Advances in Mechanical Engineering

Advances in Mechanical Engineering
2018, Vol. 10(12) 1–12
© The Author(s) 2018
DOI: 10.1177/1687814018808682
journals.sagepub.com/home/ade
SAGE

Indoor localization algorithm based on artificial neural network and radio-frequency identification reference tags

Quangang Wen[1,2], Yanchun Liang[1,2], Chunguo Wu[1,2], Adriano Tavares[3] and Xiaosong Han[1]

Abstract
With the development of Internet of Things technology, radio-frequency identification localization methods have been widely applied due to their low cost and ease of deployment. The indoor radio-frequency identification localization algorithm based on received signal strength indication technology is a currently hot topic. Because the received signal strength is highly dependent on environments, the classic algorithms may result in large errors in localization accuracy. This article proposed a new radio-frequency identification localization algorithm, named BP_LANDMARC, by utilizing the back propagation neural network, which is designed to address nonlinear changes in radio-frequency signals. A strategy for selecting different working parameters in variable environments is presented. The evaluation methods of root mean square error and cumulative distribution function are used to compare the proposed algorithm with some existing algorithms. Experimental results show that the proposed algorithm remarkably improves the localization accuracy of both absolute distance and cumulative probability. Moreover, the proposed algorithm performs effectively and efficiently when it is applied to a logistics warehouse management system.

Keywords
Artificial neural network, radio-frequency identification, indoor localization, received signal strength indication, LANDMARC

Date received: 17 June 2018; accepted: 24 September 2018

Handling Editor: Fei Yu

Introduction

With the development of Internet of Things (IOT) technology, indoor wireless localization technology is facing more and more requirements. Zhang et al.[1] and LM Ni et al.[2] discussed common localization methods including Global Positioning System (GPS), infra-red (IR), Bluetooth, Wireless Fidelity (Wifi), radio-frequency identification (RFID), and so on. However, most of these localization methods are not applicable to indoor environments. For example, GPS signal cannot pass through buildings and does not meet the standard for civilian localization with error less than 10 m, which

[1]Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun, China
[2]Zhuhai Laboratory of Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, Zhuhai College of Jilin University, Zhuhai, China
[3]Department of Industrial Electronics, University of Minho, Guimarães, Portugal

Corresponding author:
Xiaosong Han, Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun 130012, China.
Email: hanxiaosong@jlu.edu.cn

makes the GPS-based localization methods fail in indoor environments. IR localization technology could satisfy the accuracy requirement, but its signal cannot pass through obstacles. It is only used in line-of-sight scenarios, which makes IR also unsuitable for indoor localization. With the advantages of non-contact, non-line-of-sight, low-cost, and strong anti-interference performance, R Carotenuto et al.[3] introduced Bluetooth, Wifi, and RFID technologies in indoor localization. For instance, JKY Ng et al.[4] described a system of wireless location for tracking indoor mobile users. M Scherhäufl et al.[5] introduced an automatic blind calibration method which used passive UHF RFID tags to realize an indoor localization system and a root-mean-square error method is used to evaluate the positioning accuracy. Using RFID, J Maneesilp et al.[6] proposed a three-dimensional (3D) localization algorithm according to the relationship between readers and reference tags. Furthermore, HY Yu et al.[7] proposed a real-time RFID localization system used to localize a moving object. In warehouse management, the main operations include stock-in, stock-out, storage moving, and inventory verification. Compared with the Bar-code technology, RFID technology supports batching operations, which enhances the efficiency.[8,9] Hence, RFID localization technology has been used in smart shelves to track goods efficiently.[10]

RFID technology is a non-contact automatic identification technology, which uses the reflection transmission radio frequency signal to recognize objects. An RFID system has at least two basic parts, including RFID readers and tags. Electronic tags can be classified as active tags and passive tags. In current researches, localization technology based on distance calculation has three categories, including received signal strength indication (RSSI), direction of arrival (DOA), and time of arrival (TOA).[11,12] The working process of RSSI technology is to collect the signal strength of the radio frequency in an actual environment. Then, the target position is evaluated by the signal propagation attenuation model. This method is a priority solution in indoor RFID localization for its advantages of low cost, low investment, and simplicity of device. J Hightower et al.[13] proposed a method named "Spot On," in which different signal strengths from an electric tag to a number of readers are detected, and the position of the target object is calculated using a clustering-weighted algorithm. LM Ni et al.[14] introduced location identification based on dynamic active RFID calibration (LANDMARC), which defines the reference tag and tracking tag. The coordinates of the reference tags are known. By comparing different values between tracking tags and reference tags from multiple readers, the coordination values of tracking tags could be estimated with an empirical weighted formula. Ma et al.[15] improved the location accuracy using reference tags in an application of LANDMARC technology on vehicle locations. However, the location accuracy was sensitive to the distribution of tags and readers, the number of nearest neighbors was determined empirically and the weights were affected by the density of the tags.

Based on the LANDMARC algorithm, Zhao et al.[16] and NB Aldin et al.[17] proposed virtual reference elimination (VIRE) algorithm, which uses many virtual reference tags to improve the accuracy. VIRE estimates the signal strength of virtual reference tags using a linear interpolation method and then figures out the positions of the tracking tags. This algorithm has advantages of noise resistance, low cost, and high accuracy. However, due to the reflection of radio frequency signals in indoor environments, many factors, such as linear interpolation errors and low localization accuracy of the edge tags, deteriorate the accuracy of the VIRE algorithm.

Recently, to improve the robustness of the localization system, many researchers are trying to apply artificial neural network (ANN) technology to RFID localization. For instance, WS Holland et al.[18] proposed a real-time RFID localization method based on ANN and reference tags. By combining the support vector machine (SVM) and probability neural networks, G Deak et al.[19] presented a kind of device-free passive localization (DFPL) scenario. Yang et al.[20] introduced an online sequential extreme learning machine method to improve the performance and address online localization function. All of the above works have great value in using ANN to address RFID localization problems. Based on the above analysis, the classic localization model cannot perform well in a nonlinear change of radio frequency signal. To conquer this problem, an indoor localization algorithm BP_LANDMARC is proposed in this article, which can address this problem by utilizing the back propagation (BP) ANN. The proposed BP_LANDMARC takes the RSSI value of reference tags as input data and the actual coordinate of reference tags as output data to train a BP ANN. In addition, its calculation accuracy is improved by selecting different working parameters in different environment factors.

The following sections are arranged as follows. In section "Related work," related technologies of RFID indoor localization are introduced, including the LANDMARC algorithm and the VIRE algorithm, and their merits and faults are analyzed. The BP_ LANDMARC algorithm is proposed in section "Localization algorithm based on ANN and reference tag," and its working process is introduced. In section "Testing and application," simulation and testing are discussed, and the performance of the BP_LAND MARC algorithm is compared with the LANDMARC algorithm and the VIRE algorithm. Then, the BP_ LANDMARC algorithm is applied to a practical logistics warehouse. Finally, section "Conclusion" concludes our work and discusses future directions.

Related work

The LANDMARC algorithm is a classic localization method of RSSI technology.[14] The LANDMARC algorithm assumes that there are n readers, m reference tags, and u tracking tags in the localization system. The readers could detect the signal strength and identify it as one of eight levels. The scanning frequency of readers is 30 times per second. Working under these conditions, the localization process of LANDMARC is as follows.

Suppose that n readers received the signal strength of a tracking tag, denoted as $S = (s_1, s_2, \ldots, s_n)$, and received the signal strength of a reference tag, denoted as $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, where s_i and α_i denote the tracking tag and the reference tag's signal strength perceived on reader $i\,(1 \leq i \leq n)$, respectively. For each individual tracking tag $p\,(1 \leq p \leq u)$, we define a Euclidean distance formula as equation (1)

$$E_j = \sqrt{\sum\nolimits_{i=1}^{n} \left(\alpha_{ji} - S_i\right)^2},\ j \in (1, \ldots, m) \qquad (1)$$

In equation (1), the Euclidean distance of the signal strength between the tracking tags and the reference tags is denoted as vector $E = (E_1, E_2, \ldots, E_m)$. In the vector, a smaller E_j value means a smaller signal strength difference between a tracking tag and a reference tag. We select k smallest values in vector E to express the k-nearest neighbor tags, which is called k-nearest neighbor algorithm. Then, the tracking tags' position coordinate is estimated. Empirically, a larger k value does not guarantee good performance, and $k = 4$ or $k = 5$ is a suitable choice. The coordinate estimated expression of the tracking tag is defined as equation (2)

$$(x, y) = \sum_{i=1}^{k} w_i (x_i, y_i) \qquad (2)$$

where (x, y), w_i, and (x_i, y_i) denote the coordinate of tracking tag, the weighting factor, and the coordinate of reference tag i, respectively. Intuitively, the selection of w_i should depend on the value of the k-nearest neighbors. The smaller the value, the larger the weight. The expression of w_i is defined as equation (3)

$$w_i = \frac{1}{E_i^2 \sum_{j=1}^{k} (1/E_j^2)},\ i \in (1, \ldots, k) \qquad (3)$$

In equation (3), E_j denotes the jth component of the vector in the k-nearest neighbor RSSI value because the k value is estimated by the experience value, and w_i is influenced by the density of the tags and other factors. The LANDMARC algorithm has low localization accuracy sometimes. Improving the density of the reference tags in a localization area could increase the localization accuracy, but it will lead to interference if there are too many tags. In contrast, localization accuracy will decrease. To resolve these issues, the VIRE algorithm is proposed.

The VIRE algorithm is an improved method based on the LANDMARC algorithm, where the layout assumption of readers and tags is the same as that in the LANDMARC algorithm.[16] The basic idea is to improve the localization accuracy by excluding small probability position rather than adding additional hardware, such as readers or tags. Virtual reference tags based on the reference tags are utilized in this algorithm. Virtual reference tags are arranged according to some distributed rules. Generally, in an $n \times n$ actual reference tag grid, $n-1$ virtual reference tags are interpolated between two adjacent actual tags. For example, in a 4×4 actual reference tag grid, every two adjacent actual tags are inserted by three virtual reference tags, so it is summarized as a 13×13 reference tag grid. The RSSI values of the virtual reference tags are calculated by the linear interpolation method, and the RSSI values of the horizontal direction virtual reference tags are calculated as equation (4)

$$\begin{aligned} S_k\left(T_{p,b}\right) &= S_k(T_{a,b}) + p \times \frac{S_k(T_{a+n,b}) - S_k(T_{a,b})}{n} \\ &= \frac{p \times S_k(T_{a+n,b}) - (n+1-p) S_k(T_{a,b})}{n+1} \end{aligned} \qquad (4)$$

The RSSI values of vertical direction and virtual reference tags are calculated as equation (5)

$$\begin{aligned} S_k\left(T_{a,q}\right) &= S_k(T_{a,b}) + q \times \frac{S_k(T_{a,b+n}) - S_k(T_{a,b})}{n+1} \\ &= \frac{q \times S_k(T_{a,b+n}) - (n+1-q) S_k(T_{a,b})}{n+1} \end{aligned} \qquad (5)$$

In a two-dimensional plane, the RSSI values of the virtual reference tags are calculated as equation (6)

$$\begin{aligned} & S_k\left(T_{i,j}\right) \\ & = \frac{S_k\left(T_{p,b}\right) + S_k\left(T_{p,b+n}\right) + S_k\left(T_{a,q}\right) + S_k\left(T_{a+n,q}\right)}{2} \end{aligned} \qquad (6)$$

In equations (4)–(6), $S_k(T_{i,j})$ denotes the virtual reference tags' RSSI value of reader k and its coordinate value is (i,j); corresponding parameters are $a = i/n$, $b = j/n$, $0 \leq (p = i \bmod n) \leq n-1$, and $0 \leq (q = j \bmod n) \leq n-1$.

After setting all the virtual reference tags, the RSSI values combined with the actual reference tags constitute a reference tag grid. The concept of a proximity map is used to describe their relationship. Every reader creates and maintains its proximity map according to its perceived RSSI values. The tracking tags' RSSI values are compared with a given threshold; if the results are less the threshold, the related areas in the proximity map are marked as "1." This means that the tracking tag is in large proximity in these areas. Summarizing all k readers' proximity maps, a final proximity map indicates the most possible tracking tag's localization. Therefore, it is called exclusion method of small probability events.

To improve the accuracy, the calculation method of weight is optimized in the VIRE algorithm using two weights w_{1i} and w_{2i}, which is similar to the LANDMARC algorithm, denotes the different RSSI value between the selected virtual reference tags and tracking tag. The smaller the value, the nearer the area, and the larger the weight. The expression of w_{1i} is defined as equation (7)

$$w_{1i} = 1 - \sum_{k=1}^{K} \frac{S_k(T_i) - S_k(R)}{4S_k(T_i)} \tag{7}$$

where $S_k(T_i)$ denotes the ith tracking tag's RSSI value and $S_k(R)$ denotes the reference tag's RSSI value. Another weight w_{2i} is used to express the density of virtual reference tags in some area. If a certain area has a larger w_{2i} value, it means the tracking tag is more possible in this area. The expression of w_{2i} is defined by equation (8)

$$w_{2i} = \frac{p_i}{\sum_i^{n_a} p_i} = \frac{n_{ci}}{\sum_i^{n_a} n_{ci}} \tag{8}$$

$$p_i = \frac{n_{ci}}{n_a} \tag{9}$$

In equations (8) and (9), p_i denotes the ratio of conjunctive possible regions to the entire sensing area. n_{ci} is the number of conjunctive areas in the final selected *proximity map*, and n_a denotes the number of total areas of the entire sensing area. Considering the two weights, the final weight is defined as $w_i = w_{1i} \times w_{2i}$. The coordinate of a tracking tag is given by equation (10)

$$(x, y) = \sum_{i=1}^{n_a} w_i(x_i, y_i) \tag{10}$$

Compared with the LANDMARC algorithm, the localization accuracy is highly improved in the VIRE algorithm. Furthermore, the linear interpolation method is used to calculate the virtual reference tag's RSSI value in this algorithm, which will lead to some error.[21] Although Shao[22] and Wu and Huang[23] proposed some methods to improve the VIRE algorithm, such as the two-dimensional Newton interpolation method, Lagrange interpolation method, or Bayesian-based method in references, these methods perform quite differently with the actual signal strength. Kalman-Filter technology is also used to improve the VIRE algorithm.[24–26] However, when the actual measurement model is strongly nonlinear, its performance becomes poor, even if not converged. YS Chiou et al.[27] demonstrated a large localization error on the boundary area of the VIRE algorithm.

When ANN and RFID technology are used to solve the indoor location problems, some key issues should be considered, including the nonlinear mapping ability and adaptive ability of the network, the complexity of network structure and the calculation performance of the algorithm, and the real-time performance affected by different environments. For instance, radial basis function (RBF) algorithm is utilized to implement the indoor location.[28] Compared with the BP algorithm, the RBF network is more complex in the case of the same precision, and the hidden layers are greatly increased while the training samples are increased, which will increase the calculation cost. k-nearest neighbors (KNN) algorithm is utilized in an indoor location method;[29] however, the selection of k value is difficult and the calculation cost is also large in this method. Kung et al.[30] used the BP neural network and RFID reference label to realize an indoor location method. Compared with above methods, its structure becomes simple, and the efficiency of calculation meets the requirement of real-time performance. Combining the application of this article, we also need to consider the impact of environmental changes on the BP network.

Based on the above discussion, the contributions of this article are as follows. The BP ANN is utilized to conquer the nonlinear change in radio frequency signal. As the RSSI localization technology is easily affected by the environment, it is necessary to consider the environment factors to improve the localization precision. The environment factors such as temperature, humidity, luminosity, and electromagnetic radiation intensity are taken as working parameters. Then, ANN is trained in different working environments and different sets of weights are produced. They are stored in the environment factor table as different application scenarios. The environment factors are detected before the localization system works. According to these factors, a set of weights are selected in the environment factor table. Then, a specific scene of ANN is created and ready for working. If there is no suitable application scenario selected in the environmental factor table, a new

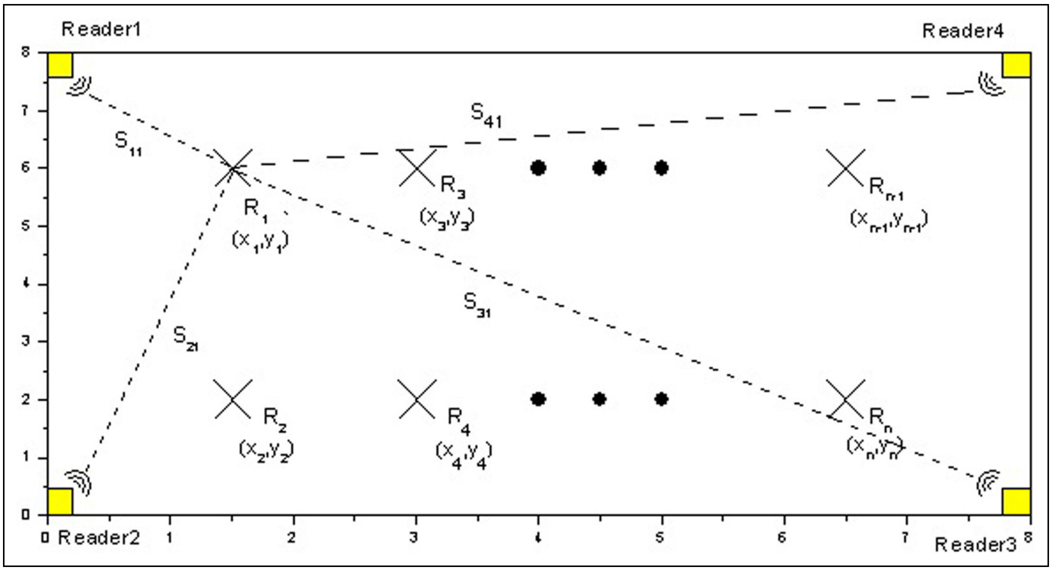

Figure 1. Structure of a localization system.

application scenario should be set up in the system. It is necessary to retrain the ANN for this scenario. Different working parameters are selected within different environment factors to improve the calculation accuracy.

Localization algorithm based on ANN and reference tag

BP ANN

By combining the BP ANN and the LANDMARC localization algorithm, an algorithm named BP_LANDMARC is proposed in this article. BP_LANDMARC sets up a nonlinear localization model by adjusting the network's parameters based on the training data. The learning process of BP-ANN includes two stages. The first stage is the forward propagation of input signals to regress certain variables, and the second stage is the back propagation of regressing error to adjust the model parameters. A three-layer BP-ANN is constructed in this article. While training, the known reference tag's RSSI value consists of the inputs to the input layer. The middle layer is designed as a single hidden layer. The output layer is associated with the reference tag's coordinate value. If the actual output does not match the expected output, then the back-propagation stage is invoked. The error will propagate back to each layer from the output layer and the hidden layer to the input layer. The weights of each layer are adjusted according to the error gradient descent method. This process works continuously until the output error is reduced to the expected value. When it is on the actual working status, the real-time tracking tag's RSSI value is fed to the BP-ANN. Then, the output is the tracking tag's estimated coordinate value.

Similar to the LANDMARC localization system, the hardware structure of the BP_LANDMARC localization system includes readers, reference tags, and tracking tags.

The structure of a localization system is shown in Figure 1, assuming there are four readers and n reference tags in this system. Taking the reference tag R_1 as an example, its RSSI values from the four readers are s_{11}, s_{21}, s_{31}, s_{41}, and the corresponding coordinate is (x_1, y_1). Analogously, in a localization system that has m readers and n reference tags, the RSSI values are recorded as follows

$$S = [s_1, s_2, \ldots, s_n] = \begin{bmatrix} s_{11}s_{12}\cdots s_{1n} \\ s_{21}s_{22}\cdots s_{2n} \\ \cdots\cdots\cdots\cdots \\ s_{m1}s_{m2}\cdots s_{mn} \end{bmatrix} \tag{11}$$

where $s_i = (s_{1i}, \ldots, s_{mi})^T$ $(1 \leqslant i \leqslant n)$ is the RSSI vector of the ith reference tags reading from m readers.

Corresponding to the RSSI matrix S, its reference coordinates are denoted as follows

$$D = [d_1, d_2, \ldots, d_n] = \begin{bmatrix} x_1x_2\cdots x_n \\ y_1y_2\cdots y_n \end{bmatrix} \tag{12}$$

where $d_i = (x_i, y_i)^T$ $(1 \leqslant i \leqslant n)$ is the coordinates of the ith reference tags.

Assuming there are k tracking tags in the localization system, the RSSI matrix of tracking tags are denoted as S', given by equation (13)

$$S' = \left[s_1', s_2', \ldots, s_k'\right] = \begin{bmatrix} s_{11}' s_{12}' \cdots s_{1k}' \\ s_{21} s_{22} \cdots s_{2k} \\ \cdots\cdots\cdots\cdots \\ s_{m1}' s_{m2}' \cdots s_{mk}' \end{bmatrix} \tag{13}$$

where $s_j' = (s_{1j}', \ldots, s_{mj}')^T$ $(1 \leq j \leq k)$ is the RSSI vector of the kth tracking tags reading from m readers.

Its reference coordinate is denoted as D', given by equation (14)

$$D' = \left[d_1', d_2', \ldots, d_k'\right] = \begin{bmatrix} x_1' x_2' \cdots x_k' \\ y_1 y_2 \cdots y_k \end{bmatrix} \tag{14}$$

where $d_j' = (x_j', y_j')^T$ $(1 \leq j \leq k)$ is the coordinate of the kth tracking tags.

The working principle of the BP-ANN is to take the vector $s_1, s_2, \ldots, s_n$ in equation (11) as its input port, while its output data are compared with supervised signals, that is, those in equation (12). After continuous training, a mapping between the input RSSI values and the output reference tag's coordinate can be set up. While it is used to localize new tracking tags, RSSI values $s_1', s_2', \ldots, s_k'$ in equation (13) are fed into the trained BP-ANN, respectively. Then, its output binary tuples are taken as estimated positions of the tracking tags.

Influenced by the environment factors, the energy attenuation of the radio frequency signal is a nonlinear process. Its mechanism is very complex and the algorithms Landmarc and Vire mentioned above do not match the nonlinear change process in essential. Theoretically, any nonlinear continuous function can be approximated with any precision by a three-layer BP neural network. Moreover, the BP neural network has strong self-learning and self-adaptive ability, which is very suitable to express the different changes of RSSI caused by environmental changes. Compared with other ANN technologies, the BP neural network is not only more mature and easy to implement but also used for solving many problems in engineering field. Based on these factors, the BP neural network is utilized in this article.

Model of BP_LANDMARC algorithm

The structure of the BP-ANN used by this article is shown in Figure 2, where there are m nodes in the input layer, corresponding to the m reader's RSSI value of one tag. In this article, we set $m = 4$. There are q nodes in the hidden layer, where we set $q = 10$. There are p nodes in the output layer. We set $p = 2$, where the outputs o_1 and o_2 correspond to the tag's coordinate value (x, y).

The input vector is $S = (s_1, s_2, \ldots, s_m)^T$. The vector of the hidden layer is $H = (h_1, h, \ldots, h_q)^T$. The output vector is $O = (o_1, o_2)^T = (x, y)^T$. The supervised signal of the output layer is $O_r^T = (o_{r1}, o_{r2})^T = (x_r, y_r)^T$

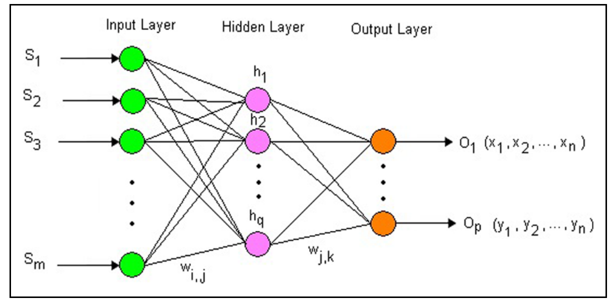

Figure 2. Structure of the BP artificial neural network.

$(r = 1, 2, \ldots, n)$. The connection weight between the input-layer node i and the hidden-layer node j is w_{ij}. The connection weight between the hidden-layer node j and the output-layer node k is w_{jk}. The threshold of the hidden layer is $\theta_j (j = 1, 2, \ldots, q)$. The threshold of the output layer is $\theta_k (k = 1, 2)$. In the output layer, expressions of the output results are shown as equations (15) and (16)

$$O_k = f(net_k), \ (k = 1, 2) \tag{15}$$

$$net_k = \sum_{j=1}^{q} w_{jk} h_j - \theta_k \ (k = 1, 2) \tag{16}$$

For the hidden layer, there are the following equations

$$h_j = f(net_j), j = (1, 2, \ldots, q) \tag{17}$$

$$net_j = \sum_{i=1}^{m} w_{ij} s_i - \theta_j, j = (1, 2, \ldots, q) \tag{18}$$

where the sigmoid function $f(x)$ is as follows

$$f(x) = \frac{1}{1 + e^{-x}} \tag{19}$$

The output error function is defined as follows

$$Er_k = \frac{1}{2} \sum\nolimits_1^2 \left\| o_k' - o_k \right\|^2 \tag{20}$$

Testing and application

In this section, the functional simulation of the proposed algorithm is first carried out in the MATLAB environment. Second, the error analysis is discussed. Third, the optimal selection of reference tag number is analyzed. Finally, an application of the proposed algorithm is presented.

Function testing

Before applied on the real-time localization, the BP_LANDMARC system should learn from the collected RSSI data of the reference tags. To improve the

Table 1. Training sample data of ANN.

No.	RSSI_1	RSSI_2	RSSI_3	RSSI_4	X	Y
1	13.01029996	7.520484478	7.520484478	10.51152522	0.5	0.5
2	4.423591485	7.753272169	5.829467168	9.744093372	1.83	0.5
3	0.118992233	8.213679386	3.731164249	9.009736853	3.17	0.5
4	3.117538611	8.836614352	0.96910013	8.356905715	4.5	0.5
5	5.350126589	9.555808602	3.058824985	7.849341225	5.83	0.5
6	7.127432089	10.31857116	10.24823584	7.554513603	7.17	0.5
7	4.423591485	5.829467168	7.753272169	9.744093372	0.5	1.83
8	1.724871348	6.168837676	6.168837676	8.81130943	1.83	1.83
9	1.267445375	6.817436024	4.268725713	7.880897729	3.17	1.83
10	3.731164249	7.655029831	1.934338148	7.013760741	4.5	1.83
11	5.727425591	8.57667471	0.939045029	6.306538347	5.83	1.83
12	7.381637254	9.51283539	3.91949645	5.87960273	7.17	1.83
13	0.118992233	3.731164249	8.213679386	9.009736853	0.5	3.17
14	1.267445375	4.268725713	6.817436024	7.880897729	1.83	3.17
15	3.022346968	5.236019669	5.236019669	6.695234907	3.17	3.17
16	4.811239972	6.395971516	3.478718675	5.515855244	4.5	3.17
17	6.440006822	7.588278164	1.679733688	4.48018873	5.83	3.17
18	7.880897729	8.734497792	0.302848039	3.812153913	7.17	3.17
19	3.117538611	0.96910013	8.836614352	8.356905715	0.5	4.5
20	3.731164249	1.934338148	7.655029831	7.013760741	1.83	4.5
21	4.811239972	3.478718675	6.395971516	5.515855244	3.17	4.5
22	6.074550232	5.11883361	5.11883361	3.891660844	4.5	4.5
23	7.346220586	6.653724963	3.969738359	2.290273342	5.83	4.5
24	8.549804123	8.035329816	3.210688451	1.120834159	7.17	4.5
25	5.350126589	3.058824985	9.555808602	7.849341225	0.5	5.83
26	5.727425591	0.939045029	8.57667471	6.306538347	1.83	5.83
27	6.440006822	1.679733688	7.588278164	4.48018873	3.17	5.83
28	7.346220586	3.969738359	6.653724963	2.290273342	4.5	5.83
29	8.328635836	5.87960273	5.87960273	0.273858005	5.83	5.83
30	9.314013624	7.486186611	5.406075122	2.685007708	7.17	5.83
31	7.127432089	10.24823584	10.31857116	7.554513603	0.5	7.17
32	7.381637254	3.91949645	9.51283539	5.87960273	1.83	7.17
33	7.880897729	0.302848039	8.734497792	3.812153913	3.17	7.17
34	8.549804123	3.210688451	8.035329816	1.120834159	4.5	7.17
35	9.314013624	5.406075122	7.486186611	2.685007708	5.83	7.17
36	10.11664406	7.164670858	7.164670858	8.573324964	7.17	7.17

ANN: artificial neural network; RSSI: received signal strength indication.

accuracy, the sample data should be collected in different times and different working environments, which correspond to different application scenarios. An example reference tag learning sample data is shown in Table 1, which presents a snapshot of the training data at one time point. In Table 1, the first column denotes the reference number, and the total number of reference tags is 36. The columns from RSSI_1 to RSSI_4 denote the RSSI values obtained from the four readers. The columns X and Y denote the coordinate values of a reference tag.

In a localization system, some indexes are used to evaluate the algorithms' behavior. In our article, Euclidean distance is used to measure the localization errors, indicating the error between a tracking tag's estimated and actual positions. Assume the tracking tag's actual coordinate value is (x_0, y_0) and its estimated coordinate value is (x, y). Then, the root mean square error (RMSE) is defined as follows

$$e = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{21}$$

To examine the stability of the localization algorithms, we calculated the cumulative distribution function (CDF), $F_e(z) = P(e \leqslant z)$, which means the probability of the error, e, is less than z meters. For example, $F_e(0.5) = 0.9$ means the probability of a localization error less than 0.5 m is 0.9.

An RFID localization system is constructed as shown in Figure 3. In an $8\,\text{m} \times 8\,\text{m}$ two-dimensional rectangle area, there are four RFID readers on each corner, and 6×6 reference tags are uniformly

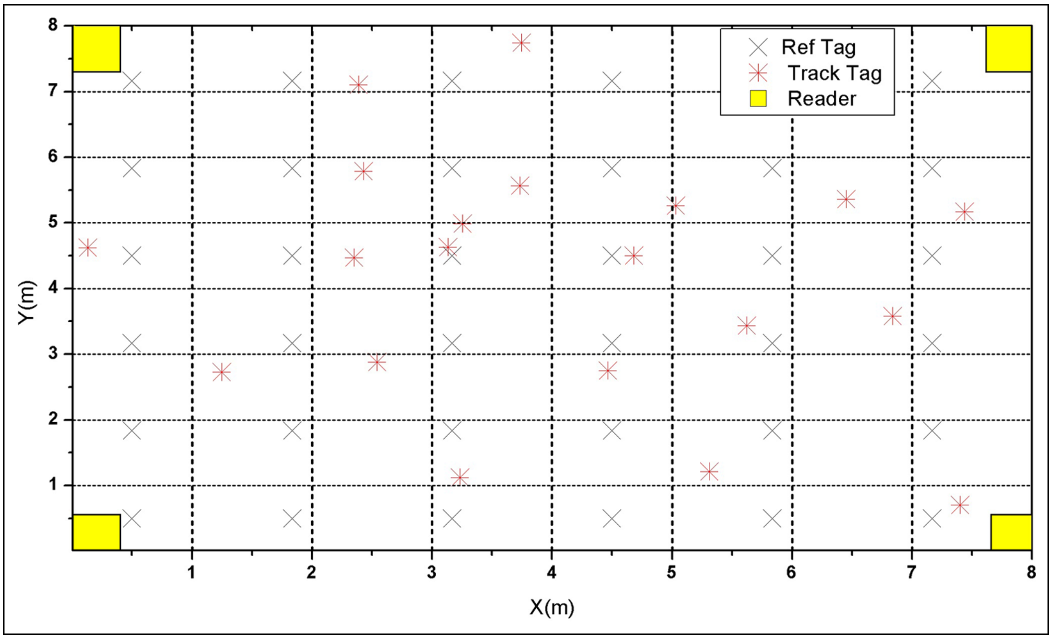

Figure 3. Simulation structure of localization system.

Table 2. Localization results of three algorithms.

No.	Act_Coordinate	Landmarc	Vire	BP_Landmarc	Err_Landmarc	Err_Vire	Err_BP_Landmarc
1	(7.400, 0.700)	(6.980, 1.010)	(6.400, 1.590)	(7.174, 0.679)	0.5239	1.3350	0.2268
2	(2.542, 2.877)	(2.990, 2.980)	(2.676, 3.069)	(2.455, 3.158)	0.4603	0.2344	0.2943
3	(3.733, 5.565)	(4.000, 6.960)	(3.755, 5.341)	(3.604, 5.831)	1.4209	0.2254	0.2953
4	(4.465, 2.748)	(4.000, 1.029)	(4.461, 2.898)	(4.322, 2.979)	1.7802	0.1504	0.2717
5	(5.623, 3.436)	(6.965, 3.985)	(5.449, 3.519)	(5.519, 3.612)	1.4505	0.1928	0.2044
6	(5.312, 1.216)	(4.652, 1.695)	(5.235, 1.727)	(5.150, 1.340)	0.8157	0.5167	0.2045
7	(3.135, 4.627)	(2.981, 5.009)	(3.143, 4.631)	(2.978, 4.832)	0.4124	0.0092	0.2584
8	(6.452, 5.358)	(6.130, 4.652)	(5.927, 5.289)	(6.259, 5.531)	0.7756	0.5298	0.2597
9	(1.249, 2.724)	(1.696, 3.348)	(1.669, 2.871)	(1.080, 2.904)	0.7672	0.4448	0.2470
10	(2.388, 7.102)	(3.348, 6.130)	(2.318, 6.354)	(2.290, 7.243)	1.3657	0.7513	0.1721
11	(6.836, 3.581)	(6.965, 3.985)	(6.217, 3.629)	(6.776, 3.758)	0.4245	0.6211	0.1869
12	(4.681, 4.497)	(5.009, 5.009)	(4.641, 4.423)	(4.587, 4.787)	0.6088	0.0837	0.3049
13	(2.430, 5.785)	(2.981, 5.009)	(2.585, 5.584)	(2.392, 5.931)	0.9512	0.2539	0.1505
14	(2.349, 4.467)	(1.034, 3.985)	(2.486, 4.442)	(2.250, 4.747)	1.4000	0.1391	0.2973
15	(5.032, 5.261)	(5.009, 5.009)	(4.930, 5.163)	(4.993, 5.533)	0.2524	0.1414	0.2743
16	(3.231, 1.121)	(4.000, 1.029)	(3.269, 1.789)	(3.054, 1.366)	0.7744	0.6700	0.3027
17	(7.435, 5.171)	(6.130, 4.652)	(6.534, 5.002)	(7.043, 5.225)	1.4039	0.9266	0.3954
18	(3.745, 7.742)	(4.000, 6.961)	(3.745, 6.611)	(3.748, 7.622)	0.8219	1.1314	0.1196
19	(3.254, 4.987)	(2.981, 5.009)	(3.352, 4.874)	(3.086, 5.197)	0.2741	0.1493	0.2696
20	(0.131, 4.626)	(1.034, 3.985)	(1.334, 4.517)	(0.264, 4.687)	1.1076	1.2080	0.1465

distributed in this area, shown as “x” in the figure. On the other hand, there are also 20 randomly distributed tracking tags, marked as “*.”

Error analysis

To compare the results of different algorithms, we set the same condition to simulate the LANDMARC algorithm, VIRE algorithm, and BP_LANDMARC algorithm. The results of the three algorithms are shown in Table 2, where the first column denotes the number of tracking tags, and the Act_Coordinate column is for the 20 tracking tags’ actual position coordinate. The columns LANDMARC, VIRE, and BP_LANDMARC are the estimated coordinates obtained from the corresponding algorithms. Then, the

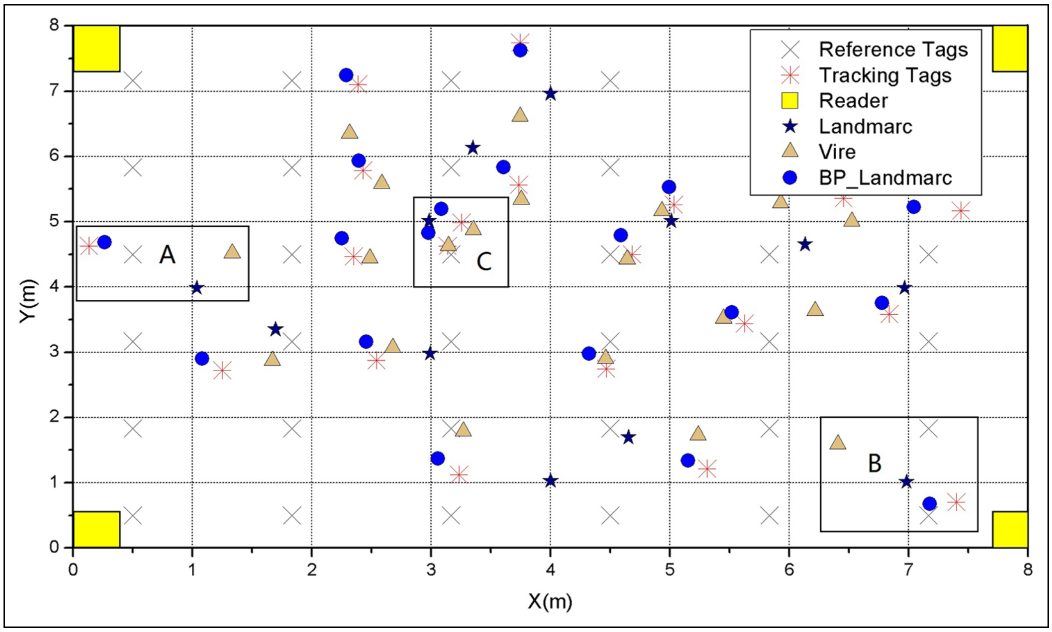

Figure 4. Simulation results of three algorithms.

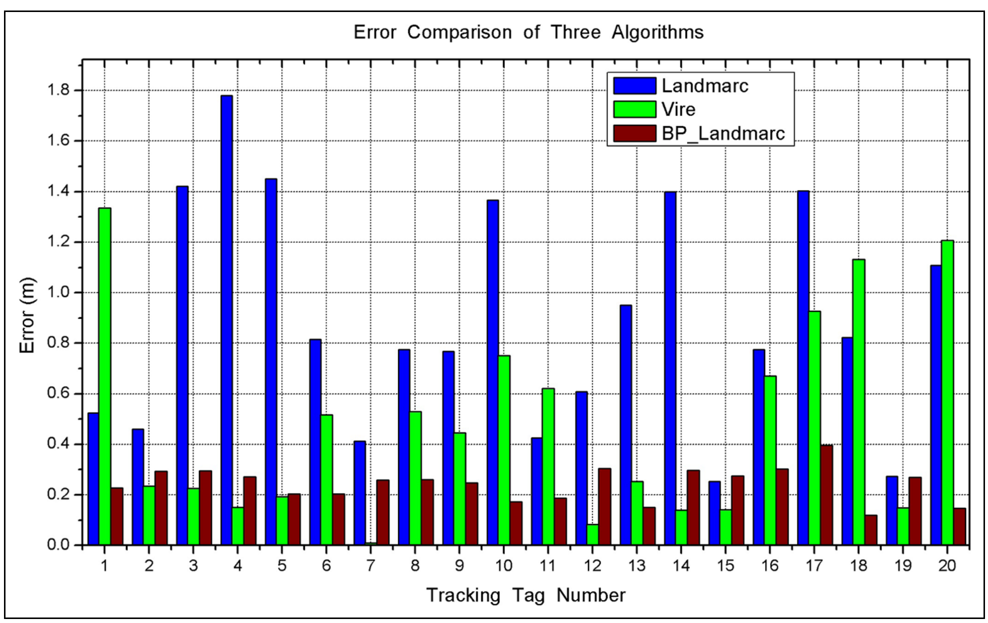

Figure 5. Estimation errors of three algorithms.

last three columns are for the corresponding RMSE errors.

The localization results are shown in Figure 4. For some tracking tags near the boundaries, the effect of BP_LANDMARC is better than those of VIRE and LANDMARC, shown as the tracking point in rectangle A and the tracking point in rectangle B, respectively. The points shown in rectangle C, estimated as one single point by LANDMARC due to their closeness, could be separated properly by both BP_LANDMARC and VIRE, while the accuracy of BP_LANDMARC is better than that of VIRE.

The RMSE values of the three localization algorithms are calculated under the condition expressed by 20 tracking tags, and comparison results are shown in Figure 5. From Figure 5, we can see that BP_LANDMARC has the best global accuracy among the three algorithms in comparison. For

LANDMARC, the largest localization error is 1.78 m, the smallest is 0.25 m, and the standard deviation of errors is ±0.89 m. For VIRE, the largest localization error is 1.33 m, the smallest is 0.009 m, and the standard deviation of errors is ±0.49 m. For BP_LANDMARC, the largest localization error is 0.39 m, the smallest is 0.12 m, and the standard deviation of errors is ±0.25 m, which means that BP_LANDMARC is the steadiest.

To examine the distribution of the localization errors about all the tracking tags, we compared the CDF, $F_e(z) = P(e \leqslant z)$, with the probability of the error e less than z meters for the three algorithms, which are shown in Figure 6. According to our numerical experiments, the $F_e(0.5)$ for BP_LANDMARC, VIRE, and LANDMARC are 0.97, 0.50, and 0.15, respectively.

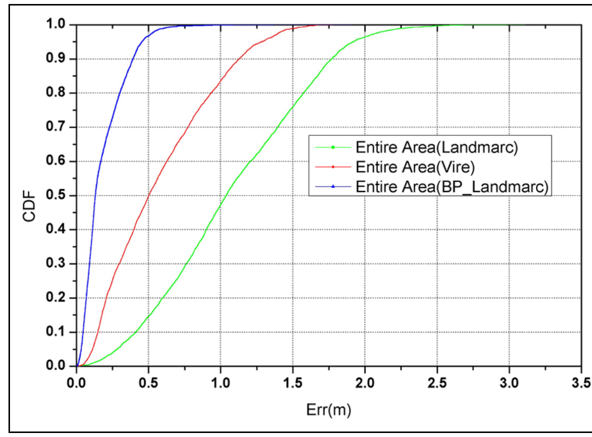

Figure 6. CDF results of three algorithms.

Selection of reference tag number

Theoretically, more reference tags means high localization accuracy. However, the actual testing results show that there is no way to improve the accuracy by only increasing reference tags, as shown in Figure 7. For the testing conditions of this work, the smallest localization error is achieved when the number of reference tags is 6. If the reference tags increase, the localization error also increases.

Under the condition of 20,000 learning times and learning rate $\eta = 0.005$, the learning error curve of the first 3000 training times is shown in Figure 8.

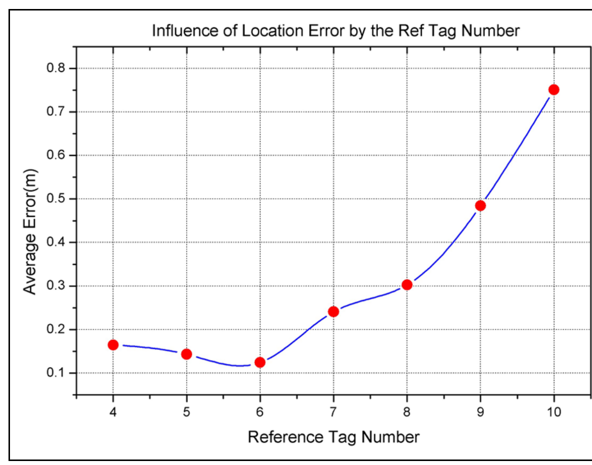

Figure 7. Changing localization accuracy with the number of reference tags.

Application effect on warehouse management system

Indoor object tracing gets more and more applications with the development of smart home and Industry 4.0. Due to the advantages of low cost, efficiency, and ease of deployment, RFID is suitable for applications of indoor object tracing. This section is devoted to a validation of application to locate goods in a warehouse environment. Suppose that a robot needs to access the goods stored on grid shelves, where the central control system senses both robots and goods in a rough granulate and robots could sense only items in their vision scope. In this example, the proposed system is mainly used to locate goods on the grid shelves. In this application, each shelf is a 6 m × 1 m × 4 m cube, and three cubes are grouped into a locating unit with 1 m distance between two adjacent cubes. Hence, the dimension of a locating unit is 6 m × 6 m × 4 m. The illustration of this application is shown in Figure 9. To simplify the problem, we only perform experiments with one locating unit.

To locate the goods, the reference labels are deployed every 1 m in both the horizontal and vertical

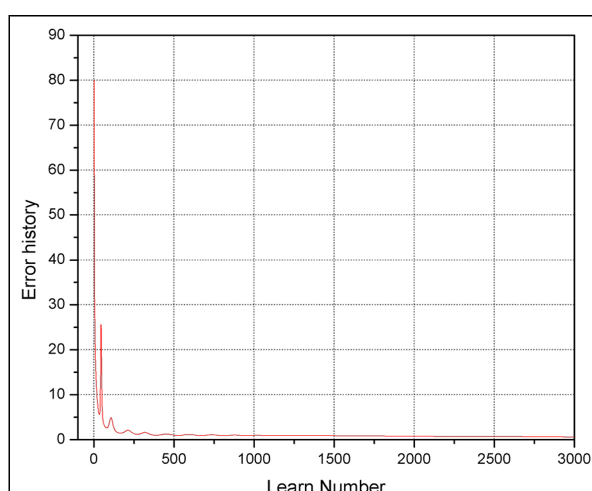

Figure 8. Learning error curve of the first 3000 training times.

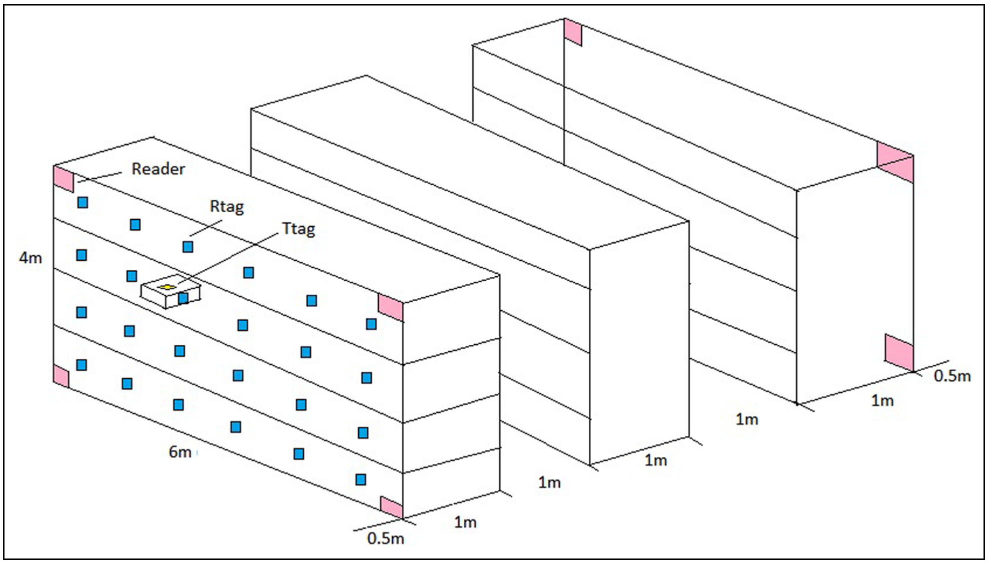

Figure 9. The illustration of a locating unit.

directions and readers are deployed at each corner of the locating unit. Hence, we have 144 ($6 \times 6 \times 4$) reference labels and 8 RFID readers as shown in Figure 9.

The RFID reader is a 4-port UHF fixed one, which can read RFID tags of ultra-high frequency bands in the range of 860–960 MHz, and the maximum communication distance is 10 m. In this application, we label each package of goods with one target tag. When goods are ready to be placed into the shelf, the system should locate an available grid cell of shelves; and when goods are ready to be taken out from the shelf, the system should locate the position of the target tag.

Based on this hardware environment, the proposed BP-LANDMARC algorithm is used to locate grid cell of shelves in the locating unit. The application data show that our design of the RFID localization system can get the locating accuracy of 98.9%. With the application of our system, the number of workers in the warehouse is reduced from 14 to 8. Hence, the efficiency is roughly increased by 42.9%.

Conclusion

This article proposes a kind of indoor locating algorithm, named BP_LANDMARC, based on the BP neural networks combined with the traditional dynamic active RFID calibration method. The proposed algorithm uses the reference tags and the RSSI to locate the targets, differing from the traditional methods on estimation of locating positions using historical data-based learning, but without an explicit experiential equation. To evaluate its validity, we compare the proposed algorithm with two classical algorithms, including LANDMARC and VIRE, under measuring indices of RMSE and CDF. The RMSE index demonstrates that the proposed algorithm could suppress the locating error below 0.39 m, while the errors of LANDMARC and VIRE have reached up to 1.78 and 1.33 m, respectively. Moreover, CDF index demonstrates the superiority of the proposed algorithm with a probability of 0.97 that the error is less than 0.5 m, while for LANDMARC and VIRE the probabilities are 0.50 and 0.15, respectively.

Declaration of conflicting interests

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

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (61472158, 61503150, and 61572228), the Zhuhai Premier Discipline Enhancement Scheme (2015YXXK02), and the Guangdong Premier Key-Discipline Enhancement Scheme (2016GDYSZDXK036) and Science & Technology Development Foundation of Jilin Province (20180201045GX and 20160520012JH).

ORCID iD

Quangang Wen https://orcid.org/0000-0002-7904-7738

References

1. Zhang D, Lu K, Mao R, et al. Fine-grained localization for multiple transceiver-free objects by using RF-based technologies. *IEEE T Parall Distr* 2014; 25: 1464–1475.

2. Ni LM, Zhang D and Souryal MR. RFID-based localization and tracking technologies. *IEEE Wirel Commun* 2011; 18: 45–51.
3. Carotenuto R, Merenda M, Iero D, et al. Ranging RFID tags with ultrasound. *IEEE Sens J* 2018; 18: 2967–2975.
4. Ng JKY, Lam KY, Cheng QJ, et al. An effective signal strength-based wireless location estimation system for tracking indoor mobile users. *J Comput Syst Sci* 2013; 79: 1005–1016.
5. Scherhäufl M, Rudić B, Stelzer A, et al. A blind calibration method for phase-of-arrival-based localization of passive UHF RFID transponders. *IEEE T Instrum Meas* 2018; 99: 1–8.
6. Maneesilp J, Wang C, Wu H, et al. RFID support for accurate 3D localization. *IEEE T Comput* 2013; 62: 1447–1459.
7. Yu HY, Chen JJ and Hsiang TR. Design and implementation of a real-time object location system based on passive RFID tags. *IEEE Sens J* 2015; 15: 5015–5023.
8. Tajima M. Strategic value of RFID in supply chain management. *J Purch Supply Manag* 2007; 13: 261–273.
9. Chang S, Klabjan D and Vossen T. Optimal radio frequency identification deployment in a supply chain network. *Int J Prod Econ* 2010; 125: 71–83.
10. Choi JS, Lee H, Engels DW, et al. Passive UHF RFID-based localization using detection of tag interference on smart shelf. *IEEE T Syst Man Cy C* 2012; 42: 268–275.
11. Xiong TW. *Study on passive UHF RFID-based real time location systems*. Shanghai, China: Fudan University, 2010 (in Chinese).
12. Buffi A, Michel A, Nepa P, et al. RSSI measurements for RFID tag classification in smart storage systems. *IEEE T Instrum Meas* 2018; 67: 894–904.
13. Hightower J, Borrielo G and Want R. SpotON: an indoor 3D location sensing technology based on RF signal strength. *UW CSE Technical Report* 2000; 345–354.
14. Ni LM, Liu YH, Lau YC, et al. LANDMARC: indoor location sensing using active RFID. *Wirel Netw* 2004; 10: 701–710.
15. Ma YW, Lai CF, Hsu JM, et al. RFID-based positioning system for telematics location-aware applications. *Wireless Pers Commun* 2011; 59: 95–108.
16. Zhao YY, Liu YH and Ni LM. VIRE: active RFID-based localization using virtual reference elimination. In: *Proceedings of the international conference on parallel processing*, Xi'an, China, 10–14 September 2007, p.56. New York: IEEE.
17. Aldin NB, Erçelebi E and Aykaç M. An accurate indoor RSSI localization algorithm based on active RFID system with reference tags. *Wireless Pers Commun* 2017; 97: 3811–3829.
18. Holland WS, Young WA II and Weckman GR. Facility RFID localization system based on artificial neural networks. *Int J Ind Eng* 2011; 18: 16–24.
19. Deak G, Curran K, Condell J, et al. Support vector machine and probability neural networks in a device-free passive localization (DFPL) scenario. *Image Process Commun* 2013; 17: 9–16.
20. Yang ZX, Zhang PB and Chen L. RFID-enabled indoor positioning method for a real-time manufacturing execution system using OS-ELM. *Neurocomputing* 2016; 174: 121–133.
21. Zhao YY. *Localization and tracking based on RFID technologies*. Hong Kong, China: Hong Kong University, 2010.
22. Shao CG. *The study of indoor location algorithm based on RFID technology*. Beijing, China: Beijing University, 2013 (in Chinese).
23. Wu L and Huang LY. Improvement of location methods based on RFID. *J Chin Univ Post Telecommun* 2013; 20: 36–41.
24. Huang YH, Lui ZY and Ling GJ. An improved Bayesian-based RFID indoor location algorithm. In: *Proceedings of the international conference on computer science and software engineering*, Hubei, China, 12–14 December 2008, pp.511–514. New York: IEEE.
25. Huang CH, Lee LH, Ho CC, et al. Real-time RFID indoor positioning system based on Kalman-filter drift removal and Heron-bilateration location estimation. *IEEE T Instrum Meas* 2015; 64: 728–739.
26. Aldin NB, Erçelebi E and Aykaç M. Advanced boundary virtual reference algorithm for an indoor system using an active RFID interrogator and transponder. *Analog Integr Circ S* 2016; 88: 415–430.
27. Chiou YS, Wang CL and Yeh SC. An adaptive location estimator using tracking algorithms for indoor WLANs. *Wirel Netw* 2010; 16: 1987–2012.
28. Moreno-Cano MV, Zamora-Izquierdo MA, Santa J, et al. An indoor localization system based on artificial neural networks and particle filters applied to intelligent buildings. *Neurocomputing* 2013; 122: 116–125.
29. Soltani MM, Motamedi A and Hammad A. Enhancing cluster-based RFID tag localization using artificial neural networks and virtual reference tags. *Automat Constr* 2015; 54: 93–105.
30. Kung HY, Chaisit S and Phuong NTM. Optimization of an RFID location identification scheme based on the neural network. *Int J Commun Syst* 2015; 28: 625–644.