Sensors **2011**, *11*, 4207-4230; doi:10.3390/s110404207

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Communication

Applying Rprop Neural Network for the Prediction of the Mobile Station Location

Chien-Sheng Chen [1],* and Jium-Ming Lin [2]

[1] Department of Information Management, Tainan University of Technology, No. 529, Jhongjheng Rd., YongKang Dist., Tainan City 71002, Taiwan

[2] Department of Communication Engineering, Chung-Hua University, No. 707, Sec.2, WuFu Rd., Hsinchu, Taiwan; E-Mail: jmlin@chu.edu.tw

* Author to whom correspondence should be addressed; E-Mail: t00243@mail.tut.edu.tw; Tel.: +8-866-253-2106 ext. 5038; Fax: +8-866-243-4897.

Received: 5 January 2011; in revised form: 17 March 2011 / Accepted: 1 April 2011 / Published: 8 April 2011

Abstract: Wireless location is the function used to determine the mobile station (MS) location in a wireless cellular communications system. When it is very hard for the surrounding base stations (BSs) to detect a MS or the measurements contain large errors in non-line-of-sight (NLOS) environments, then one need to integrate all available heterogeneous measurements to increase the location accuracy. In this paper we propose a novel algorithm that combines both time of arrival (TOA) and angle of arrival (AOA) measurements to estimate the MS in NLOS environments. The proposed algorithm utilizes the intersections of two circles and two lines, based on the most resilient back-propagation (Rprop) neural network learning technique, to give location estimation of the MS. The traditional Taylor series algorithm (TSA) and the hybrid lines of position algorithm (HLOP) have convergence problems, and even if the measurements are fairly accurate, the performance of these algorithms depends highly on the relative position of the MS and BSs. Different NLOS models were used to evaluate the proposed methods. Numerical results demonstrate that the proposed algorithms can not only preserve the convergence solution, but obtain precise location estimations, even in severe NLOS conditions, particularly when the geometric relationship of the BSs relative to the MS is poor.

Keywords: time of arrival (TOA); angle of arrival (AOA); back-propagation nseural network (BPNN); resilient back-propagation (Rprop)

1. Introduction

The problem of position determination of a mobile user in a wireless network has been studied extensively in recent years. It has received significant attention and various location identification technologies have been proposed in the past few years. A recent report and order issued by the U. S. Federal Communications Commission (FCC) in July 1996 requires that all wireless service providers provide the location information to emergency 911 (E-911) public safety services. The separate accuracy requirements of the E-911 mandate were set for network-based technologies: within 125 meters for 67 percent of calls, and within 300 meters for 95 percent of the calls. To date, satisfying the FCC accuracy requirement is very difficult. Most papers and their algorithms could not achieve this goal.

The various techniques proposed include signal strength (SS), angle of arrival (AOA), time of arrival (TOA) and time difference of arrival (TDOA). The signal strength scheme uses a known mathematical model to describe the path loss attenuation with distance. A fuzzy logic technique with a geometrical solution was applied to calculate range estimates through signal strength measurements [1]. The AOA scheme estimates the signal direction of arrival [2], to derive the mobile station (MS) location, by using either a directive antenna, or an antenna array leading to multiple lines of position. The TOA location scheme measures the propagation time for a radio wave to travel between the MS and a base station (BS). The TDOA scheme determines the position of MS by examining just the difference in time from MS to multiple BSs, rather than the absolute arrival time. Different potential applications of wireless location services have been well developed, including the E-911 subscriber safety services, location-based billing, fleet management and intelligent transportation system (ITS) [3].

One critical problem in wireless location systems is the non-line-of-sight (NLOS) propagation effect. A common requirement for high location accuracy is the presence of a line-of-sight (LOS) path between the MS and each participating BS. Due to the signal reflection or diffraction between MS and BSs, NLOS errors can significantly impact wireless location performance. Extensive research on NLOS effect mitigation for location estimation have been carried out in the past few years. Since in NLOS the delay has a higher variance than under LOS conditions, [4] proposed a decision framework to detect NLOS BSs' via time series of estimates. An algorithm was proposed in [5] for TOA systems to mitigate NLOS effects by applying weights which are inversely proportional to their residuals for all possible BS combinations, then the NLOS BS with a larger residual has a lower effect on the MS location estimation. Similar residual schemes were proposed for both AOA systems in [6] and TDOA systems in [7]. The algorithms described in [5-7] perform well provided there are many available BSs being LOS with the MS in the system. Otherwise, they cannot improve the location accuracy. Based on the NLOS situation and how much *a priori* knowledge of the NLOS error is available, different NLOS identification and correction algorithms for mobile user location are proposed [8].

Another major concern that affects the choice of location scheme to deploy in cellular communication systems is *hearability* [9]. Here, *hearability* is the ability to receive signals from a sufficient number of BSs simultaneously at a sufficient power level [10]. It is difficult for the MS to receive signals from neighboring BSs in cellular communication systems, and the *hearability* is poor due to near-far effect and multiple access interference. [11] indicated the respective signal strength thresholds clearly show that the coverage in rural areas is much less than in urban areas. The likelihood of finding three BSs with received signal strength indication stronger than −100 dB is only 35 percent in rural areas, whereas it is about 84 percent in urban areas. It is difficult for an MS to detect three or more BSs for location purposes in rural areas. It is evident that the *hearability* in an IS-95 system is extremely poor [12]. The number of BSs that can be heard by a MS is only one if the MS is near its serving BS. Two or three BSs can be heard only at the edge of a cell. The insufficient number of available BSs limits the location-based services and impedes the implementation of location systems.

Due to poor *hearability*, it is reasonable to consider the hybrid methods by integrating two or more schemes. Comparing time-based methods with angle-based categories, both have their own advantages and limitations. In general, the AOA (angle-based) scheme requires only two BSs for location estimation, but it is necessary to deploy antenna array at BS for the AOA to work properly. In the case of time-based methods such as TOA and TDOA schemes, they require at least three BSs to be precisely located for a 2-D location estimation [13], nevertheless, they offer better accuracy than those of angle-based AOA schemes. The hybrid algorithm in [14] combining a TDOA technique with an additional AOA at the serving BS, can offer more accurate estimation in small error conditions. A hybrid TOA/AOA algorithm in [15], based on a nonlinear constrained optimization, can locate the MS even in NLOS environments for the case of three participating BSs. A hybrid range/range difference algorithm was used to estimate MS location in a global system for mobile communications (GSM) when only two BSs are available and the MS is located at the center of mass of the serving cell [16]. There is always an ambiguity in the MS location if only two TOA measurements are used. In order to resolve the ambiguity when an MS can be heard by only two BSs, both TOA and AOA information are required. Hence, we have proposed the hybrid TOA/AOA positioning methods for such *hearability*-constrained conditions in [17]. The MS is located at the various intersections of two circles and two lines, because the NLOS error always appears as a positive bias in the TOA measurements. Since the NLOS range errors are always positive, TOA measurements are greater than the true values. Therefore the true MS location should be inside the region enclosed by the overlap of the two circles. The above intersections are defined as feasible intersections. By using two AOA measurements to eliminate the least likely feasible intersection, the geometrical positioning methods are based on the weighted sum of the remaining feasible intersections enclosed by two TOA circles and two AOA line. These methods with different weighting algorithms can effectively eliminate the NLOS errors and provide more accurate positioning.

Artificial neural network (ANN) is an information processing method inspired by the biological nervous system, which can approximate nonlinear functions based on data sets. The system employs a set of activation functions and input-output of sample patterns that do not require *a priori* selection of a mathematical model. The back-propagation neural network (BPNN) is currently the most representative learning algorithm in ANN [18], and has been successfully applied to a wide range of scientific areas, especially in applications involving forecasting, image processing, pattern recognition and signal processing, and many others. BPNN continuously adjusts a set of weights of inputs and their corresponding outputs for the connections in the network to produce a mapping from input vectors to output vectors. It is an iterative algorithm using the gradient steepest descent method to minimize the error between the network actual output and desired output. This is particularly useful for those problems with an unknown optimal algorithm. The major drawbacks of the traditional BPNN are their slow learning process and have a tendency to be trapped into a local minimum.

Resilient back-propagation (Rprop) is the best algorithm in terms of convergence speed, accuracy as well as robustness with respect to the training parameters [19]. The Rprop is a local adaptive learning algorithm, the basic idea is to eliminate the harmful influence due to the weight step size of the partial derivative. Comparing to the back-propagation algorithm, the Rprop converges faster and needs less training. In this paper, we propose the algorithm based on Rprop to estimate MS location if both TOA and AOA measurements are simultaneously available from two BSs. We proposed two types for input data collection. The first type (divided type) establishes different input data subsets according to the number of the remaining feasible intersections. The second type (composite type) is to make the remaining feasible intersections in order. During the training period, the Rprop is trained to establish the nonlinear relationship between the remaining feasible intersections and the MS location. The remaining feasible intersections become the input data after training, and are passed through the trained Rprop to estimate the MS location. Simulation results show that the proposed algorithm always performs consistently better than Taylor series algorithm (TSA) [20,21], hybrid lines of position algorithm (HLOP) [22], and even the geometrical positioning methods proposed by us in [17]. Although it requires training, our algorithm are satisfied the FCC standard of accuracy in most of the NLOS error models. For both TSA and HLOP, the MS location accuracy can be critically affected by the relative geometry between BSs and MS. The proposed algorithm performs equally well for any MS location while TSA may not converge when the MS is located on the straight line passing through two of the BSs, and HLOP would produce a large location error when the measured angle is close to 90 ° or 270 °.

The remainder of this paper is organized as follows. In Section 2, we describe the MS positioning methods by using TSA and HLOP. The geometrical positioning methods are reviewed in Section 3. Section 4 briefly describes BPNN and Rprop methods. In Section 5, we propose the algorithm based on Rprop to determine the position of the MS. Next, Section 6 compares the performance of the proposed algorithm with the other methods through simulation result. Finally, Section 7 draws conclusions.

2. Taylor Series Algorithm (TSA) and Hybrid Lines of Position Algorithm (HLOP)

If both the TOA and AOA measurements are accurate, then only one BS is required to locate the MS [17]. In reality, both TOA and AOA measurements contain errors due to NLOS propagation. Thus more than one BS is required for MS location of reasonable accuracy. Taking into account the constraint on *hearability*, the number of BSs available for estimating MS location is limited to two in this paper. However, each BS has both TOA and AOA measurement capabilities. Let t_i denote the propagation time from the MS to BS*i*, *i* = 1, 2. The distances between BS*i* and MS can be expressed as:

$$r_i = c \cdot t_i = \sqrt{(x - X_i)^2 + (y - Y_i)^2} \tag{1}$$

where (*x*, *y*) and (X_i, Y_i) are the locations of MS and BS*i*, respectively. *c* is the propagation speed of the signals. If θ_i is the angle between MS and BS*i*, with respect to a reference direction (x-axis), BS1 is located at (X_1, Y_1) = (0, 0), BS2 is located at (X_2, Y_2) = (X_2, 0), and MS is located at (*x*, *y*), as shown in Figure 1, then θ_i can be obtained as:

$$\theta_i = \tan^{-1}(\frac{y - Y_i}{x - X_i}) \tag{2}$$

TSA [20,21] and HLOP [22] methods are commonly used to estimate the MS location, which are briefly described in this section.

2.1. Taylor Series Algorithm (TSA)

TOA and AOA measurements are used as inputs to the Taylor series position estimator. Let (*x*, *y*) is the MS location and (x_v, y_v) is the initially estimated position, let $x = x_v + \delta_v$ and $y = y_v + \delta_v$. The MS location is obtained by linearizing the TOA and AOA equations through the use of a Taylor series expansion and retaining second-order terms, we have:

$$A\delta \cong z \tag{3}$$

where $A = \begin{bmatrix} \kappa_{11} & \kappa_{12} \\ \kappa_{21} & \kappa_{22} \\ \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix}$, $\delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}$, $z = \begin{bmatrix} r_1 - r_{v1} \\ r_2 - r_{v2} \\ \theta_1 - \theta_{v1} \\ \theta_2 - \theta_{v2} \end{bmatrix}$, and $\kappa_{i1} = \left.\frac{\partial r_i}{\partial x}\right|_{x_v, y_v}$, $\kappa_{i2} = \left.\frac{\partial r_i}{\partial y}\right|_{x_v, y_v}$,

$r_{vi} = \sqrt{(x_v - X_i)^2 + (y_v - Y_i)^2}$, $\phi_{i1} = \left.\frac{\partial \theta_i}{\partial x}\right|_{x_v, y_v}$, $\phi_{i2} = \left.\frac{\partial \theta_i}{\partial y}\right|_{x_v, y_v}$, $\theta_{vi} = \tan^{-1}\frac{y_v - Y_i}{x_v - X_i}, i = 1, 2$.

The least-squares (LS) solution to the estimation problem is given by:

$$\delta = (A^T A)^{-1} A^T z \tag{4}$$

It requires a proper initial position guess close to the true solution and can achieve high accuracy. This method is recursive and the computational overhead is intensive in the iteration. Due to the initial guess of the MS location is not accurate enough, the convergence of the iterative process is not assured [20,21]. In addition, to avoid the divergent problems in the simulations, the true MS location is used as an initial position.

2.2. Hybrid Lines of Position Algorithm (HLOP)

This scheme makes use of the original nonlinear range equations to produce linear lines of position (LOP), rather than circular LOP, to locate the MS. The method takes the advantage of simpler computation of MS location. The details of the linear LOP approach can be acquired by using the TOA measurements as in [23], and the hybrid linear LOP algorithm with AOA measurement in [22]. Combining the linear LOP and two AOA lines, the MS location can be determined by:

$$Gl = h \tag{5}$$

where $l = \begin{bmatrix} x \\ y \end{bmatrix}$ denotes the MS location, $G = \begin{bmatrix} X_2 & 0 \\ \tan\theta_1 & -1 \\ \tan\theta_2 & -1 \end{bmatrix}$ and $h = \frac{1}{2}\begin{bmatrix} r_1^2 - r_2^2 + X_2{}^2 \\ 0 \\ 2(X_2 \cdot \tan\theta_2) \end{bmatrix}$.

Again, the LS solution to Equation (5) is given by:

$$l = (G^T G)^{-1} G^T h \tag{6}$$

3. Geometrical Positioning Methods

From the viewpoint of geometric approach, the TOA value measured at any BS can be used to form a circle centered at the BS. The MS position is then given by the intersection of the circles from multiple TOA measurements. Similarly, a single AOA measurement constrains the MS along a line. Each of the following equations describes a circle for TOA, a line for AOA, as shown in Figure 1:

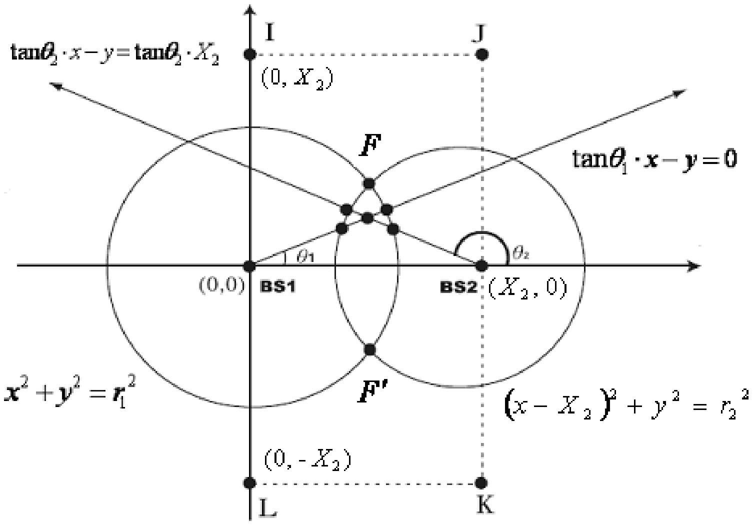
$$\text{Circle 1: } x^2 + y^2 = r_1^2 \tag{7}$$

$$\text{Circle 2: } (x - X_2)^2 + y^2 = r_2^2 \tag{8}$$

$$\text{Line 1: } \tan\theta_1 \cdot x - y = 0 \tag{9}$$

$$\text{Line 2: } \tan\theta_2 \cdot x - y = \tan\theta_2 \cdot X_2 \tag{10}$$

Figure 1. Geometric layout of the two circles and two lines.


I
J
$\tan\theta_2 \cdot x - y = \tan\theta_2 \cdot X_2$
$(0, X_2)$
F
$\tan\theta_1 \cdot x - y = 0$
θ_1
θ_2
(0,0)
BS1
BS2
$(X_2, 0)$
$x^2 + y^2 = r_1^2$
F'
$(x - X_2)^2 + y^2 = r_2^2$
$(0, -X_2)$
L
K

If there is no error or even no noise at all, the circles and lines will intersect at only one point. However, this is usually not the case in practice where the NLOS effect exists. NLOS propagation is quite common and it seriously degrades location accuracy. The intersections of two TOA circles and two AOA lines will be spread over a region, which will be offset from the true MS location. Because of the fact that NLOS effect always increases the propagation delay, the measured TOA estimated are always greater than the true values due to the excess path length. The true MS location must lie in the region of overlap of the two circles. As mentioned earlier, the intersecting points that are within this are defined as feasible intersections. Hence, the feasible intersections must satisfy the following inequalities simultaneously:

$$x^2 + y^2 \le r_1^2 \tag{11}$$

$$(x - X_2)^2 + y^2 \le r_2^2 \tag{12}$$

The most direct method is to utilize these feasible intersections of the circles and lines to estimate the MS location. To achieve high accuracy of MS location with less complexity, we have proposed a class of geometrical positioning methods in [17] and outlined as follows.

3.1. Averaging Method

By using two AOA measurements, the least likely intersection is first eliminated. The MS location is obtained to calculate the average value of all the remaining AOA measurements with feasible intersections.

Step 1. Find all the feasible intersections of the two circles and two lines.

Step 2. Assume *F* and *F'* are the intersections of the two circles as shown in Figure 1, *F'* is considered to be the least likely intersection if 0° < θ_1, θ_2 < 180°, and *F* is considered to be the least likely intersection if 180° < θ_1, θ_2 < 360°. Delete the least likely intersection from the set of feasible intersections and there will be N remaining feasible intersections.

Step 3. The MS location $(\bar{x}_N, \bar{y}_N)$ is estimated by averaging these remaining feasible intersections, where:

$$\bar{x}_N = \frac{1}{N}\sum_{i=1}^{N} x_i \text{ and } \bar{y}_N = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{13}$$

3.2. Distance-Weighted Method

However, not all the remaining feasible intersections can always provide information of the same value for location estimation. In this method, the weights are inversely proportional to the squared value of the distance between the remaining feasible intersections and the average MS location.

Steps 1–3 are the same as those of the averaging method.

Step 4. Calculate the distance d_i between each remaining feasible intersection (x_i, y_i) and the average location $(\bar{x}_N, \bar{y}_N)$:

$$d_i = \sqrt{(x_i - \bar{x}_N)^2 + (y_i - \bar{y}_N)^2},\ 1 \le i \le N \tag{14}$$

Step 5. Set the weight for the *i*th remaining feasible intersection to $(d_i^2)^{-1}$. Then the MS location (x_d, y_d) is determined by:

$$x_d = \frac{\sum_{i=1}^{N} \left(d_i^2\right)^{-1} \cdot x_i}{\sum_{i=1}^{N} \left(d_i^2\right)^{-1}} \text{ and } y_d = \frac{\sum_{i=1}^{N} \left(d_i^2\right)^{-1} \cdot y_i}{\sum_{i=1}^{N} \left(d_i^2\right)^{-1}} \tag{15}$$

One can see in the averaging method and distance-weighted method, all the remaining feasible intersections will affect the MS location estimation. In the following we also propose two methods of sort averaging and sort-weighted, which can be applied without considering the influence of feasible intersections for too far away from the average MS location.

3.3. Sort Averaging Method

Steps 1–4 are the same as those of the distance-weighted method.

Step 5. Rank the distances d_i in increasing order and re-label the remaining feasible intersections in this order.

Step 6. The MS location $(\bar{x}_M, \bar{y}_M)$ is estimated by the mean of the first M remaining feasible intersections:

$$\bar{x}_M = \frac{1}{M}\sum_{i=1}^{M} x_i\ ,\ \bar{y}_M = \frac{1}{M}\sum_{i=1}^{M} y_i\ \ (M = 0.5 * N \le N) \tag{16}$$

3.4. Sort-Weighted Method

Steps 1–5 are the same as those of the sort averaging method.

Step 6. The MS location is estimated by a weighted average of the first M remaining feasible intersections with weight = $(d_i^2)^{-1}$:

$$x = \frac{\sum_{i=1}^{M}\left(d_i^{\,2}\right)^{-1} \cdot x_i}{\sum_{i=1}^{M}\left(d_i^{\,2}\right)^{-1}},\ y = \frac{\sum_{i=1}^{M}\left(d_i^{\,2}\right)^{-1} \cdot y_i}{\sum_{i=1}^{M}\left(d_i^{\,2}\right)^{-1}}\ \ (M = 0.5 * N \le N) \tag{17}$$

3.5. Threshold Method

The weight of this method is based on how close the remaining feasible intersections are. Those feasible intersections that are closer to one another are assigned with greater weights. In other words, those intersections that are in close proximity will be assigned with greater weights.

Steps 1 and 2 are the same as those of the averaging method.

Step 3. Calculate the distance d_{mn}, $1 \le m$, $n \le N$, between any pair of feasible intersections.

Step 4. Select a threshold value D_{thr} as the average of all the distances d_{mn}.

Step 5. Set the initial weight I_k, $1 \le k \le N$, to be zero for all remaining feasible intersections.

If $d_{mn} \le D_{thr}$, then $I_m = I_m + 1$ and $I_n = I_n + 1$ for $1 \le m,\ n \le N$.

Step 6. The MS location (x_t, y_t) is estimated by:

$$x_t = \frac{\sum_{i=1}^{N} I_i \cdot x_i}{\sum_{i=1}^{N} I_i} \text{ and } y_t = \frac{\sum_{i=1}^{N} I_i \cdot y_i}{\sum_{i=1}^{N} I_i} \tag{18}$$

4. The Traditional BPNN Algorithm and the Rprop Algorithm

4.1. The Traditional BPNN Algorithm

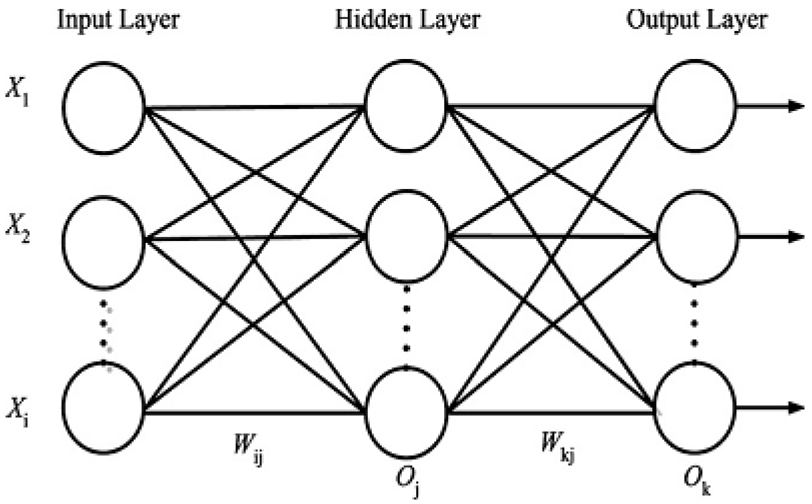
In this section, we describe the methodology based on artificial neural network (ANN). It is a technique that models the learning procedures of a human brain, and employs a set of activation functions, either nonlinear or linear, thus one doesn't require *a priori* selection of a mathematical model. Further, this method has been proved to be very useful for various applications. One of the most influential developments in ANN was the invention of the BPNN, which provides advantages of non-linear problem solving ability. BPNN is a multi-layered, feed-forward architecture with supervised learning method for computer learning and modeling. A supervised feed-forward neural network can not only learn from the training data to discover patterns representing the input and output variables, but approximate many problems with high accuracy. In a supervised learning approach, a set of input variables is used for which the corresponding output variables are known.

Generally speaking, the BPNN architecture comprises one input layer, one output layer, with one or a number of hidden layers in between them. Although a network with multiple hidden layers is possible, a single layer is sufficient to model arbitrarily complex nonlinear functions. With proper selection of architecture, it is capable of approximating most problems with high accuracy and generalization ability. The input layer receives information from the external sources and passes this information to the network for processing. The hidden layer determines the mapping relationships between neurons are stored as weights of connecting links. When the input and output variables are related nonlinearly, the hidden layer can extract higher level features and facilitate generalization. The output from the output layer is the prediction of the net for the corresponding input. The structure of BPNN chosen for the present problem is shown in Figure 2. Each layer consists of several neurons and the layers are interconnected by sets of correlation weights. A standard ANN comprises numerous simple processing units called neurons. Each node is connected to other neurons through directed connecting links; each neuron is a processing unit that contains an activation function and an associated weight. The active function is mathematical formula and used to transform the output such that it falls within an acceptable range. In this paper, the activation functions of hidden layer and output layer are hyperbolic tangent sigmoid function and linear transfer function. A weight returns a mathematical value for the relative strength of connections to transfer data from one layer to another layer.

BPNN estimate relation between input and output of sample patterns by updating iteratively the weights in the network so as to minimize the difference between the actual output vectors and the desired output vectors. The back propagation learning algorithm is composed of initialization, a forward pass, and a backward pass. The weights and biases in the network are initialized to small random numbers. Once these parameters have been initialized, the network is ready for training. A training pattern consists of a set of the input vectors and the corresponding output vectors. In the beginning, a set of training patterns are fed to the input layer of the network. The forward pass starts from the input layer, the net inputs of the neurons are multiplied with corresponding weights, then summated, and transferred to the hidden layer. The activated signals are outputted from the hidden layer, and are passed forward to the output layer. Finally, the output of BPNN is generated. Subsequently in the backward pass, the error between actual output and desired output is calculated. The error function Ψ is defined as the mean squared sum of differences between the actual output vector T_k and the desired output vector O_k:

$$\Psi = \frac{1}{2}\sum_{k}(T_k - O_k)^2 \tag{19}$$

Figure 2. A fully connected multilayer feed-forward network with one hidden layer.



The error signal at the output layer is propagated backward to the input layer through the hidden layer in the network. Back-propagation is so named because the error derivatives are calculated in the opposite direction of signal propagation. In the training process, the gradient descent method calculates and adjusts the weight of the network to minimize the error. In the weight updating algorithm, the derivative of the error with respect to the weight was first negated then multiplied by a small constant β known as the learning rate, as expressed in the following equation:

$$\Delta w_{ij}^{(t)} = -\beta \frac{\partial \Psi^{(t)}}{\partial w_{ij}} \tag{20}$$

The negative sign indicates that the new weighting vector is moving in a direction opposite to that of the gradient. In the learning process of neural network, the learning rate affects the speed of convergence. The training process may lead to an oscillatory state if a learning rate is too fast, on the other hand, the convergence speed may suffer if the learning rate is too slow. The training process may not converge in the case of either a too high or too low value for the learning rate *β*. To accelerate the convergence, a momentum *α* can be added to the learning procedures [24]:

$$\Delta w_{ij}^{(t)} = -\beta \frac{\partial \Psi^{(t)}}{\partial w_{ij}} + \alpha \Delta w_{ij}^{(t-1)} \tag{21}$$

In Equation (21) *α* is between 0 and 1. When *α* = 0, a weight change is completely dependent on the value of gradient. When *α* = 1, the amount of new weight change is set to that of the last weight change and the gradient is simply ignored. The weights are adjusted to make the actual output move closer to the desired output and to obtain the final outputs. This process is repeated until the error is less than a pre-specified level for each of the training data points, or a large number of training iterations have already been run. In summary, the flow chart of training procedure of BPNN is in Figure 3 and the steps are listed as follows.

Set the number of the layer and the number of neurons in each layer:

(1) Set *β*, *α* and initial values of the weights, and the biases in the network are initialized to small random numbers.
(2) Giving input and output vectors.
(3) Compute the output values of each layer and unit in a feed-forward direction.

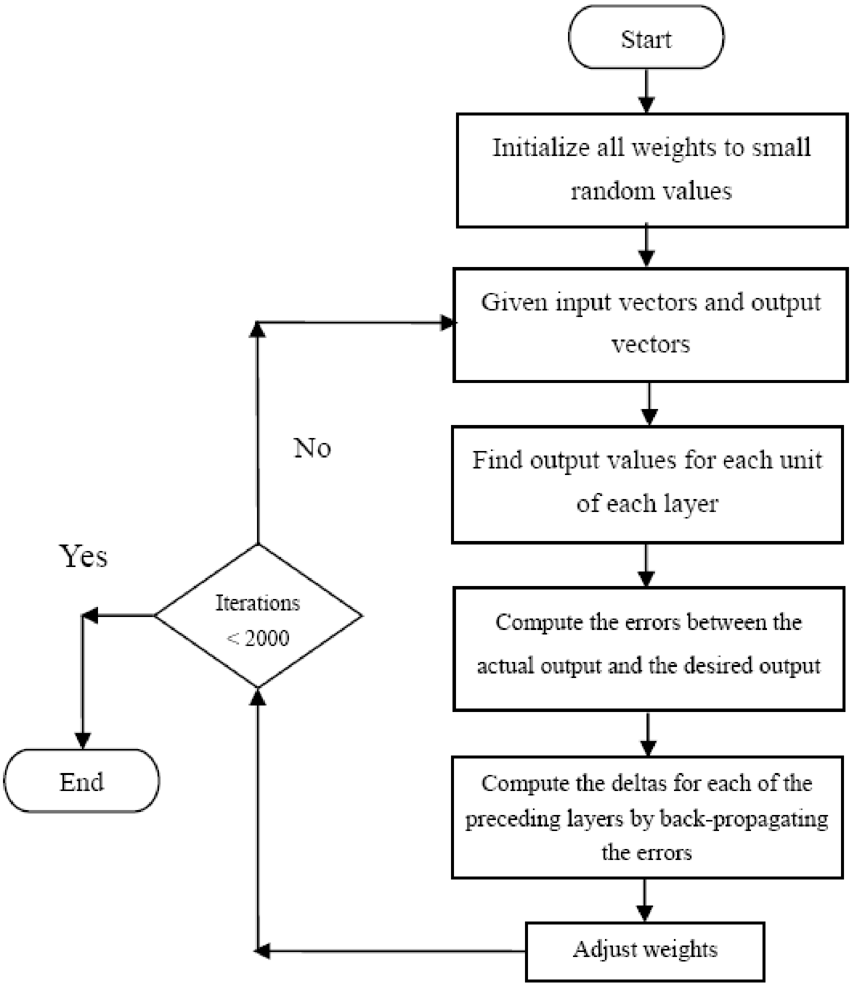
(i) Calculate the output for the *j*th hidden neuron.
(ii) Calculate the output for the *k*th output neuron.

(4) Calculate the error function at the output neuron.
(5) Compute the deltas for each of the preceding layers by back propagating the errors.

(i) Calculate error for the *k*th output neuron.
(ii) Calculate error for the *j*th hidden neuron.

(6) Update all weights and biases
(7) Repeat steps 3-7 until the iteration has finished or the algorithm is convergent.

Figure 3. The flow chart of the calculation procedure for BPNN.


Start
Initialize all weights to small random values
Given input vectors and output vectors
Find output values for each unit of each layer
Compute the errors between the actual output and the desired output
Compute the deltas for each of the preceding layers by back-propagating the errors
Adjust weights
Iterations < 2000
No
Yes
End

4.2. Rprop Algorithm

Compared to the traditional BPNN algorithm, the Rprop algorithm can provide faster training and rate of convergence, and has the capability to escape from local minima. The Rprop is known to be very robust with respect to their internal parameters and therefore regarded as one of the best first-order learning methods among the ANN algorithms. Rprop is a first-order algorithm and its time and memory requirement scale is linear with the number of parameters to optimize. The Rprop algorithm is probably the most easily adjustable learning rule, slight variations of the values of parameters can not affect the convergence time. The activation function of the hidden and output layers is treated as linear transfer function. Rprop is easy to implement and the hardware implementation is described in [25]. Comparing to back-propagation, one of the advantages of Rprop algorithm is that the magnitude of the partial derivative does not affect weight update and it depends only on the signs of the partial derivative. Thus it allows for faster convergence than the back-propagation can do. Rprop performs a direct adaptation of the weighting step based on local gradient information. A crucial difference to the previously developed adaptation techniques is that the adaptation effort won’t be blurred by the gradient behavior. The main idea of Rprop is to reduce the potential spurious effect of the partial derivative on weight-updates by retaining only the sign of the derivative as an indication of the direction in which the error function will be changed by the weight-update. We introduce an individual update-value $\Delta_{ij}(t)$ for each weight, which solely determines the size of the weight-update. This adaptive update-value evolves during the learning process based on its local sight on the error function Ψ, according to the following learning rule [19]:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^{+} \cdot \Delta_{ij}^{(t-1)} & , \text{ if } \dfrac{\partial \Psi}{\partial w_{ij}}^{(t-1)} \cdot \dfrac{\partial \Psi^{(t)}}{\partial w_{ij}} > 0 \\ \eta^{-} \cdot \Delta_{ij}^{(t-1)} & , \text{ if } \dfrac{\partial \Psi}{\partial w_{ij}}^{(t-1)} \cdot \dfrac{\partial \Psi^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)} & , \ else \end{cases} \quad (22)$$

where $0 < \eta^{-} < 1 < \eta^{+}$. We can simply describe the adaptation rule as follows: Whenever the partial derivative of the error function ψ with respect to the corresponding weight w_{ij} changes its sign, it indicates that the value of last update was too big and the algorithm has jumped over a local minimum. The update-value Δ_{ij} is decreased by a factor η^{-}. If the derivative retains its sign, the update-value is slightly increased by the factor η^{+} in order to accelerate convergence in shallow regions.

Once the update-value for each weight is adapted, the weight-update itself follows a very simple rule: if the derivative is positive (increasing error), the weight is decreased by its update-value, if the derivative is negative, the update-value is added to the weight:

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & , \text{ if } \dfrac{\partial \Psi}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)} & , \text{ if } \dfrac{\partial \Psi}{\partial w_{ij}}^{(t)} < 0 \\ 0 & , \ else \end{cases} \quad (23)$$

There is one exception to the rule above. If the partial derivative changes sign, *i.e*., the previous step was too large and the minimum was missed, the previous weight-update is reverted:

$$\Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)}, \text{ if } \frac{\partial \Psi^{(t)}}{\partial w_{ij}} \cdot \frac{\partial \Psi^{(t-1)}}{\partial w_{ij}} < 0 \tag{24}$$

Due to that 'backtracking' weight step, the derivative is supposed to change its sign once again in the following step. In order to avoid a double punishment of the update value, there should be no adaptation of the update value in the succeeding step. In practice this can be done by setting $\partial\psi^{(t-1)}/\partial w_{ij} = 0$ in the Δ_{ij} update-rule above.

5. Proposed Location Algorithm Based on Rprop

To improve the accuracy of MS location, we proposed the employment of Rprop, a supervised learning neural network to obtain an approximation of MS location. The remaining feasible intersections are fed to the input layer, and MS location is the only one variable in the output layer. Given a number of known input-output training patterns, the Rprop models are trained continuously and deployed to adjust the weights with one hidden layer. A trained Rprop is to minimize the difference between the actual MS location and the desired MS location. The network has the following input-output mapping:

Input: V remaining feasible intersections (V = 1, 2,…, 6)
Output: desired MS location

The number of the remaining feasible intersections depends on the geometric relationship of the two TOA circles and two AOA lines. In this case, the number of the remaining feasible intersections is between 1 and 6. Every measurement will result in one input data. Figure 4 illustrates the structure used in Rprop MS location forecasting model of a three-layered network. Two types of input layer for training purpose are identified and explained in detail as follows:

Type 1 (Divided Type):

According to the number of the remaining feasible intersections, the first type establishes different input data subsets respectively. For each measurement, we collect the V remaining feasible intersections and put them into the V-th input data subsets separately. There are six data subsets in this input layer for training purpose, and the measurement number of each subset won't be identical. From simulation results, the measurement number of 4 remaining feasible intersections is the maximum, while the measurement number of 1 remaining feasible intersections is the minimum.

The detailed steps are as follows:

(1) Collect the V remaining feasible intersections of two TOA circles and two AOA lines. (V = 1, 2,…, 6).

(2) If the number of remaining feasible intersection is V, then placing these V points in the V-th subset. The V remaining feasible intersections are belonging to the corresponding V-th input data subsets separately.

(3) The 6 input data subsets with various measurement numbers are trained according to Rprop.

Type 2 (Composite Type):

The second type is a collection of the V remaining feasible intersections in order. Regardless of the number of remaining feasible intersections in each measurement, we will only establish one input data set. The summation of all the measurement number for the 6 subsets is equal to the number of all measurements. The detailed steps are as follows:

(1) Collect the V remaining feasible intersections for each measurement and expand to six ones in a data set.

(2) The method to expand the remaining feasible intersections to 6 ones during each measurement is as follows.

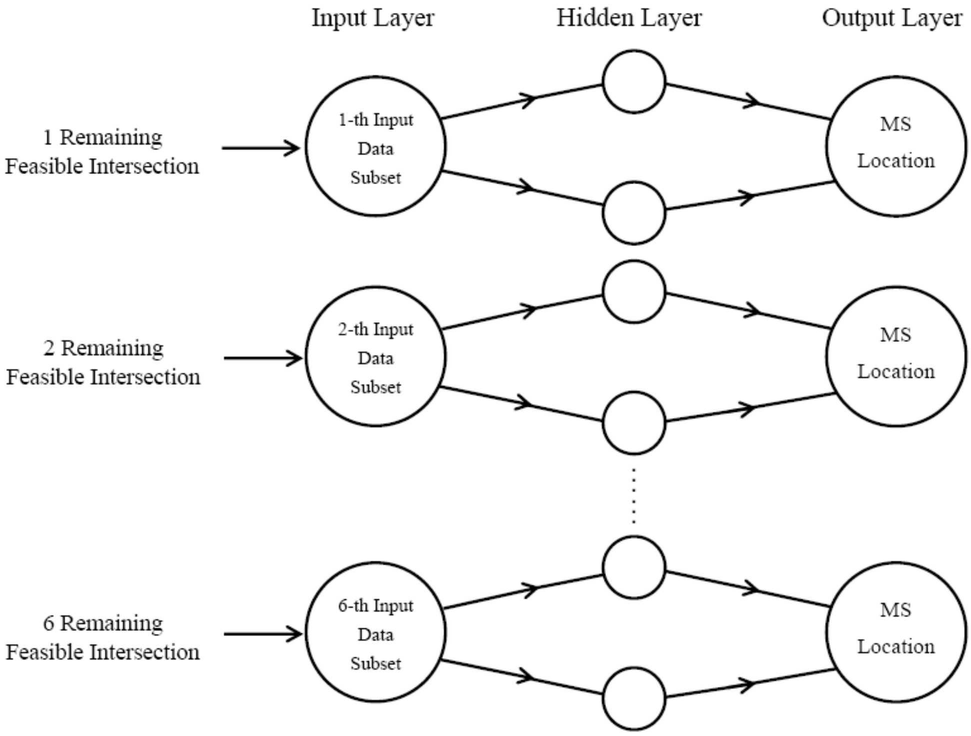
(i) If the number of remaining feasible intersections is V, replicate them by (6/V) times. (V = 1, 2, 3)

(ii) If the number of remaining feasible intersections is 4, take the average value of these 4 points and treat it as the fifth point. By this manner the 6th point is the average of the 5 previous numbers.

(iii) If 5 remaining feasible intersections are collected, take an average of these 5 points as the 6th one.

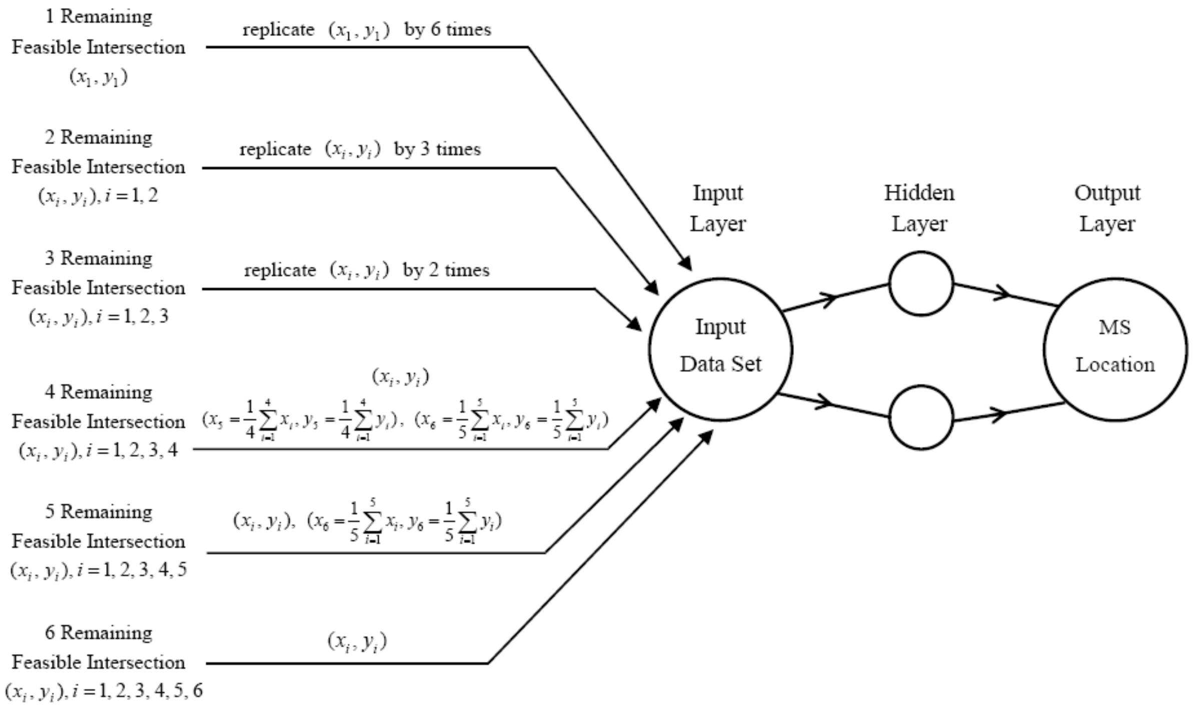
(3) After expansion, placing the 6 remaining feasible intersections in the input data set for training purposes.

Figure 4. Structure of the prediction models for **(a)** Type 1 (Divided Type) and **(b)** Type 2 (Composite Type).



(a)

Figure 4. *Cont.*


1 Remaining
Feasible Intersection
(x_1, y_1)
replicate (x_1, y_1) by 6 times
2 Remaining
Feasible Intersection
$(x_i, y_i), i = 1, 2$
replicate (x_i, y_i) by 3 times
3 Remaining
Feasible Intersection
$(x_i, y_i), i = 1, 2, 3$
replicate (x_i, y_i) by 2 times
4 Remaining
Feasible Intersection
$(x_i, y_i), i = 1, 2, 3, 4$
(x_i, y_i)
$(x_5 = \frac{1}{4}\sum_{i=1}^{4} x_i, y_5 = \frac{1}{4}\sum_{i=1}^{4} y_i)$, $(x_6 = \frac{1}{5}\sum_{i=1}^{5} x_i, y_6 = \frac{1}{5}\sum_{i=1}^{5} y_i)$
5 Remaining
Feasible Intersection
$(x_i, y_i), i = 1, 2, 3, 4, 5$
(x_i, y_i), $(x_6 = \frac{1}{5}\sum_{i=1}^{5} x_i, y_6 = \frac{1}{5}\sum_{i=1}^{5} y_i)$
6 Remaining
Feasible Intersection
$(x_i, y_i), i = 1, 2, 3, 4, 5, 6$
(x_i, y_i)
Input Layer
Hidden Layer
Output Layer
Input Data Set
MS Location

(b)

The training data is different from the data that uses to estimate the MS location. That is, the training input-output patterns is no longer be used after training is down. In real application, we collect the remaining feasible intersections and the desired MS location to train the neural network prior to the practical use. After the training, then the remaining feasible intersections as input data (with the MS locations be unknown) can not only pass through the trained Rprop more quickly, but estimate the better appropriate MS location. Whenever we start to find the positions, the "remaining feasible intersections" can be used as the trained input, as we expect this model can estimate MS locations quickly and precisely. In addition, we have found that when there are only 200 pieces of input-output patterns as training data, the proposed algorithm still work better than the other methods. Therefore, the conclusion is that the proposed algorithm can be applied in practical situations.

6. Simulations Results

In this section for fairly comparison with various methods we apply the computer simulations to demonstrate the performance of the proposed algorithm. In the simulations, the BSs are respectively located at (0, 0) and (2,000 m, 0). Each simulation is performed by 10,000 independent runs, and the MS location is chosen randomly according to a uniform distribution within the rectangular area formed by the points I, J, K and L, as shown in Figure 1. To validate the effectiveness of the proposed algorithm, the remaining feasible intersections and the desired MS location are collected for comparison. The proposed hybrid TOA/AOA algorithm employs Rprop for MS location estimation and performance evaluation. Regarding the NLOS effects, three propagation models are adopted, namely, the uniformly distributed noise model [15], the biased uniform random model [22], as well as the distance-dependent model [15].

The first NLOS propagation model is based on the uniformly distributed noise model [15], in which the TOA measurement error is assumed to be uniformly distributed over $(0, U_i)$, for $i = 1, 2$ where U_i is the upper bound and the AOA measurement errors are assumed to be $f_i = w\,5°$ and $f_i = w \cdot 10°$, where w is a uniformly distributed variable over [−1, 1] [26]. Before applying Rprop to estimate MS location, the parameter must be set in advance, such as the numbers of hidden neurons, and training iterations (epochs). The parameter settings for network architectures must be determined carefully to avoid constructing a worse network model; otherwise, they may increase the computational cost and produce worse results. Trial-and-error methods are used to determine the parameter settings for network architectures. We attempted to find the optimal parameter as well as maintain good performance at the same time.

Single hidden layer is the most widely used one among various learning methods for neural networks. It is well enough to model arbitrarily complex nonlinear functions [27]. Therefore, the number of hidden layers is set at one. To examine how close the forecast to the real MS location, the root-mean-square (RMS) error is employed to evaluate the performance of the proposed algorithm. If the network outputs are relatively close to the real MS location, RMS error will have small values. Figure 5 shows the RMS error of convergence *versus* the increased number of epochs. One can see the Rprop with one hidden layer can map the remaining feasible intersections to MS location. At the beginning of the training period, the error is reduced rapidly. When the number of epochs is above 2,000, it will not give further performance improvement. Some general rules to determine the number of hidden neurons are: (i) $0.5\,(p + q)$, (ii) p, (iii) $2p + 1$, (iv) $3p + 1$, where p and q are input neurons and output neurons, respectively [28]. The RMS errors with various numbers of hidden-layer neurons are listed in Figure 6 for comparison. One can see the RMS error converged to the same minimum value for various hidden layer neurons. The accuracy of MS location is hardly affected by the numbers of hidden-layer neurons. The number of neurons in the hidden layer is set to be 2 because of the satisfactory prediction performance. In order to avoid increasing the computation load, 2 hidden neurons and 2,000 training iterations are used in the simulations.

If BS1 is the serving BS of MS, its TOA and AOA measurements should be more accurate. The variables of this model are chosen as follows: $U_1 = 200$ m, $U_2 = 400$ m, $\tau_1 = 5°$ and $\tau_2 = 10°$. The proposed algorithm based on Rprop algorithm produce more accurate estimations of MS location than those based on BPNN with a learning rate of 0.01, as shown in Figure 7. We can see that BPNN-based with 2,000 epochs has the worst performance and Rprop-based with 2,000 epochs has the best performance. Even Rprop-based with 200 epochs perform better than BPNN-based with 2,000 epochs. The number of epochs of Rprop is less than the traditional BPNN. So Rprop method can offer much faster convergence and require much less convergence time. The results show that Rprop algorithm can accurately estimate MS location. The superior performance for the Rprop algorithm has been demonstrated by comparing the RMS errors. One can see that Rprop algorithm produces more accurate estimations of MS location than traditional BPNN algorithm. Therefore, we select Rprop algorithm to estimate MS location in this paper.

Figure 5. RMS errors reduction according to the number of epochs.

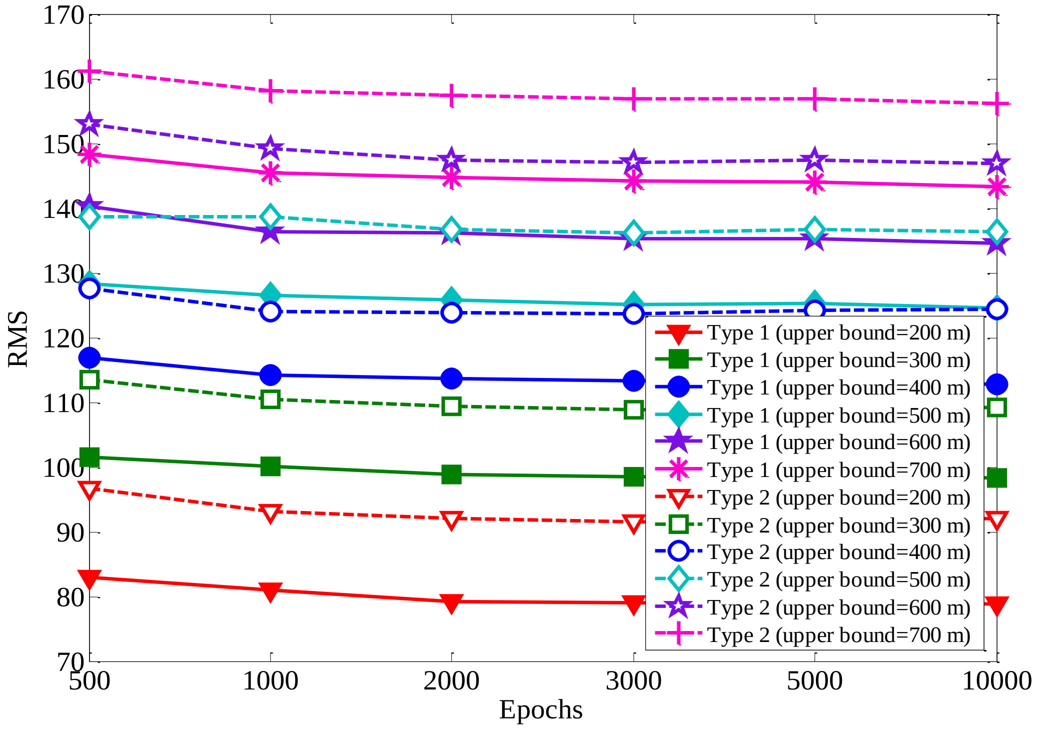


Figure 6. The RMS errors with various neurons numbers of hidden layer.

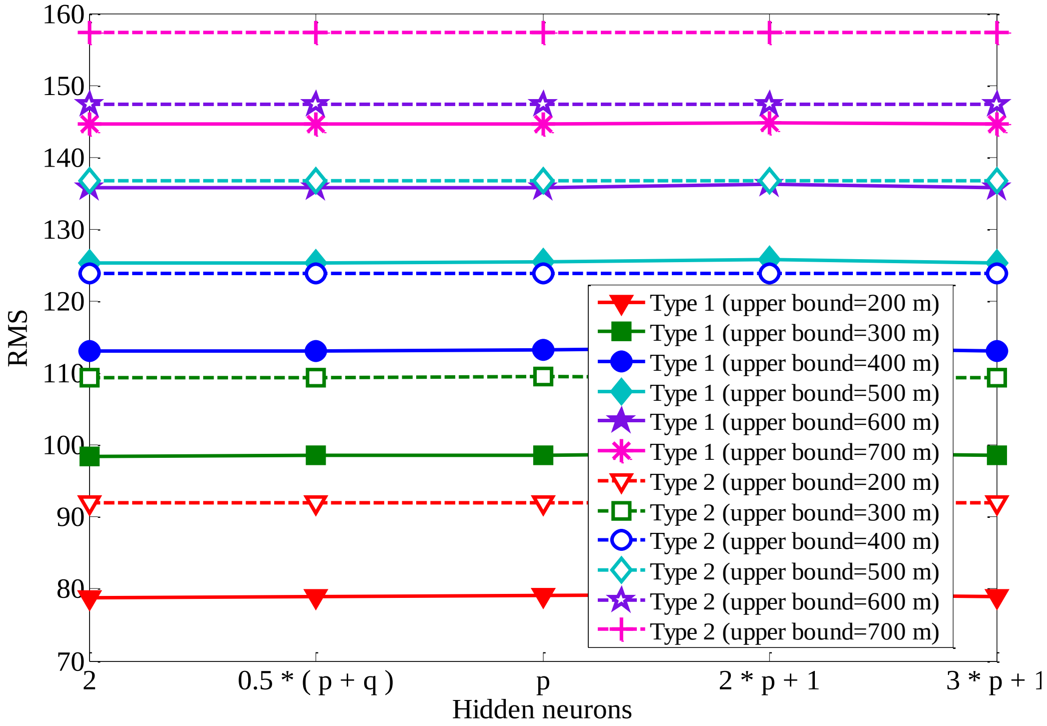


Figure 7. Comparison of average MS location based on BPNN and Rprop.

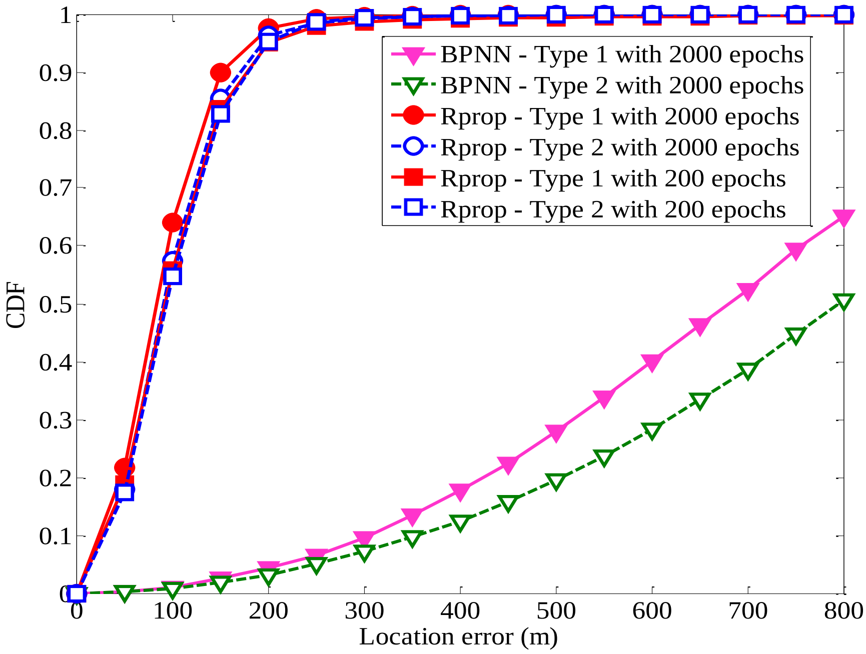
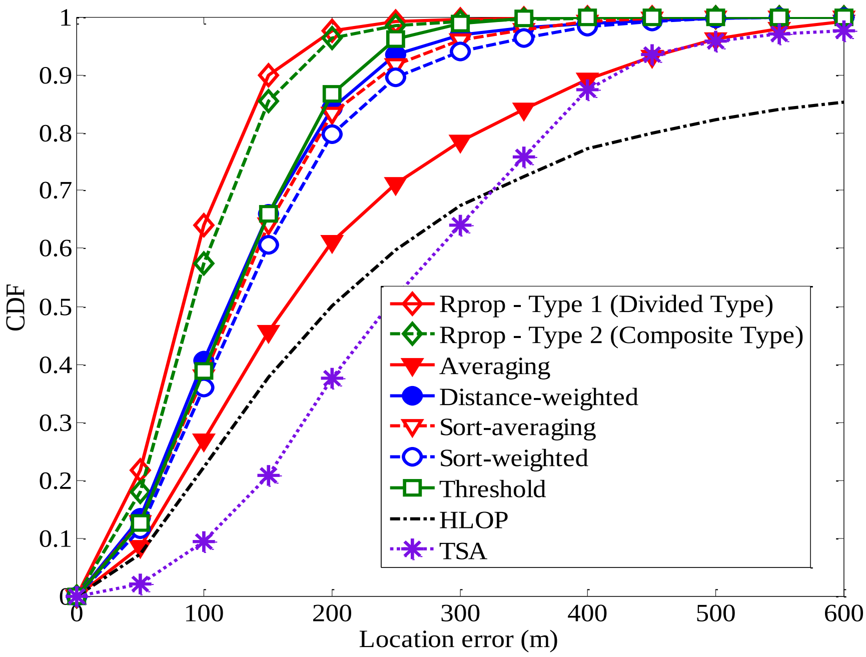


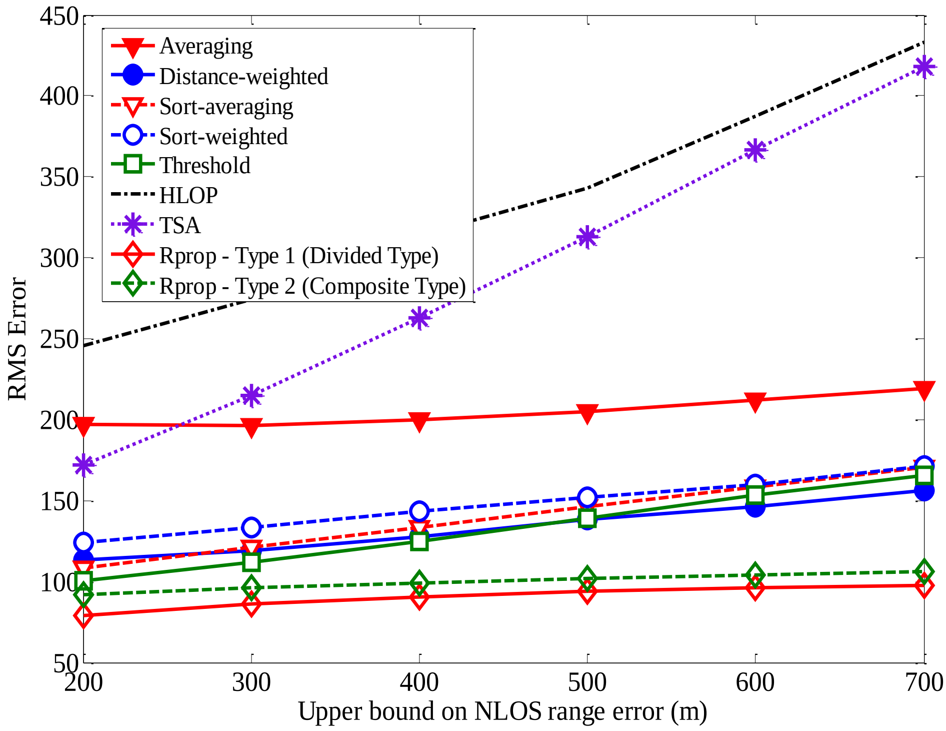
Figure 8 shows cumulative distribution functions (CDFs) of the location error for different methods based on uniformly distributed noise model. To check for convergence, the initial guess of MS location in TSA is chosen to be the true solution in our simulations. Simulations demonstrated that at least three iterations are required for TSA to converge. We can see that TSA and HLOP offer the worst performance, and the proposed algorithm has good ability to make more accurate estimations of MS location. The divided type with six input data subsets for training performs better than the composite type. Thus the proposed algorithm can always yield better performance, compared with TSA, HLOP and the other geometrical positioning methods.

Figure 8. Comparison of error CDFs when NLOS errors are modeled as the upper bound.



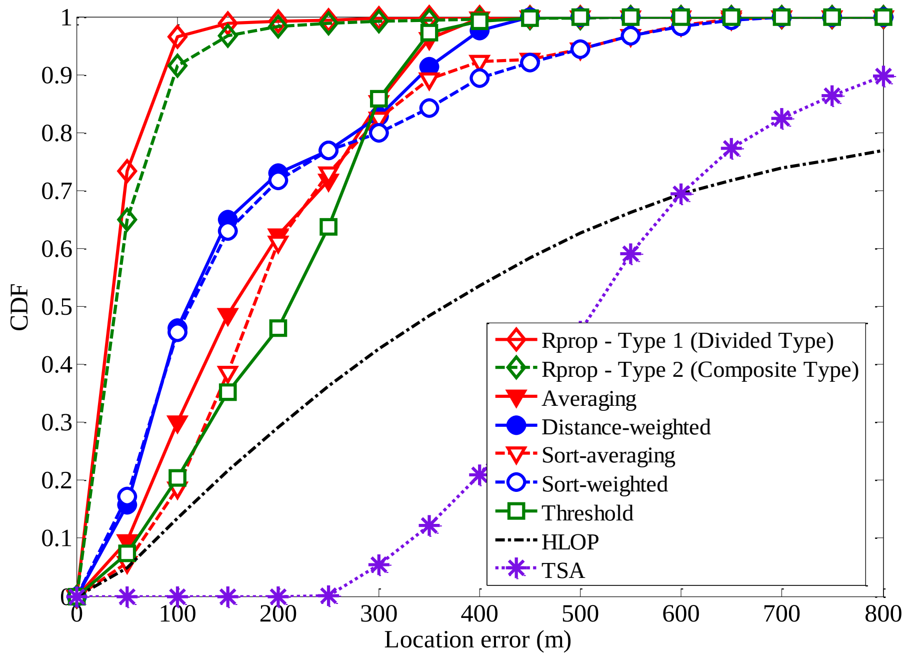
Based on the proposed neural network structure stated above, the Rprop can be applied to estimate the location of MS for every input data. Figure 9 shows how the average location error is affected by the upper bound on NLOS error. The upper bound of NLOS error for BS1 is 200 m and those of other BSs range from 200 m to 700 m. After the training period, the superior performance for the proposed algorithm can be proved by comparing the RMS error of MS location. In general, as the upper bound on NLOS error increases, then the average magnitudes of the NLOS errors will also increase, which leads to less accurate location estimation. Note that the proposed algorithm can deal with large errors more effectively than the other methods. For most of the upper bound on NLOS errors, the average location errors for TSA and HLOP are larger than at least twice that for the proposed algorithm. The sensitivity of the proposed algorithm with respect to the NLOS effect is much less than those of TSA and HLOP methods.

Figure 9. Performance comparison of the location estimation methods when the upper bound is used to model the NLOS.



The second NLOS propagation model is based on the distance-dependent NLOS error model [15]. The NLOS range error for the *i*th range is taken to be $\xi_i = \chi_i R_i$, where χ_i is a proportional constant and R_i is the true range between *i*-th BS and MS [15]. It makes intuitive sense to view NLOS errors as being proportional to the distance traveled by the signal. The AOA measurement error is assumed to be $f_i = w\, \tau_i$, for $i = 1, 2$ [26]. The variables are chosen as follows: $\chi_1 = 0.13$, $\chi_2 = 0.2$, $\tau_1 = 2.5°$ and $\tau_2 = 5°$ Figure 10 shows the CDF of the average location error of different algorithms with distance-dependent NLOS error. Compared with the other methods, the accuracy of MS location was considerably improved with the proposed algorithm. We can see that TSA has the worst performance, and the proposed algorithm with divided type has the best performance, followed by the proposed algorithm with composite type.

Figure 10. CDFs of the location error with distance-dependent NLOS error.



The third NLOS propagation model is based on a biased uniform random variable [22], in which the measured error of TOA between MS and BS*i* is assumed to be $\gamma_i = \rho_i + w{\cdot}\mu_i$, for $i = 1, 2$ where ρ_i and μ_i are constants. Similarly, the measured error of AOA, is modeled as $|a_i| = b_i + w\, c_i$, for $i = 1, 2$ where b_i and c_i are constants. The error variables for the two BSs are chosen as follows: $\rho_1 = 50$ m, $\rho_2 = 150$ m, $\mu_1 = \mu_2 = 200$ m, $b_1 = 2.5°$, $b_2 = 3°$ and $c_1 = c_2 = 5°$. The resulting CDF curves of the location error are as shown in Figure 11. The divided type with six input data subsets provides better location estimation than the composite type. TSA and HLOP lead to less accurate results under this condition, and the proposed algorithm can still give better MS location estimate.

Figure 11. Comparison of location error CDFs with biased uniform random error.

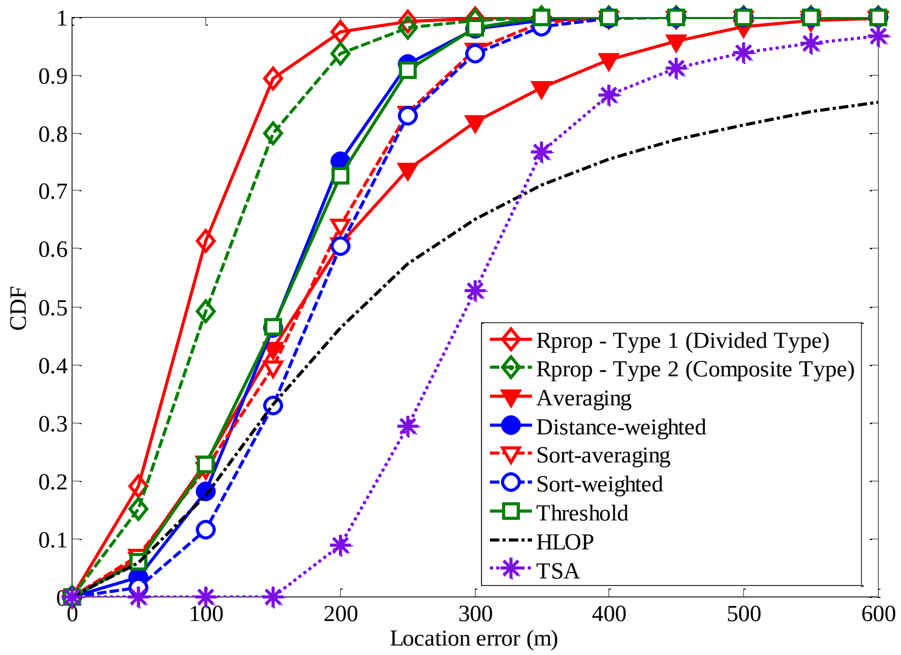


Figure 12 shows CDFs of the MS location error for all methods based on a biased uniform random variable model. The proposed algorithm with 2,000 epochs is slightly better than that with 200 epochs. The simulation results show that the positioning precision of the proposed algorithm with only 200 epochs still yield superior performance when compared with TSA, HLOP and the other geometrical positioning methods. Our proposed algorithm with 200 epochs still enhances the performance of MS location estimation effectively.

Figure 12. CDFs of the location error of the other different methods and the proposed algorithm with 2,000 and 200 epochs.

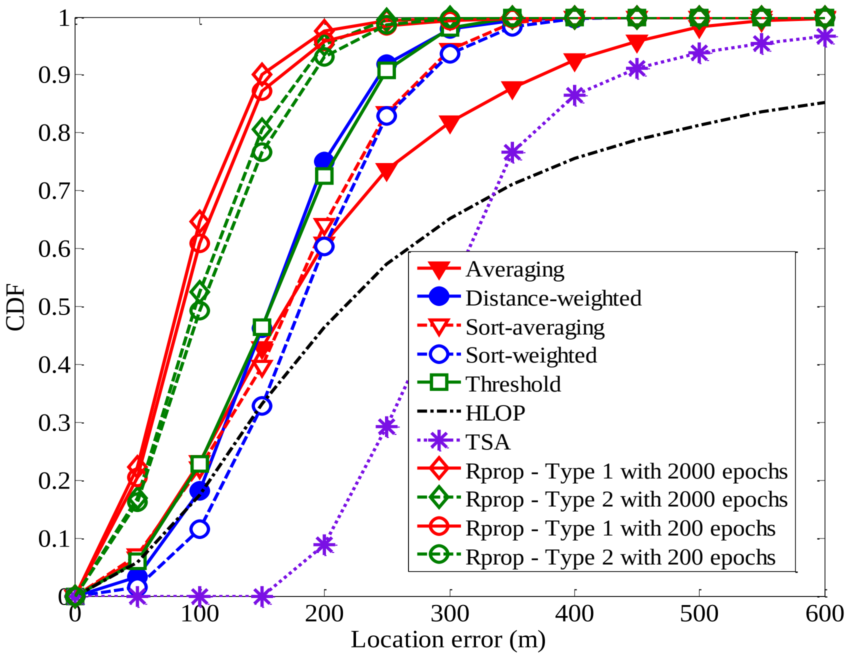


Figure 13. Distribution of the divergence points for TSA.

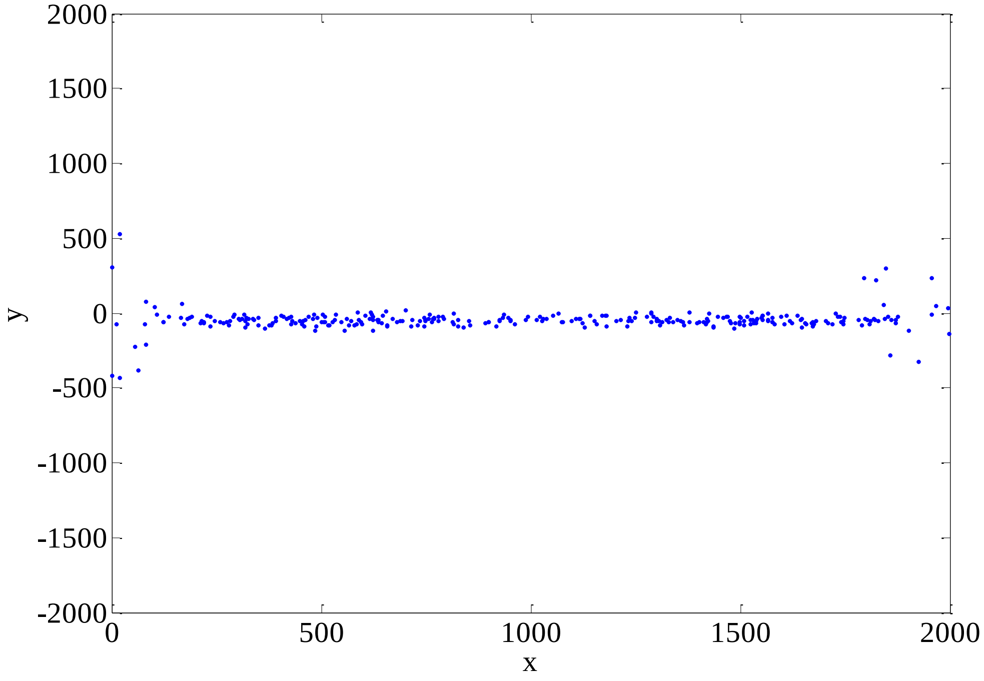
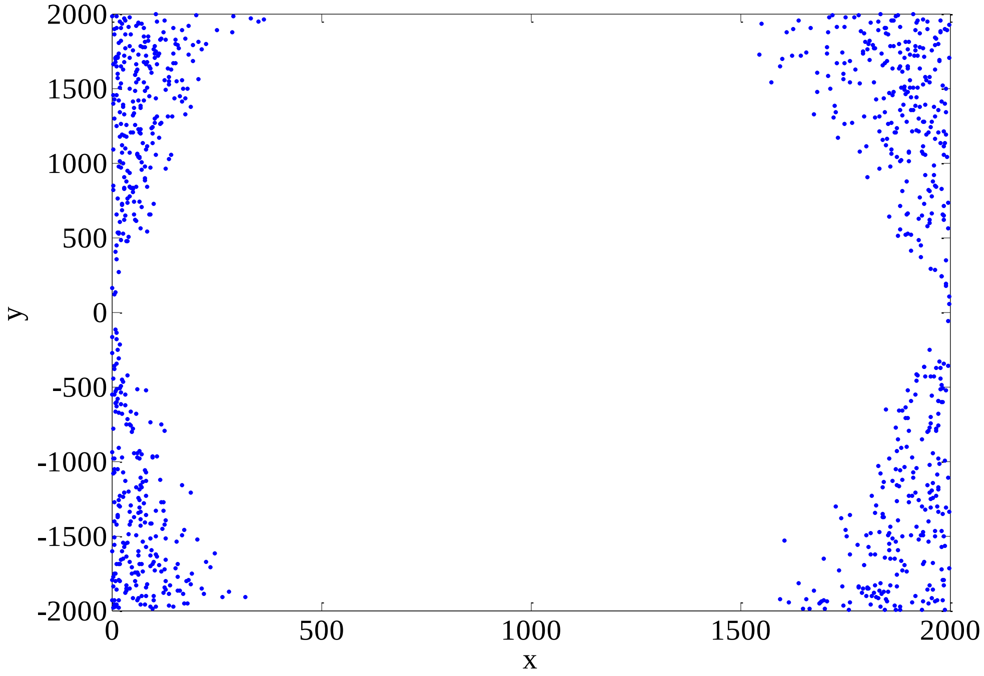


Figure 14. Distribution of the divergence points for HLOP.



When the MS is close to the condition of being aligned with the two BSs, TSA may not converge. HLOP can result in large location errors when the measured angle approaches 90 ° or 270 °. We define the divergence point when the RMS error is above 3,000 m. The distributions of the divergence points of TSA and HLOP are shown in Figures 13 and 14, respectively. The divergence probabilities for different NLOS errors are between 0.32% and 4.66% [17]. The divergence points of TSA and HLOP are not used to calculate the RMS errors and CDF's in our simulations. Note that the proposed algorithms do not have such divergence problem for this situation. TSA and HLOP won't be any divergent problems in the case of more than two BSs available to use. No matter what NLOS propagation model is considered, the simulation results show that the proposed algorithms for Rprop can give very accurate results in the MS location estimation after the training period.

7. Conclusions

This paper proposes novel Rprop-based algorithm to obtain approximate MS location. We combine both TOA and AOA measurements to estimate the MS location under the condition that the MS is heard by only two BSs. The key issue is to apply Rprop to model the relationship of the remaining feasible intersections and MS location. After training, the proposed algorithm can reduce the effects of NLOS errors and improve MS location performance. One the other hand, the traditional methods of TSA and HLOP may not converge when the MS/BSs have an undesirable geometric layout. The positioning accuracy of the proposed algorithms is hardly affected by the relative position between the MS and BSs. Simulation results show that the convergence performance of the proposed algorithms are quite well and provides the capabilities to explicitly reduce the effects of NLOS errors. In summary, the proposed algorithm can always yield better performance than TSA, HLOP and the geometrical positioning methods for different levels of NLOS errors.

References

1. Song, H.L. Automatic vehicle location in cellular systems. *IEEE Trans. Veh. Technol.* **2007**, *43*, 902-908.
2. Rappaport, T.S.; Reed, J.H.; Woerner, B.D. Position location using wireless communications on highways of the future. *IEEE Commun. Mag.* **1996**, *34*, 33-42.
3. Caffery, J.J.; Stuber, G. Overview of radiolocation in CDMA cellular systems. *IEEE Commun. Mag.* **1998**, *36*, 38-45.
4. Borras, J.; Hatrack, P.; Mandayam, N.B. Decision theoretic framework for NLOS identification. In *Proceedings of 48th IEEE Vehicular Technology Conference*, Ottawa, Canada, 18–21 May 1998; pp. 1583-1587.
5. Chen, P.C. A nonline-of-sight error mitigation algorithm in location estimation. In *Proceedings of IEEE Wireless Communications and Networking Conference*, New Orleans, LA, USA, 21–24 September 1999; pp. 316-320.
6. Xiong, L. A selective model to suppress NLOS signals in angle-of-arrival (AOA) location estimation. In *Proceedings of The Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Boston, MA, USA, 8–11 September 1998; pp. 461-465.
7. Cong, L.; Zhuang, W. Non-line-of-sight error mitigation in TDOA mobile location. *Proc. IEEE Globecom* **2011**, *1*, 680-684.
8. Cong, L.; Zhuang, W. Non-line-of-sight error mitigation in mobile location. *IEEE Trans. Wirel. Commun.* **2005**, *4*, 560-573.
9. Yap, J.H.; Ghaheri-Niri, S.; Tafazolli, R. Accuracy and hearability of mobile positioning in GSM and CDMA networks. In *Proceedings of Third International Conference on 3G Mobile Communication Technologies*, London, UK, 8–10 May 2002; pp. 350-354.
10. Reed, J.H.; Krizman, K.J.; Woerner, B.D.; Rappaport, T.S. An overview of the challenges and progress in meeting the E-911 requirement for location service. *IEEE Commun. Mag.* **1998**, *36*, 30-37.
11. Reed, J.H.; Krizman, K.J.; Woerner, B.D.; Rappaport, T.S. An overview of the challenges and progress in meeting the E-911 requirement for location service. *IEEE Commun. Mag.* **1998**, *36*, 30-37.
12. Ma, C. Integration of GPS and cellular networks to improve wireless location performance. In *Proceedings of ION GPS/GNSS 2003*, Portland, OR, USA, 9–12 September 2003; pp. 1585-1596.
13. Caffery, J.; Stuber, G. Subscriber location in CDMA cellular networks. *IEEE Trans. Veh. Technol.* **1998**, *47*, 406-415.
14. Cong, L.; Zhuang, W. Hybrid TDOA/AOA mobile user location for wideband CDMA cellular systems. *IEEE Trans. Wirel. Commun.* **2002**, *1*, 439-447.
15. Venkatraman, S.; Caffery, J.; You, H.R. A novel TOA location algorithm using LOS range estimation for NLOS environments. *IEEE Trans. Veh. Technol.* **2004**, *53*, 1515-1524.
16. Spirito, M.A. Mobile station location with heterogeneous data. In *Proceedings of 52nd Vehicular Technology Conference*, *IEEE VTS-Fall VTC 2000*, Boston, MA, USA, 24–28 September 2000; pp. 1583-1589.

17. Chen, C.S.; Su, S.L.; Huang, Y.F. Hybrid TOA/AOA geometrical positioning schemes for mobile location. *IEICE Trans. Commun*. **2009**, *E92-B*, 396-402.
18. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1996**, *323*, 533-536.
19. Riedmiller, M.; Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of IEEE International Conference on Neural Networks*, San Francisco, CA , USA, 28 March–1 April 1993; pp. 586-591.
20. Foy, W. Position-location solutions by Taylor series estimation. *IEEE Trans. Aerosp. Electron. Syst.* **1976**, *AES-12*, 187-193.
21. Torrieri, D. Statistical theory of passive location systems. *IEEE Trans. Aerosp. Electron. Syst.* **1984**, *AES-20*, 183-197.
22. Venkatraman, S.; Caffery, J. Hybrid TOA/AOA techniques for mobile location in non-line-of-sight environments. In *Proceedings of IEEE Wireless Communications and Networking Conference*, Atlanta, GA, USA, 21–25 March 2004; pp. 274-278.
23. Caffery, J. A new approach to the geometry of TOA location. In *Proceedings of 52nd Vehicular Technology Conference, IEEE VTS-Fall VTC 2000*, Boston, MA, USA, 24–28 September 2000; pp. 1943-1949.
24. Jacobs, R.A. Increased rates of convergence through learning rate adaptation. *Neural Netw.* **1988**, *1*, 295-307.
25. Patnaik, L.M.; Rajan, K. Target detection through image processing and resilient propagation algorithms. *Neurocomputing* **2000**, *35*, 123-135.
26. Chen, C.L.; Feng, K.T. An efficient geometry-constrained location estimation algorithm for NLOS environments. In *Proceedings of International Conference on Wireless Networks, Communications and Mobile Computing*, Maui, HI, USA, 13–16 June 2005; pp. 244-249.
27. Kohzadia, N.; Boyd, M.S. A comparison of artificial neural network and time series models for forecasting commodity prices. *Neurocomputing* **1996**, *10*, 169-181.
28. Venkatachalan, A.R.; Sohl, J.E. An intelligent model selection and forecasting system. *J. Forecast.* **1999**, *18*, 167-180.

© 2011 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).