate is, the less the time to reach the training target will be. However, if the learning rates is too high, it may lead to a locally optimal solution, so we take a moderate value here.

By setting the maximum number of iterations and the training target, the modified results of K at the corresponding time of group B data were solved and combined with the data of group A to obtain more accurate

$$[\hat{X}] = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix}. \quad (59)$$

4. Simulation Results

In this paper, the positioning accuracy of the algorithm is simulated by the simulation software MATLAB.

The environment set by this simulation is roughly as follows:

All beacon nodes and mobile nodes are in a plane with an area of 100×100 during the measurement time. The mobile node moves in a smooth trajectory, and there is no nonphysical movement less than the measured interval. The coordinates of n beacon nodes set are an array of n rows and 2 columns randomly generated by the function "rand" in MATLAB, whose horizontal and vertical coordinates are known and fixed during measurement. In addition, a total of N measurements have been made.

The error of LOS and NLOS satisfies as follows.

Since the LOS error is caused by white noise satisfying the normal distribution, its mean value is 0 and its standard deviation is σ_{LOS} . There are many errors in NLOS, and three kinds of noise satisfying Gaussian distribution, uniform distribution, and Poisson distribution are selected as representatives. The mean value of μ_{NLOS} and the standard deviation of σ_{NLOS} are satisfied. Poisson distribution satisfies the condition that rate parameter is λ_{NLOS} .

In determining whether a certain beacon node and mobile node are in NLOS state at a certain moment, a number in the interval of $[0,1]$ is randomly generated by the function "rand." If the random number is greater than the

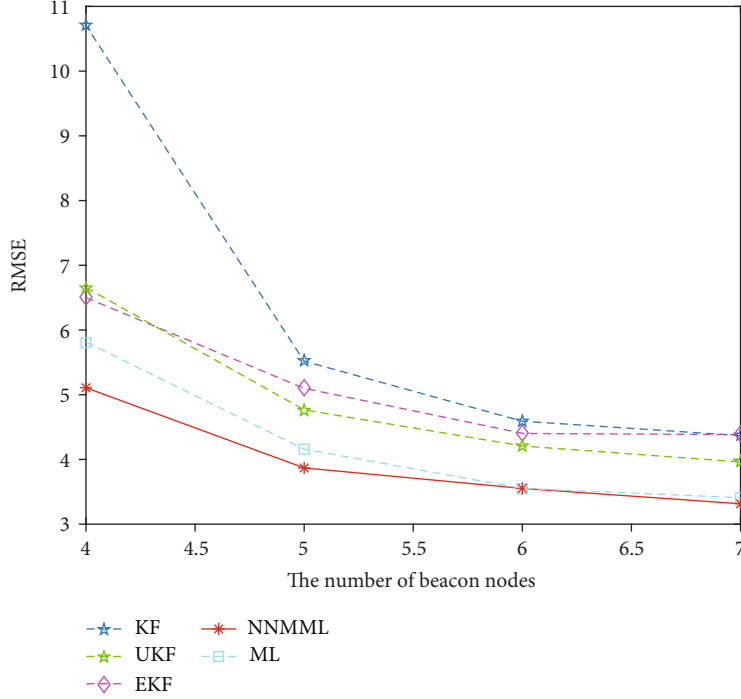


FIGURE 2: Difference in the number of BN.

NLOS threshold S , it is considered to be in the NLOS environment; otherwise, it is deemed to be in the LOS environment.

In the correction of the BP neural network, we set three parameters with partial empirical direction, maximum number of iterations, training target, and number of nodes in the implication layer. The training target is the gradient value target set in the neural network. The setting of the maximum number of iterations prevents the neural network from falling into an endless cycle when it cannot reach the gradient goal for a long time. Using too few nodes in the implication layer results in underfitting. On the contrary, using too many nodes may also lead to overfitting, so 5 is selected for reference [29].

In order to show the localization effect of the NNMML algorithm in the LOS/NLOS mixed environment, the Kalman Filtering (KF) algorithm, Extended Kalman Filtering (EKF) algorithm, Unscented Kalman Filtering (UKF) algorithm, and Multiple-filter Localization (ML) without NNs were used as the comparison algorithm group to carry out one thousand repeated experiments ($NUM = 1000$). Refer to Table 1 for some basic parameters of the data and the neural network without comparison items. The root mean square error is used as the error measurement standard, and its formula is as follows:

$$RMSE = \sqrt{\frac{1}{N} \frac{1}{NUM} \left(\sum_{k=1}^N [(x^{\wedge}_k - x_k)^2 + (y^{\wedge}_k - y_k)^2] \right)}. \quad (60)$$

Figures 2–4, respectively, show the variation trend of RMSE of the five algorithms, when the NLOS error obeys Gaussian distribution, by changing the number of BN, the mean of NLOS error, and the standard of NLOS error.

In Figure 2, as the number of mobile nodes n increases from 4 to 7, it can be clearly seen that the positioning errors of the five algorithms all show a downward trend. When the number of mobile nodes is small, the NNMML algorithm has obvious advantages and can better deal with the localization problem. However, as the number of mobile nodes continues to rise, the positioning accuracy of the NNMML algorithm and the ML algorithm is gradually approached, and its advantages become weaker.

In Figure 3, when the mean of NLOS errors μ_{NLOS} rises from 2 to 6, the positioning errors of the five algorithms all rise gradually, but the NNMML algorithm rises at the gentlest speed. In this process, KF, EKF, UKF, and ML increased by 49.33%, 31.74%, 35.71%, and 19.40%, respectively, while NNMML only increased by 13.42%, and its initial value was the lowest. Compared with other algorithms, the localization effect was the best.

In Figure 4, when the standard of NLOS errors σ_{NLOS} rises steadily from 1 to 5, all the five algorithms have significant changes. When $\sigma_{NLOS}=1$, the NLOS error tends to be a constant error, and the positioning error of NNMML and ML algorithm is similar. When the standard rises, the error ratio of NNMML to ML decreases from 99.24% to 89.54%, indicating that NNMML is more suitable for the case of relatively large NLOS.

Then, Figure 5 shows how RMSE of each algorithm changes with the rate parameter when the NLOS error satisfies the Poisson distribution. In this figure, when the rate

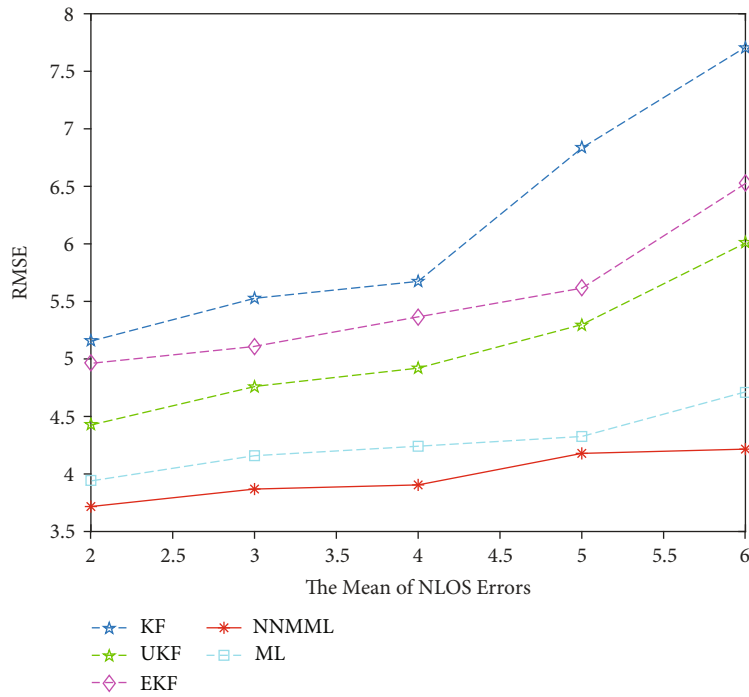


FIGURE 3: Difference in the mean of NLOS error (Gaussian).

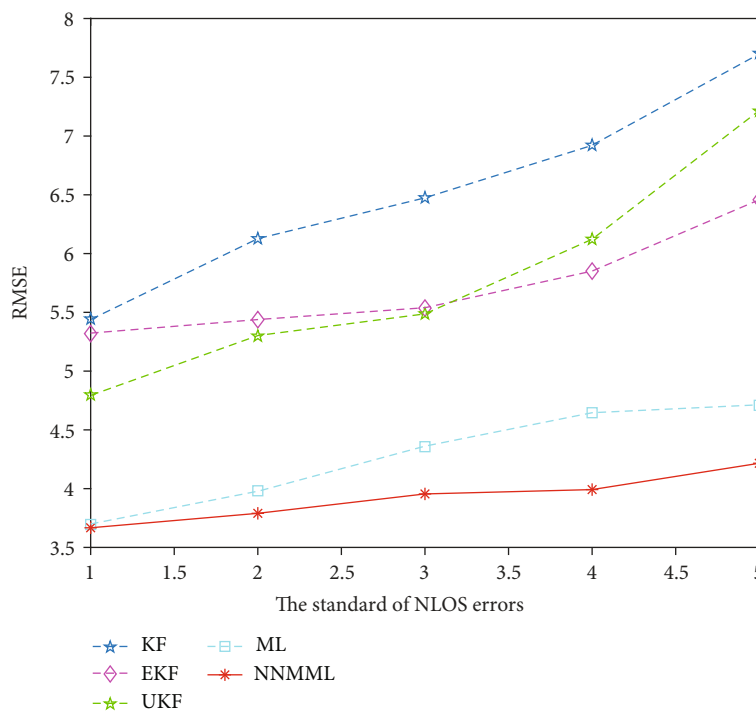


FIGURE 4: Different in the standard of NLOS error (Gaussian).

parameter changes from 4 to 7, RMSE of all algorithms shows an upward trend, while RMSE of the NNMML algorithm keeps the lowest and the upward trend is not obvious. With the increase of error, the NNMML algorithm

has a sharp increase in advantages over other algorithms. It can be preliminarily confirmed that the NLOS error of Poisson distribution is well suppressed by the NNMML algorithm.

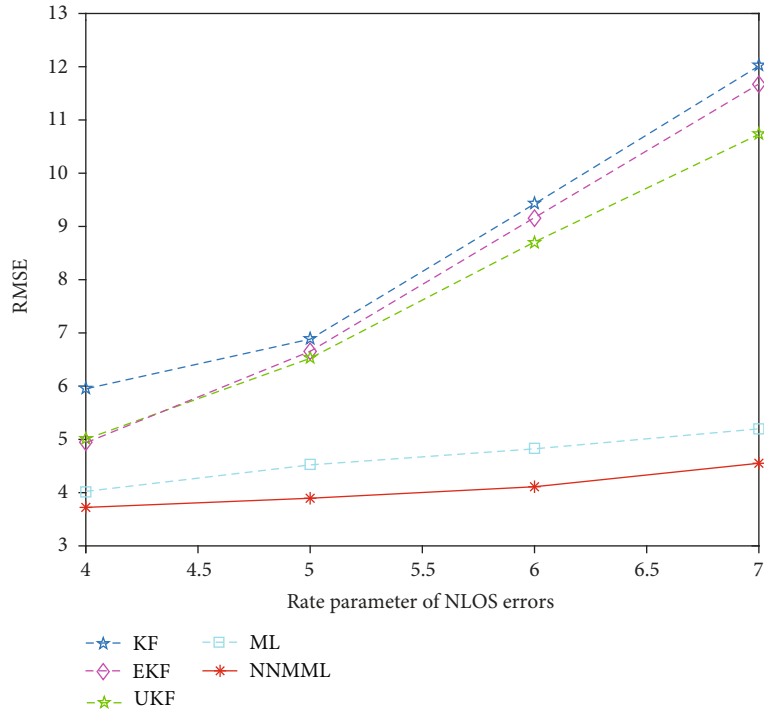


FIGURE 5: Difference in the rate parameter (Poisson).

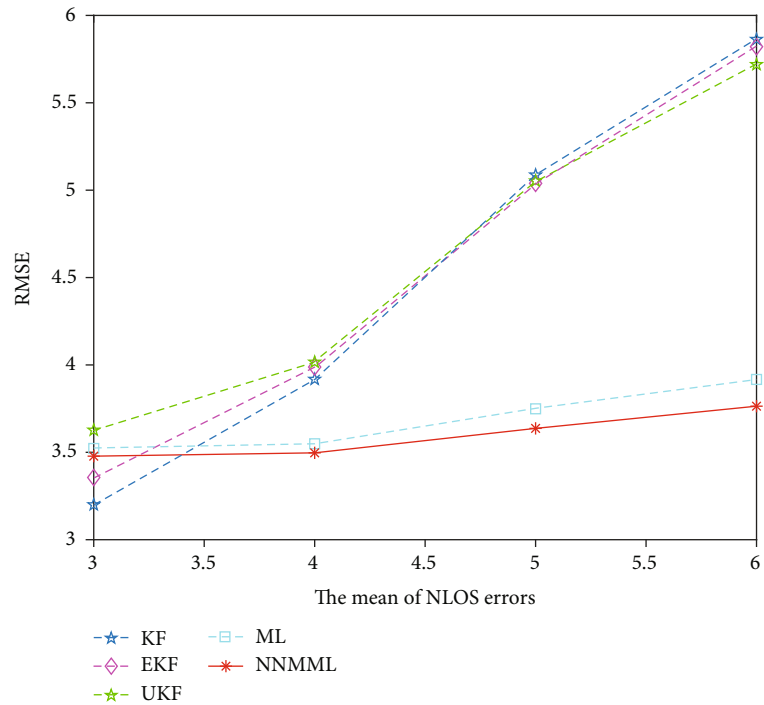


FIGURE 6: Difference in the mean of NLOS error (uniform).

Finally, Figure 6 shows the effect of the algorithm, if the NLOS error is uniform.

In Figure 6, the standard of NLOS errors μ_{NLOS} rises steadily from 3 to 6 and σ_{NLOS} remains 2. In this process, KF, UKF, and EKF change dramatically, while ML and

NNMML algorithms fluctuate less. Although KF and EKF have good performance when the maximum and minimum values are 5 and 1, respectively, NNMML has obvious advantages of universality from the perspective of overall RMSE and its stability.

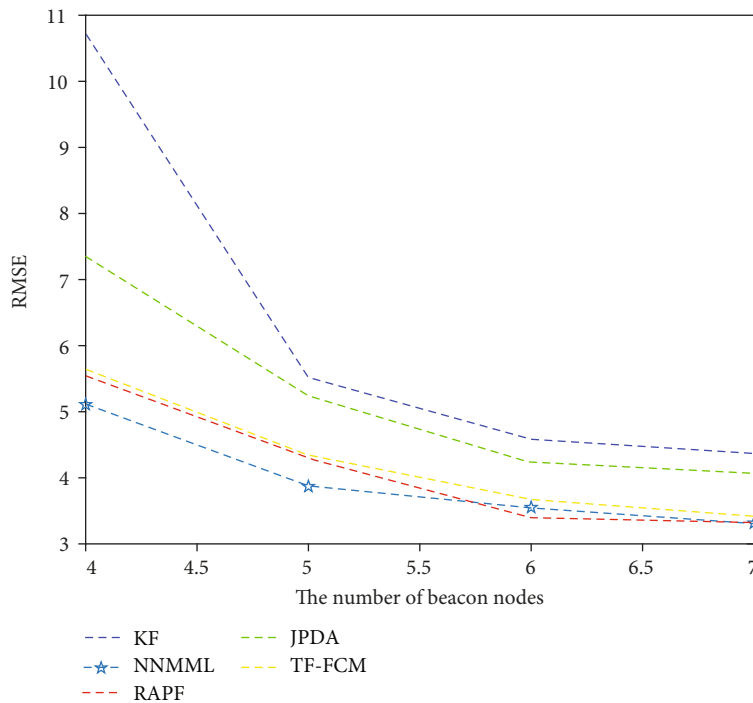


FIGURE 7: Further contrast on number of BN.

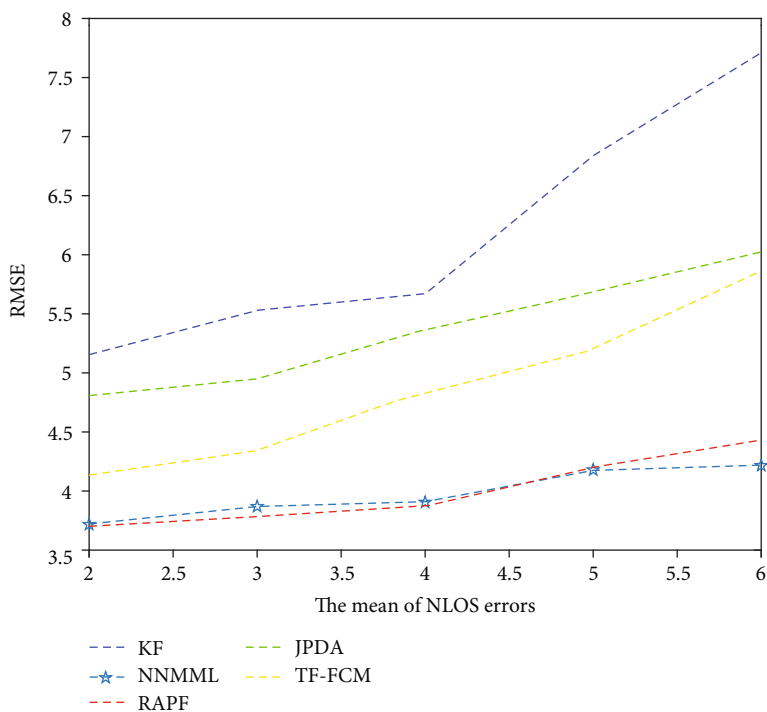


FIGURE 8: Further contrast on mean of NLOS error.

In addition to the above basic comparison, we selected TF-FCM, JPDA, RAPF, and other improved algorithms in [16, 30, 31] for horizontal comparison.

From Figures 7–9, it can be seen that our MMNNL algorithm is significantly stronger than TF-FCM and JPDA

when the number of BN changes or the error parameters of NLOS change. Compared with the improved algorithm RAPF based on PF, it is better in partial effect when the overall filtering effect is similar. For example, when the number of BN is low, it has an advantage of 8.5% to 11.2%. The

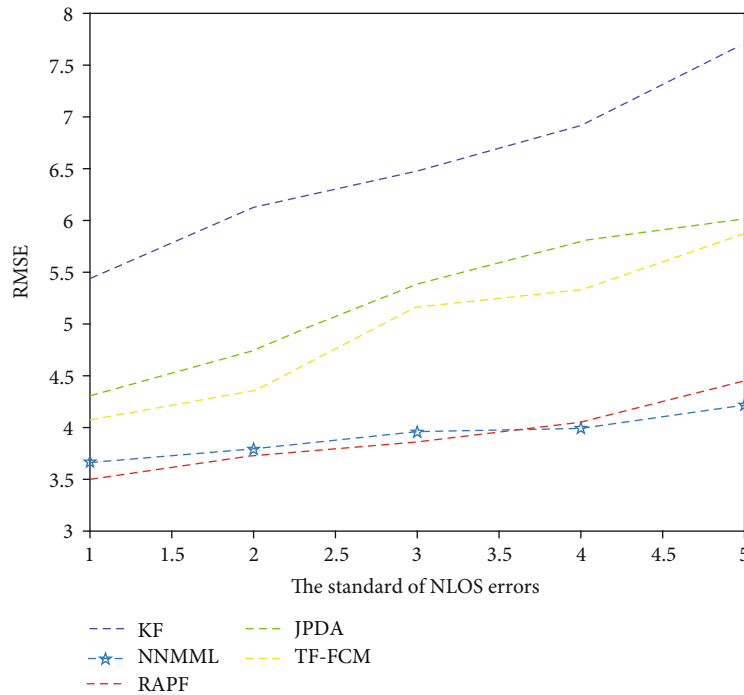


FIGURE 9: Further contrast on standard of NLOS error.

effect is gradually better when the mean of NLOS error increases. Compared with the RAPP algorithm, the optimization effect is enhanced in the high error region.

All in all, the NNMML algorithm has stronger error suppression ability and stronger robustness for uniformly distributed NLOS.

5. Discussion

The NNMML algorithm has excellent performance compared with conventional KF, EKF, and UKF in LOS/NLOS mixed environment. At the same time, compared with the ML algorithm without neural network, it also has a good effect in alleviating errors, with high robustness and precision. However, due to the addition of the neural network for correction, the overall algorithm consumes a little longer time. Therefore, our future work will focus on the optimization of time complexity and the improvement of redundant parts to reduce the time consuming of the algorithm. At the same time, our future work will also apply the NNMML algorithm to more specific and complex actual environments, instead of representing NLOS error completely through simple probability distribution, in order to improve the practical application value of the algorithm.

6. Conclusion

In this paper, an algorithm called the neural network modified multiple-filter localization is proposed. Firstly, LOS and NLOS cases are distinguished. Then, KF and UKF are applied in different environments, and appropriate grouping processing is carried out for NLOS. Finally, the positioning

results after multiple filtering are corrected by the neural network. Simulation results show that it is better than KF, EKF, UKF, and multiple filtering without the neural network. It achieves better accuracy and robustness in LOS/NLOS hybrid environments.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of Hebei Province under Grant No. F2020501012.

References

- [1] M. S. Rahman, Y. Park, K. D. Kim, and J. Arab, "RSS-based indoor localization algorithm for wireless sensor network using generalized regression neural network," *Arabian Journal for Science and Engineering*, vol. 37, no. 4, pp. 1043–1053, 2012.
- [2] R. M. Vaghefi and R. M. Buehrer, "Cooperative localization in NLOS environments using semidefinite programming," *IEEE Communications Letters*, vol. 19, no. 8, pp. 1382–1385, 2015.
- [3] I. Guvenc and C. C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE*

- Communications Surveys and Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.
- [4] F. Gustafsson and F. Gunnarsson, “Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements,” *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, 2005.
 - [5] T. Aso and T. Miyajima, “Sensor localization based on AOA-assisted NLOS identification,” *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. 93, no. 6, pp. 1274–1276, 2010.
 - [6] A. S. Paul and E. A. Wan, “RSSI-based indoor localization and tracking using sigma-point Kalman smoothers,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 860–873, 2009.
 - [7] X. W. Luo, W. J. O’Brien, and C. L. Julien, “Comparative evaluation of received signal-strength index (RSSI) based indoor localization techniques for construction jobsites,” *Advanced Engineering Informatics*, vol. 25, no. 2, pp. 355–363, 2011.
 - [8] J. A. John and N. R. Draper, “An alternative family of transformations,” *Applied Statistics*, vol. 29, no. 2, pp. 190–197, 1980.
 - [9] K.-T. Lay and W.-K. Chao, “Mobile positioning based on TOA/TDOA/TDOA measurements with NLOS error reduction,” in *2005 International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 545–548, Hong Kong, China, 2005.
 - [10] J.-F. Liao and B.-S. Chen, “Robust mobile location estimator with NLOS mitigation using interacting multiple model algorithm,” in *IEEE Transactions on Wireless Communications*, 2006.
 - [11] B.-S. Chen, “Mobile location estimator in a rough wireless environment using extended Kalman-based IMM and data fusion,” in *IEEE Transactions on Vehicular Technology*, 2009.
 - [12] J. Svečko, M. Malajner, and D. Gleich, “Distance estimation using RSSI and particle filter,” *ISA Transactions*, vol. 55, pp. 275–285, 2015.
 - [13] U. Hammes, E. Wolsztynski, and A. M. Zoubir, “Robust tracking and geolocation for wireless networks in NLOS environments,” *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING*, vol. 3, no. 5, pp. 889–901, 2009.
 - [14] U. Hammes and S. Member, “Robust mobile terminal tracking in NLOS environments based on data association,” *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 58, no. 11, pp. 5872–5882, 2010.
 - [15] X. Fang, L. Nan, Z. Jiang, and L. Chen, “Robust node position estimation algorithms for wireless sensor networks based on improved adaptive Kalman filters,” *Computer Communications*, vol. 101, pp. 69–81, 2017.
 - [16] L. Cheng, Y. Li, Y. Wang, Y. Bi, L. Feng, and M. Xue, “A triple-filter NLOS localization algorithm based on fuzzy C-means for wireless sensor networks,” *Sensors*, vol. 19, no. 5, p. 1215, 2019.
 - [17] A. Balamurugan, M. D. Priya, A. C. J. Malar, and S. Janakiraman, “Raccoon optimization algorithm-based accurate positioning scheme for reliable emergency data dissemination under NLOS situations in VANETs,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 11, pp. 10405–10424, 2021.
 - [18] Y. Xiaozhou and C. Ma, “Robust real-time kinematic positioning method based on NLOS detection and multipath elimination in GNSS challenged environments,” *Electronics Letters*, vol. 56, no. 24, pp. 1332–1335, 2020.
 - [19] F. Zhu, T. Ba, Y. Zhang, X. Gao, and J. Wang, “Terminal location method with NLOS exclusion based on unsupervised learning in 5G-LEO satellite communication systems,” *International Journal of Satellite Communications and Networking*, vol. 38, no. 5, pp. 425–436, 2020.
 - [20] Z. Deng, H. Wang, X. Zheng, and L. Yin, “Base station selection for hybrid TDOA/RTT/DOA positioning in mixed LOS/NLOS environment,” *Sensors*, vol. 20, no. 15, p. 4132, 2020.
 - [21] X. Tian, G. Wei, J. Wang, and D. Zhang, “A localization and tracking approach in NLOS environment based on distance and angle probability model,” *Sensors*, vol. 19, no. 20, p. 4438, 2019.
 - [22] K. K. Cwalina, P. Rajchowski, O. Blaszkiewicz, A. Olejniczak, and J. Sadowski, “Deep learning-based LOS and NLOS identification in wireless body area networks,” *Sensors*, vol. 19, no. 19, p. 4229, 2019.
 - [23] Y. Wang, J. Huihui, and L. Cheng, “A fusion localization method based on a robust extended Kalman filter and track-quality for wireless sensor networks,” *Sensors*, vol. 19, no. 17, p. 3638, 2019.
 - [24] T. Qiao and L. Cheng, “Coarse NLOS detection algorithm based on discrete power levels,” *IET Communications*, vol. 13, no. 10, pp. 1482–1487, 2019.
 - [25] Y. Wang, Y. Yan, Z. Li, and L. Cheng, “A mobile localization method in smart indoor environment using polynomial fitting for wireless sensor network,” *Journal of Sensors*, vol. 2020, 17 pages, 2020.
 - [26] Z. Bian, F. Kong, and L. Cheng, “An indoor localization algorithm based on IMM in mixed LOS/NLOS environments,” in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, Chengdu, China, 2020.
 - [27] I. Rasool and A. H. Kemp, “Statistical analysis of wireless sensor network Gaussian range estimation errors,” *IET Wireless Sensor Systems*, vol. 3, no. 1, pp. 57–68, 2013.
 - [28] N. Liu, Z. L. Xu, and M. Brain, “Geolocation performance with biased range measurements,” *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2315–2329, 2012.
 - [29] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
 - [30] L. Cheng, Y. Li, M. Xue, and Y. Wang, “An indoor localization algorithm based on modified joint probabilistic data association for wireless sensor network,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, 2021.
 - [31] L. Cheng, L. Feng, and Y. Wang, “A residual analysis-based improved particle filter in mobile localization for wireless sensor networks,” *Sensors (Basel, Switzerland)*, vol. 18, no. 9, p. 2945, 2018.