Wireless Networks (2024) 30:5503–5518
https://doi.org/10.1007/s11276-023-03265-0

Improved 3D localization algorithm for large scale wireless sensor networks

Rahma Mani[1,2] · Antonio Rios-Navarro[2] · Jose-Luis Sevillano-Ramos[2] · Noureddine Liouane[1]

Accepted: 30 January 2023 / Published online: 13 March 2023
© The Author(s) 2023

Abstract
As localization represents the main core of various wireless sensor network applications, several localization algorithms have been suggested in wireless sensor network research. In this article, we put forward an iterative bounding box algorithm enhanced by a Kalman filter to refine the unknown node's estimated position. In fact, several research efforts are currently in progress to extend the 2D positioning algorithm in WSNs to 3D that reflects reality and the most practical applications. Subsequently, we replace a large number of GPS-equipped anchors with a single mobile anchor. In our studies, we consider the type of range-free sensor network exploiting the wireless sensors connectivity. We assess the performance of our algorithm using exhaustive experiments on several isotropic and anisotropic topologies. Our proposed algorithm can fulfill the joint goals of algorithm transparency and accuracy for various scenarios by evaluating parameters such as localization accuracy whilst changing other simulation parameters such as the effect of communication range, mobile anchor node position and sensor node deployment topology. It has been proven by the results of the experiments that the proposed algorithm effectively reduces the location error without requiring more equipment or increasing the communication cost.

Keywords Localization · Mobility · 3D · Bounding_box · Wireless sensor networks · DV_Hop · Kalman filter

1 Introduction

The wireless sensor network is composed essentially of sensors capable to collect and transmit data autonomously. The sensors can be deployed randomly in a defined area to detect a phenomenon. The applications of this particular ad-hoc network, wireless sensor network (WSN), are invading many fields such as the IoT, IIOT, e-health, e-logistics, e-agriculture, etc. Some of these applications have potential, such as surveillance of natural parks, surveillance of nuclear power plants, forest fire detection, etc. The main purpose of almost all the applications in WSN is to determine when and where a marked event occurs which is considered an identifier of that occurrence. Some of the nodes constituting a WSN environment are considered anchors that have a stable source of location information via the global positioning system (GPS). For computational processing scenarios, we consider the advancement of power management strategies for better solutions in particular for low-power WSN using an alternative method than using several GPS devices for a low cost and computational effort. In this work, our main goal is to find an efficient algorithm to achieve highly accurate data and higher localizable nodes with lower computational effort, memory size and energy. We are concerned with the strategy of a mobile anchor node to reduce hardware costs. Our proposal is based on the bounding box approach to identify the possible location area of the unknown node. Also, the proposed approach

✉ Rahma Mani
rahmam@us.es

Antonio Rios-Navarro
arios@us.es

Jose-Luis Sevillano-Ramos
jlsevillano@us.es

Noureddine Liouane
noureddine.liouane@enim.rnu.tn

[1] Electrical Engineering Department, University of Monastir, 5019 Monastir, Tunisia

[2] Architecture and Computer Technology, University of Seville, 41012 Seville, Spain

uses an infinite impulse response filter which is a Kalman filter that approximates the states of a dynamic system from a set of noisy or incomplete information. For wireless sensor network localization problem, the KF algorithm iteratively refines the positions determined by the bounding box algorithm (BBX) in order to improve the accuracy of the undetermined node positions. Actually, KF uses the dynamic model of the geolocation system, modeled by state equations and onboard sensor measurements, to obtain an estimate of the state vector that minimizes the covariance error. In effect, an important interest oriented to mobility that has a major role in such networks. The combination of mobile systems and WSNs creates what we will be working on, the mobile wireless sensor networks (MWSNs). In a large-scale network, the cost value is important for the application's deployment. MWSNs help to solve this problem by decreasing the number of anchors which costs more than normal nodes. Therefore, every localization process needs an anchor node to provide information and references to unknown nodes. Consequently, using the range-free method, specifically, the connectivity process of all nodes estimates their inter-distance via the calculation of the lowest number of hops. Thus, MWSN is capable of deploying a single anchor in the network and determining the positions of unknown nodes based on information about the number of hops and the path of the anchor. The mobile anchor node trajectory could be random or has a predetermined path planning. The purpose is to calculate the connectivity information and then try to compute the positions of unknown nodes using an iterative bounding box. Besides, using a highly utilized filter, Kalman Filter, to refine the localization process and improve localization accuracy. The rest of the paper is organized as follows: After the introduction in Sect. 1, the related works will be presented in detail in Sect. 2. As an introduction to our algorithm, we represent in Sect. 3 the basics of the Kalman filter in general. Section 4 is about the proposed algorithm. In Sect. 5, a comparative study between the experimental results has been presented and analyzed. Finally, we discuss this work and conclude this paper and we put forward the future work in Sect. 6.

2 Related works

Wireless sensor networks (WSNs) have become increasingly prevalent in the Internet of Things (IoT) and Industrial Internet of Things (IIoT) due to their ability to exploit the capabilities of 5 G networks. Many research efforts have been focused on addressing various challenges related to WSNs, including location tracking, routing, task allocation, spectral optimization, energy optimization, and more. To improve the performance of these networks, researchers have utilized a range of techniques including machine learning and neural networks [1–3]. Various types of fields deployed in WSNs require localization techniques to be used in different applications. Some applications such as healthcare treatment, environmental monitoring, and agricultural management require the deployment of nodes in three-dimensional (3D) irregularly shaped fields. The localization of sensor nodes in different types of fields is a challenge [4]. In 3D fields, network connectivity can be more diverse compared to the 2D scenarios and WSN boundary conditions become more important [5]. Range-free localization algorithms are most appropriate for large-scale WSNs since they do not require any additional hardware on the sensor nodes to be used to measure the radio signal characteristics. The most well-known of these algorithms are techniques based on hops [6]. These types of algorithms use only the connectivity between nodes to determine the minimum number of hops and subsequently to estimate the distance [7]. Hop-based techniques use the length of hops between sensor nodes and reference nodes for the estimation of the location. Also, mobile localization techniques do not need additional hardware and estimate the node site area sequentially. The localization with a mobile beacon in WSN was proposed in various research works in literature as a solution for sensor nodes to compute their positions in real time. In fact, the unknown nodes receive three or more signals from the same beacon node from different locations and calculate their locations by applying a trilateration or multilateration algorithm [8]. In this way, we are defining some of the popular range-free methods employed for localization for WSN with the presence of mobility. We can mention the most well-known method Monte Carlo localization (MCL) algorithm proposed by Lingxuan Hu et al. [9]. Using one mobile anchor [10], the authors presented the mobile anchor centroid localization (MACL) algorithm for localization in WSNs. The described algorithm is based on radio frequency connectivity and the centroid method for preliminary localization via the single mobile anchor. The considered path planning for the localization process is spiral trajectory. The former algorithm produced acceptable results for some WSN deployment scenarios, but due to the filtering operation, integrated into the algorithm, it is possible to reject the effect of some anchor positions to improve the localization accuracy. In [11], the mobile anchor assisted localization algorithm based on regular hexagon (MAALRH) is presented. This method is designed based on the regular hexagon in 2D-WSNs which maximizes the ratio of localized nodes and minimizes the average localization errors. Moreover, for overcoming the irregularity of the communication range and the co-linearity problem, a boundary compensation method (BCM) is described. Furthermore, different path planning for a

mobile anchor is presented like HILBERT, CIRCLE and S-CURVE. The research work in [12] presents a mobile anchor node for localization using the trilateration method and path planning choice with two objective functions such as the reduction of localization errors and minimizing the energy consumption for WSNs. Moreover, the inter-distance between the mobile anchor and the unknown nodes is estimated with the RSSI indicator. In addition, the localization errors are analyzed for five trajectories for mobile anchors like SPIRAL, SCAN, DOUBLE SCAN and HILBERT path. The authors of the paper [13] proposed a collaborative localization algorithm with bounding restrictions for wireless mobile networks. The proposed technique implies trajectory prediction using Newton's polynomial interpolation and bounding box. Earlier, most localization techniques were based on numerical computation of many network parameters such as transmitter or receiver power, network connectivity information, transmission range, transmission and reception time, etc. These parameters have many restrictions, such as environmental conditions and complications related to the environment. The Monte Carlo boxed (MCB) [14] localization method has been proposed by Baggio Aline and Koen Langendoen where the samples are defined by constructing a box on the basis of MCL. This algorithm accelerates the design of position samples, which improves localization accuracy. Many improvements of MCB are presented in the literature, such as RSSI-MCB [15], IMCB [16] and MCB-PSO [17]. Specifically, in range-based cases, many localization techniques in the literature have concentrated on minimizing localization errors during the localization operation. In [18], the uncentered Kalman filter (UKF) and particle filter (PF) were presented based on the Kalman filter (KF) for localization. The published localization technique can be used for any kind of wireless sensor network problem and in particular for Autonomous robot localization.

3 The basics of Kalman-filter

The Kalman filter is a method to estimate parameters of a system that evolves in time from noisy measurements. This filter can be found in many fields related to signal processing, radar, image processing, etc. An example of the use of this filter could be the determination of the position and speed of a vehicle from GPS data provided by several satellites. The power of this filter is its capability to fix errors and forecast parameters, both from the sensors and from the actual model. In fact, in order to implement a Kalman filter, the system of which we want to estimate its parameters must be modeled in a linear way. Variants of the Kalman filter exist to take into account non-linear models. In a conventional approach for estimation, such as the least squares method, a single error in the modeling of the system necessarily leads to an error in the estimation. The strength of the Kalman filter is that it integrates an inaccuracy element on the actual model, making it possible to obtain correct estimations in spite of modeling errors (as long as these errors remain reasonable). Another strength of the algorithm is its capacity to determine the average error of its estimate. Indeed, the Kalman tool provides a matrix of the estimated parameters, as well as a covariance matrix of the error. This provides information about the accuracy of the estimate, which might be valuable in many different purposes. Another advantage of the Kalman filter is that the convergence of this error is guaranteed. The Kalman filter process can be divided into two steps. The first step of prediction is according to the model of the system. For this, the Kalman filter uses the previous estimation of values for the system's parameters and errors and predicts the new parameters and errors based on the system model. The second step will update this prediction with the new measurements. These measurements will allow obtaining an estimate of the parameters and the error from the prediction model. The whole process is illustrated in Fig. 1.

If the model contains errors, this stage of the update will make it possible to correct them.

4 The proposed algorithm

Kalman filter is generally widely used with range-based methods for filtering the Received Signal Strength Indication 'RSSI' signal to properly estimate distances. Indeed, in our case, it is important to carefully consider features of the hop-based methods such as network coverage, node mobility and communication range, in order to ensure that the range-free mobile anchor sensor network is able to effectively cover and monitor the large scale domain, as well as achieve accurate nodes location in different scenarios. In fact, we propose a novel localization strategy for wireless sensor networks with a mobile anchor node, utilizing a combination of a Kalman filter and bounding box localization algorithm (BBLA-KF). Our approach utilizes connectivity measurement and the nonlinear bounding box techniques to identify the bounded possible location zone for the unknown nodes, and then applies the kalman filter to minimize uncertainty in the connectivity process and achieve high levels of accuracy in determining the bounded-boxes. This innovative approach holds the potential to significantly improve localization accuracy in wireless sensor networks in isotropic and anisotropic 3-Dimensional fields.

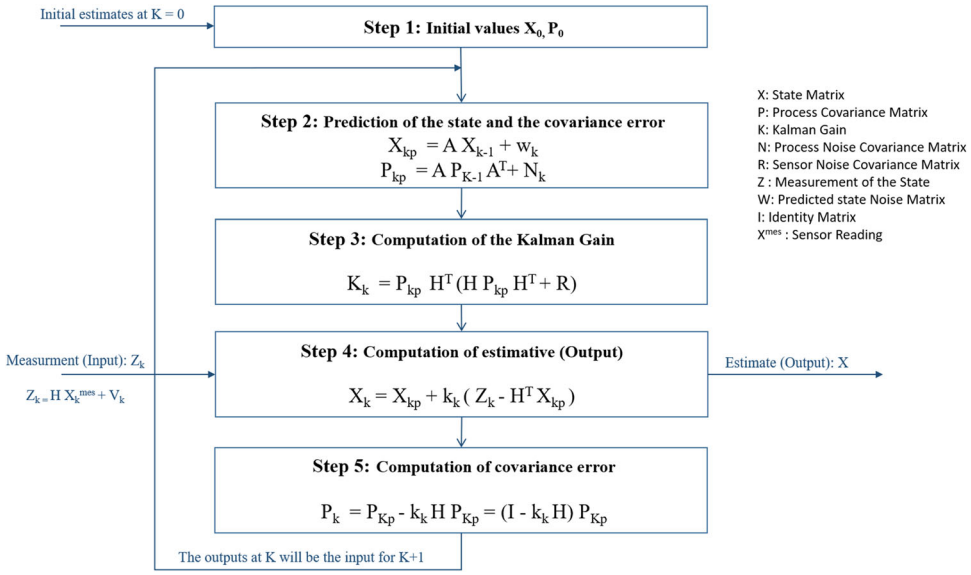


Fig. 1 The Kalman-filter algorithm

4.1 Mobility method

The WSN exploits one mobile anchor that moves in any direction following a planned path in the deployment area. The mobile anchor node can be a vehicle, a robot or a drone, etc., which is equipped with GPS. As in our case as illustrated in Fig. 2, the mobile anchor try to detect the occurrence of target events inside the region of interest and then assess the positions of the target events.

The mobile anchor node is able to tackle the problem of the low density of anchors. Communication in a mobile network does not cost much because only one anchor communicates its position information. Subsequently, the rest of the nodes assess their positions using the information provided by the mobile anchor. At each position k of the mobile anchor, as it travels through the network, the anchor node transmits a packet to its neighboring nodes to inform them of its current position (x_{Mk}, y_{Mk}, z_{Mk}) and the minimum hop-count H_k, where $k = 1, 2, \ldots N$. Figure 3 is a model illustrating an example that calculates the hop count of the unknown node UN_6.

4.2 Mobile 3D bounding box algorithm 3D-MBBX

The bounding box algorithm makes reference to the construction of a Monte Carlo box (MCB) with a range of k hops considered. Using a mobile anchor for the localization process is a good alternative to the use of multiple anchor nodes and reduces the complexity of the wireless

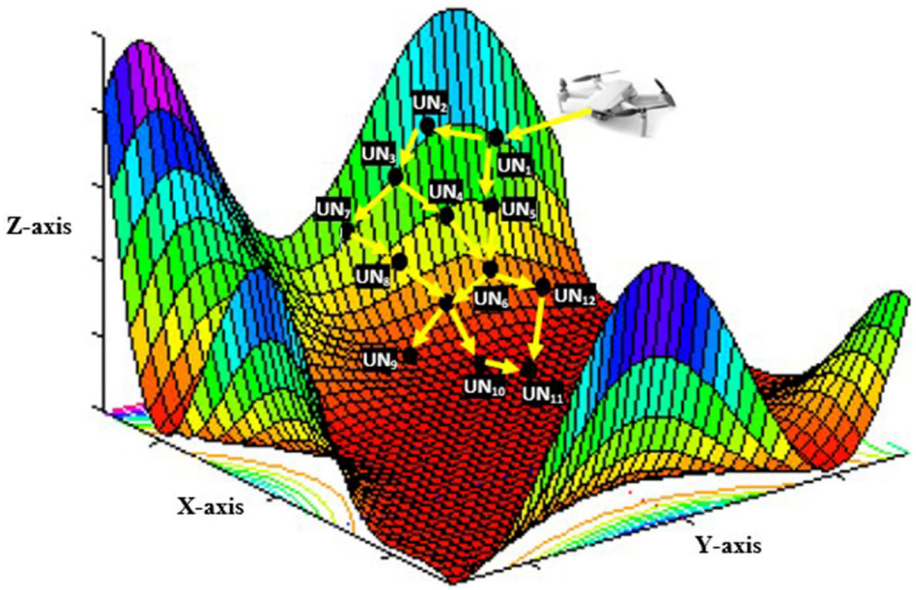


Fig. 2 Localization provided by mobile anchor node

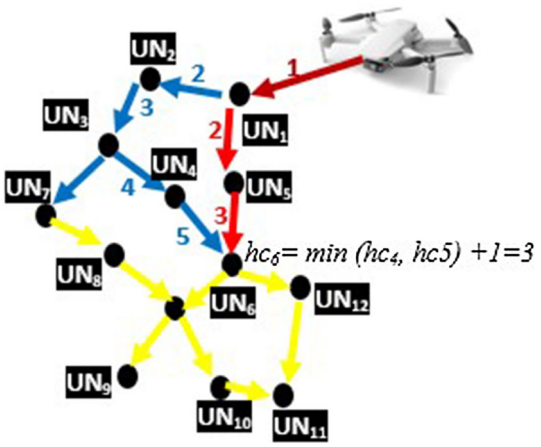


Fig. 3 Flooding phase and hop-count values

sensor network. Each time the anchor node travels in the network, it distributes its information to the connected unknown nodes. With repeated occurrences of a moving anchor, unknown nodes recursively approximate their detection area with a rectangular bounding box and enhance location accuracy over time. The bounding box proposed in our algorithm uses cubes to make an estimate of the locations of the unknown node extended over the 3D structure of the sensor nodes. Figure 4 shows the principle of the moving bounding box algorithm.

We notice that each unknown node possesses its own unique search area computed by the anchor in its transmission range. The purpose of using the bounding box is to determine the search space of the unknown node, which is intended to minimize the complexity along with the computation time compared to the trilateration method. Let us regard a connectivity-based wireless sensor network composed of N_n nodes that are distributed randomly in the area of interest. The communication range of each sensor node is set equal to R and the connected sensor nodes are all networked. As with the original DV-Hop algorithm, the minimum number of hops between the nodes and the mobile anchor node is estimated. The 3D bounding box method searches the possible area where the unknown node can be placed. Network connectivity measurement and bounding box localization method are used to identify the bounded possible location zone. The 3D bounding box method algorithm exploits cubic shapes as an alternative to spherical shapes. The 3D-MBBX limits the possible area of a node's location. In Fig. 4, we differentiate between the MQ_k anchor boxes and the Q_N bounding box. For each position of the moving anchor M_k, an anchor box MQ_k is set as a cube whose center lies at the position of the anchor node (x_{Mk},y_{Mk},z_{Mk}), with edges of size $2 \times h_{Mk} \times R$ where h_{Mk} is the minimum number of hops between the mobile node M_k at each position $k = 1, 2, 3 \ldots N_a$ and the unknown nodes.

The coordinates of the anchor box MQ_k are defined by the following system of equations:

$$\begin{cases} x_{MK}^{-} = x_{MK} - h_{MK} \times R \\ x_{MK}^{+} = x_{MK} + h_{MK} \times R \\ y_{MK}^{-} = y_{MK} - h_{MK} \times R \\ y_{MK}^{+} = y_{MK} + h_{MK} \times R \\ z_{MK}^{-} = z_{MK} - h_{MK} \times R \\ z_{MK}^{+} = z_{MK} + h_{MK} \times R \end{cases} \quad (1)$$

For each anchor node M_K, we define the box range MQ_k which defines the location of the unknown node as follows:

$$MQ_k = \begin{bmatrix} x_{Mk}^{-} & x_{Mk}^{+} \\ y_{Mk}^{-} & y_{Mk}^{+} \\ z_{Mk}^{-} & z_{Mk}^{+} \end{bmatrix} \quad (2)$$

The bounding box Q_k determines the intersection of all

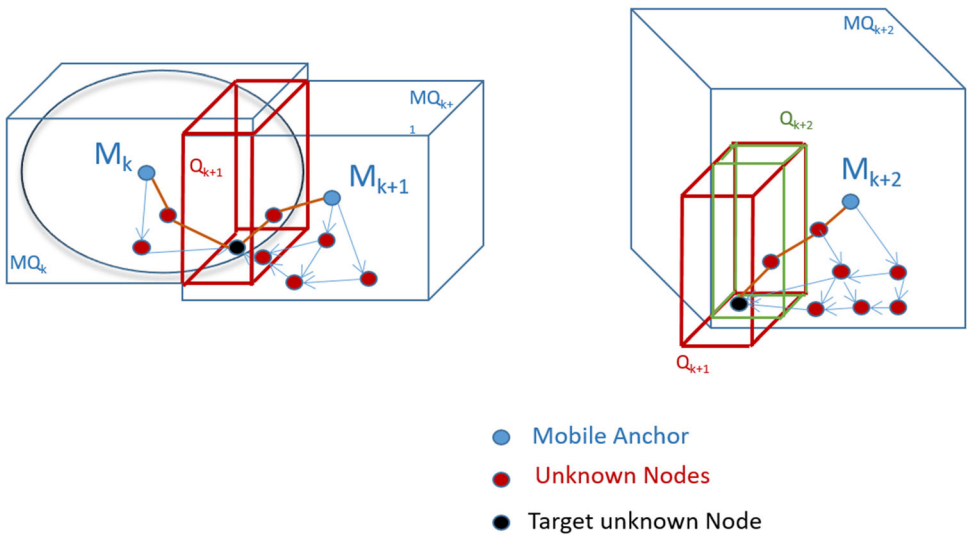


Fig. 4 3D bounding box localization method for 3D-WSN

anchor boxes MQ_k via the Max-Min algorithm as mentioned in the following form:

$$Q_k = \begin{bmatrix} max(x_{MK}^-) & min(x_{Mk}^+) \\ max(y_{MK}^-) & min(y_{Mk}^+) \\ max(z_{MK}^-) & min(z_{Mk}^+) \end{bmatrix} \tag{3}$$

So let Q_k be the bounding box of an unknown node UN_i where i is between 1...N_n. Subsequently, the node will upgrade its bounding box using the constraint set by the anchor M_k when the anchor moves to M_{k+1} at position (x_{Mk+1}, y_{Mk+1},z_{Mk+1}). The recursive rules for updating the bounding box are given by:

$$\begin{aligned} Q_k &= MQ_1 \cap MQ_2 \ldots MQ_{k-1} \cap MQ_k, \quad \forall k = 1 \ldots N_a \\ &= Q_{k-1} \cap MQ_k \end{aligned} \tag{4}$$

then,

$$Q_k = \begin{bmatrix} x_k^- & x_k^+ \\ y_k^- & y_k^+ \\ z_k^- & z_k^+ \end{bmatrix} \tag{5}$$

The final location of the unknown node UN_i is then calculated as the centroid of the resulting Q_k bounding box.

Afterwards, the measured location $X^{mes} = (x^{mes}, y^{mes}, z^{mes})$ of the unknown node UN_i is yielded by the centroid of the bounding box of Q_k.:

$$X^{mes} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x^+ \\ x^- \\ y^+ \\ y^- \\ z^+ \\ z^- \end{bmatrix} \tag{6}$$

4.3 Kalman filter refinement process

The Kalman filter represents an optimal algorithm to predict the location of the unknown nodes in the presence of noisy measurements. The Kalman Filter is then used to minimize uncertainty produced by the connectivity process. Indeed, through time, for each anchor node position, the Kalman filter refines the measured coordinate of the unknown node position X_k^{mes} which is determined by the mobile bounding box localization approach with the anchor positions M_k and the covariance of the perceived noise. The Kalman filter recursively approximates the unknown node position X_k through the discrete-time equation of state:

$$X_k = AX_{k-1} + w_k \tag{7}$$

wherein $X_k = (x_k, y_k, z_k)^T$ is the approximate position of the unknown node when the anchor is at station k, the vector of process noise w_k considers the perturbations on the position of the unknown node. Similarly, the noise of the process adopts a Gaussian distribution modeled by a random noise vector of mean zero and a covariance matrix N_k. In this scenario, we consider the position of the unknown node as fixed and unaltered during the localization process, so the value of w_k is null. The transition matrix $A_{3\times3}$ = Identity Matrix. For this system, the measurement equation is visualized as follows:

$$Z_k^{mes} = HX_k^{mes} + v_k \tag{8}$$

$vk \sim N(0, R_k)$ where $X_k^{mes} = (x^{mes}, y^{mes}, z^{mes})^T$ is the measurement vector given by the moving bounding box process $Q_k = Q_{k-1} \cap MQ_k$ which estimates the measurements at position k of the anchor in which X_k^{mes} is the centroid of Q_k. This step is posterior to the flooding phase where the hop-counts between anchor M_k and unknown nodes are found. H is the observation matrix with $H_{3\times3}$ = Identity Matrix and v_k is the vector of observation noise supposed to be a normally distributed vector of random variables with zero average and covariance matrix R. Considering the two models the prediction equation and the update equation, the Kalman filter algorithm iteratively follows the evolution of the state via the following flowchart (Fig. 5).

5 Simulation results and discussion

The performance evaluation of our BBLA-KF algorithm is performed by studies in the case of a 3D-WSN for an isotropic shape and in the case of a 3D-C form for an anisotropic shape. The impact of the change of the 3D-WSN properties on the results of the localization is analyzed by evaluating the average localization error (ALE) that is presented by the next expression:

$$ALE = \frac{\sum_1^N \sqrt{(x_{k,mes} - x_k)^2 + (y_{k,mes} - y_k)^2 + (z_{k,mes} - z_k)^2}}{R \times N} \tag{9}$$

This section presents some metrics that compare our proposed method's performance with other methods in different situations. Performance measures are produced by simulation through the MATLAB software, the results are compared to other algorithms of localization like RRBL [19], OIDVHop [20], PERLA [21], I3D-DVLAIN [22] and DV-Hop-3D [23]. These algorithms are selected as a combination of the classical and the latest hop-based methods. Consequently, we evaluated the reported BBLA-KF localization algorithm at different isotropic conditions and node densities, following a similar simulation methodology, albeit with more in-depth analysis, to that used in [24] for a simpler two-dimensional protocol. In all

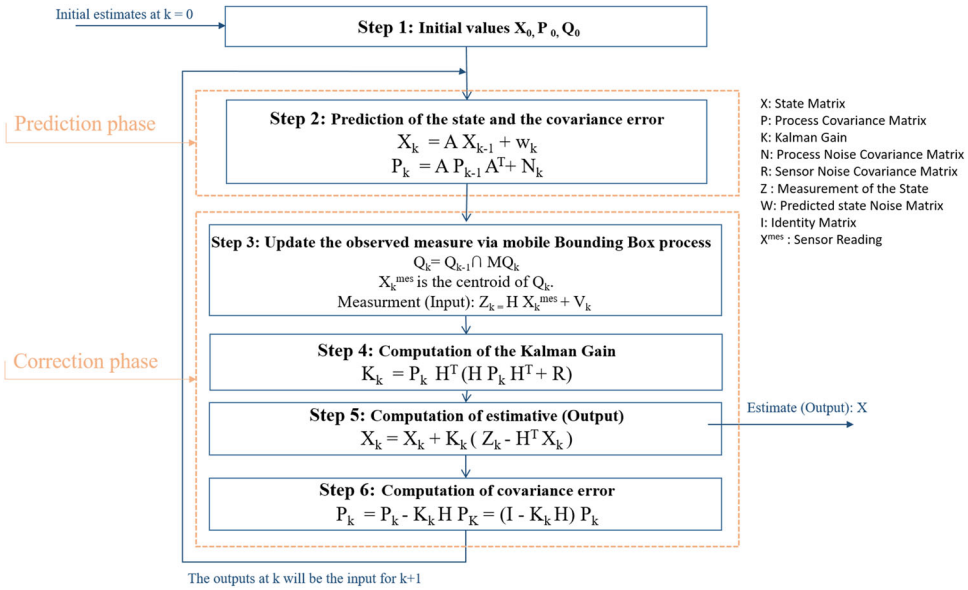


Fig. 5 The proposed BBLA-KF algorithm

our experiments, we consider that 400 sensor nodes are uniformly and randomly distributed in a detection area of 100 m×100 m×100 m where there are no obstacles and considering always a communication range error between 0–10%. A mobile anchor node is set up to handle the location of unknown nodes. The trajectory of the mobile anchor node is considered a random waypoint (RWP). In our simulation case, the positions of the mobile anchor are considered as a set of virtually fixed anchors.

5.1 Experimental analysis of positioning

We use MATLAB to build our sensing area, in order to follow up on the point that all nodes are able to be located. We distribute 340 unknown nodes and 60 anchor nodes randomly and uniformly in the deployment zone.

Nodes distribution is shown in Fig. 6, where yellow stars represent anchor node positions, and red (*) represent unknown nodes. We examine our method in the environment shown in Fig. 6. Then, we get the result of MATLAB simulation shown in Fig. 7, where (*) is the true node location and (o) the estimated location.

From the graph, we can see that the BBLA-KF method can locate the unknown nodes very well, and give a higher precision. Figure 8 is the result obtained after conducting a number of tests for evaluating the convergence of the BBLA-KF algorithm towards the real positions. From the figure, we can see the efficiency of our proposed algorithm to converge.

Figure 9 give the convergence of ALE over the number of iterations. This behavior may be due to the use of the Kalman filter in position adjustment, which utilizes a set of observed measurements through time, including statistical noise and other imprecisions, and generates estimates of unknown variables that tend to be more precise.

Figure 10 shows the location error of all unknown nodes. It can be seen that the biggest error does not exceed 7 m equivalent to 40% of R, communication range. The error range is acceptable for mobile WSN Localization.

5.2 Comparison of the localization accuracy of different localization techniques

Following, we present some simulation results that compare the localization accuracy performance of our proposed algorithm BBLA-KF versus RRBL, OIDVHop, PERLA, I3D-DVLAIN and DV-Hop-3D. We perform simulations to compare the performance in the same scenario set in Sect. 3. The I3DDVLAIN, OIDV-Hop, PERLA, and DV-Hop-3D algorithms were run for one hundred aleatory displays and the percentage of localization error for all trials is shown in Table 1. Compared to other existing algorithms, the results of our algorithm are better. When we look at the average value at the end of 100 trials, we find that BBLA-KF has an average of 23% while I3D-DVLAIN has an average LE of about 25%, DV-Hop has about 37%, OIDV-Hop has about 39%, and PERLA has about 35%. The resulting localization error of BBLA-KF is the lowest among the five techniques compared. This shows that our newly proposed algorithm is much more accurate than the other algorithms. We may justify this with the fact that our proposed algorithm is built on both the bounding box method and the Kalman filter to match the estimated position. Therefore, BBLA-KF performs better in terms of localization accuracy.

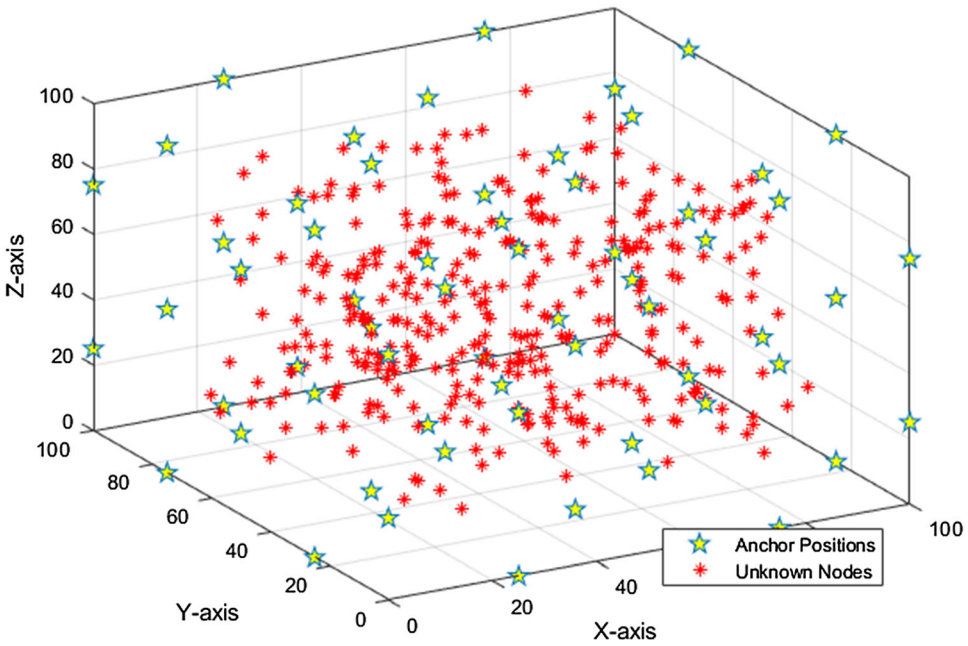


Fig. 6 3D-WSN nodes distribution

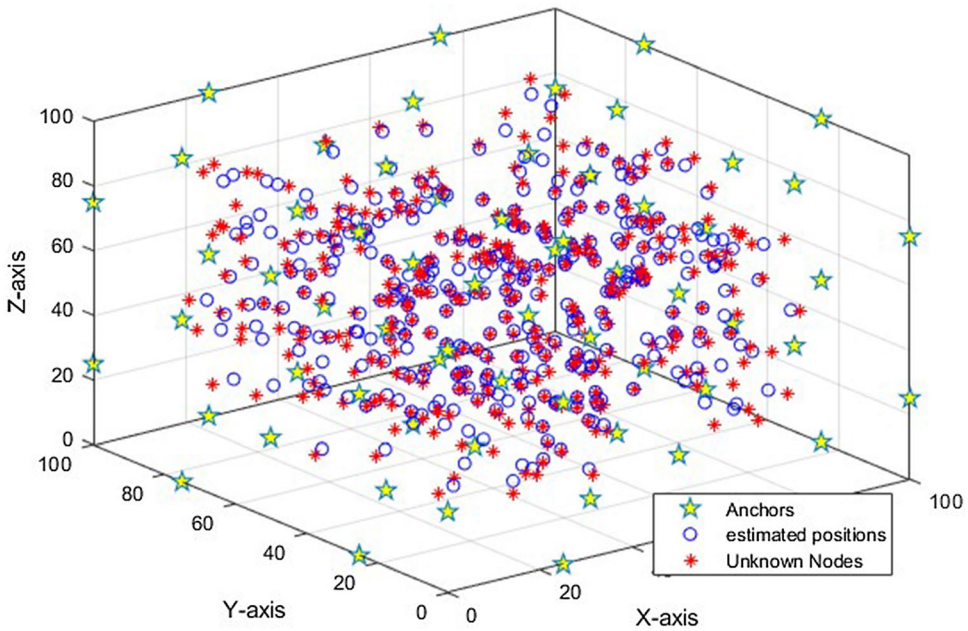


Fig. 7 Localization results of our proposed algorithm

5.3 Influence of the variation of the number of nodes

The performance of the presented algorithms is evaluated for various node densities in 3D fields. The number of nodes (anchor nodes and unknown nodes) is increased from 100 to 800 and the ratio of reference nodes is kept at 15% with a communication range of 18 m. The simulation parameters are presented in Table 2.

The simulation results of the influence of the number of nodes are illustrated in Fig. 11.

We notice that the ALE for BBLA-KF is smaller than that of I3D-DVLAIN, OIDV-Hop, PERLA and DV-Hop-3D for the same value of the total number of nodes. The ALE for DV-Hop-3D, OIDV-Hop, and PERLA decreases slowly as compared to BBLA-KF. The grade in ALE is very significant from 100 to 300 nodes for BBLA-KF, which means that the rate of ALE improvement was elevated at lower node densities and the variation in ALE was decreased at greater node densities. Also, there is a fall of around 40% in ALE for BBLA-KF as compared to DV-Hop-3D and a fall of 20% with respect to I3D-DVLAIN, as

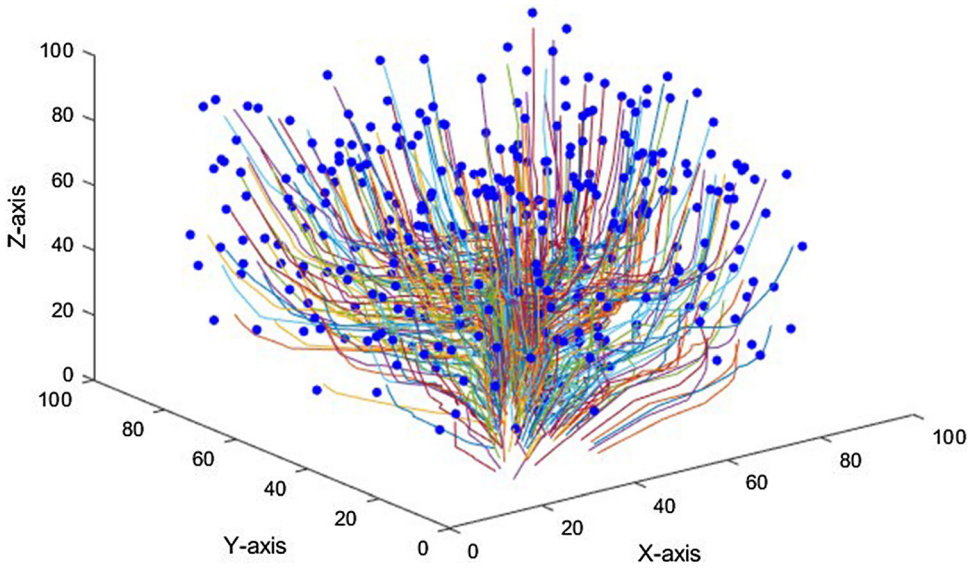


Fig. 8 Convergence of BBLA-KF algorithm towards the real positions

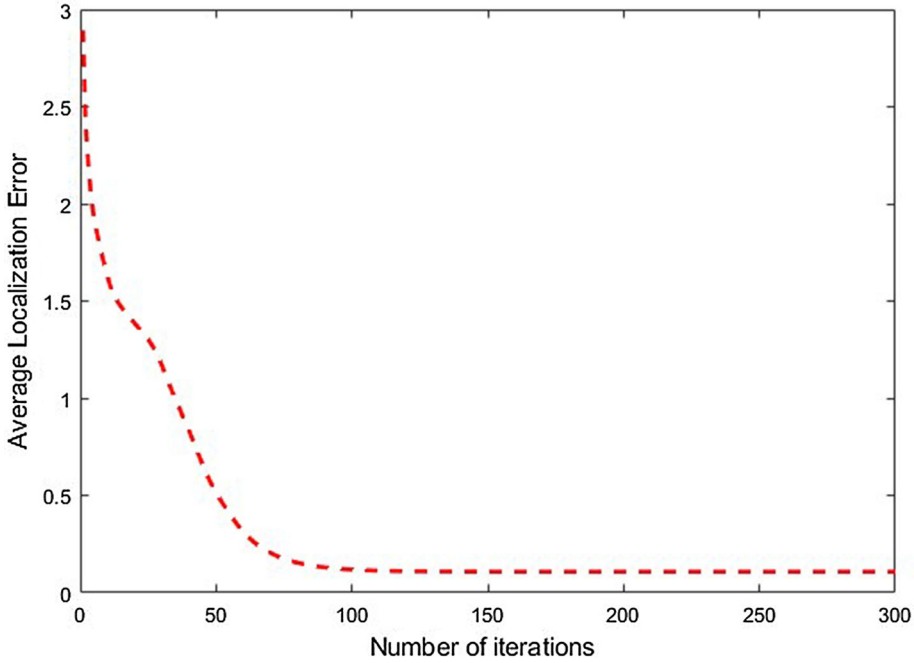


Fig. 9 Average localization error of unknown nodes over the number of iterations

seen in Fig. 11. Figure 12 gives the cumulative distribution functions (CDFs) of the average localization error for different WSN densities. For example, in this figure, we notice that for 100 nodes, 14% of the unknown nodes are localized with an error less than 0.25 of the communication range "R", while for 200 nodes, only 42% of the unknown nodes are localized with an error inferior to 0.25R. However, for 400 nodes the localization error is 91% and for 800 nodes, all the unknown nodes are localized with an error less than the previous error range. Next, Table 3 resume the CDFs of localization error for different number of nodes.

We can interpret from the Table 3 that for the case of 100 nodes, there are some nodes non-localizable which is explained by the fact that the localization error is greater than the communication range. We also find that as the number of nodes decreases, the CDFs tend to decrease. This may be due to the growth in network connectivity, which is intended to give an accurate estimate of the distance between nodes in large-scale WSNs. In conclusion, we can say that increasing the density of nodes improves the performance of the algorithm by improving the accuracy.

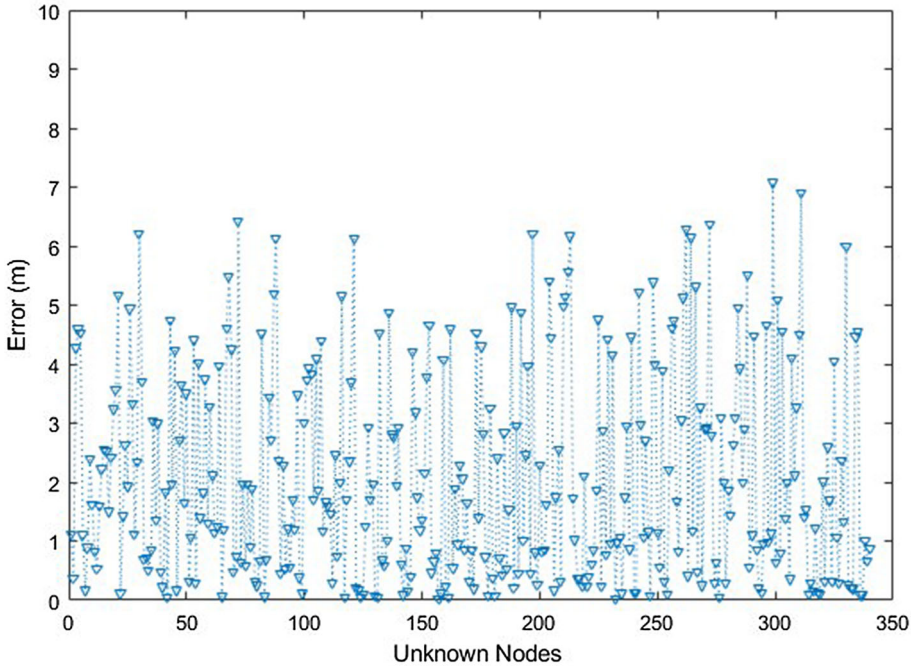


Fig. 10 The error given by each unknown node

Table 1 Localization error results of hundred trials

	Localization erorr		
Localization algorithms	Max	Min	Mean
BBLA-KF	0.25	0.10	0.23
I3D-DVLAIN	0.37	0.04	0.25
PERLA	0.38	0.27	0.35
OIDV-Hop	0.45	0.35	0.39
DV-HOP-3D	0.42	0.25	0.37

Table 2 Simulation parameters of nodes number variation

Parameter	Value
Sensing field	100 m × 100 m × 100 m
Communication range	18 m
Node amount	100, 200, 300, 400, 500, 600, 700, 800
Anchor ratio	15 %

5.4 Influence of anchor node positions ratio

The communication range of the sensor nodes has been set to 18 m with an error ranging from 0 to 10%. The total number of sensors in the network is set to 400. Therefore, we vary the anchor node positions ratio from 5 to 35% from the total number of nodes and analyze the result with respect to its localization error. Figure 11 shows a comparison of the localization error of the algorithms I3D-DVLAIN, DV-Hop-3D, OIDV-Hop, PERLA and BBLA-KF. It can be clearly seen that as the ratio of anchor node positions increases, the localization error of the different algorithms decreases because the unknown nodes receive more accurate information due to the multiple anchor node positions. Similarly, for various values of anchor node position ratio, the localization error of BBLA-KF is always lower than that of other algorithms due to the position correction applied by the Kalman filter for unknown nodes (Fig. 13).

5.5 Effect of communication range

Next, we analyze the effect of varying communication range on the average location error (ALE). The communication range of sensor nodes varies from 15 to 45 m. The total number of nodes has been kept to 400 which contains 15% of anchor node positions. Figure 14 presents the effect of the variation in the communication range on the localization accuracy by comparing with I3D-DVLAIN, DV-Hop-3D, OIDV-Hop and PERLA algorithms. The ALE of BBLA-KF is lower than that of I3D-DVLAIN, DV-Hop-3D, OIDV-Hop and PERLA for each value in the communication range. The decrease in ALE is more severe in the 15–25 m communication range than in the 35–45 m range, where the ALE value almost normalizes. There is an approximately 40% decrease in overall ALE for BBLA-KF compared to the I3D-DVLAIN algorithm. The effect of varying communication range on the ALE for our algorithm is also smaller compared to other algorithms. Based

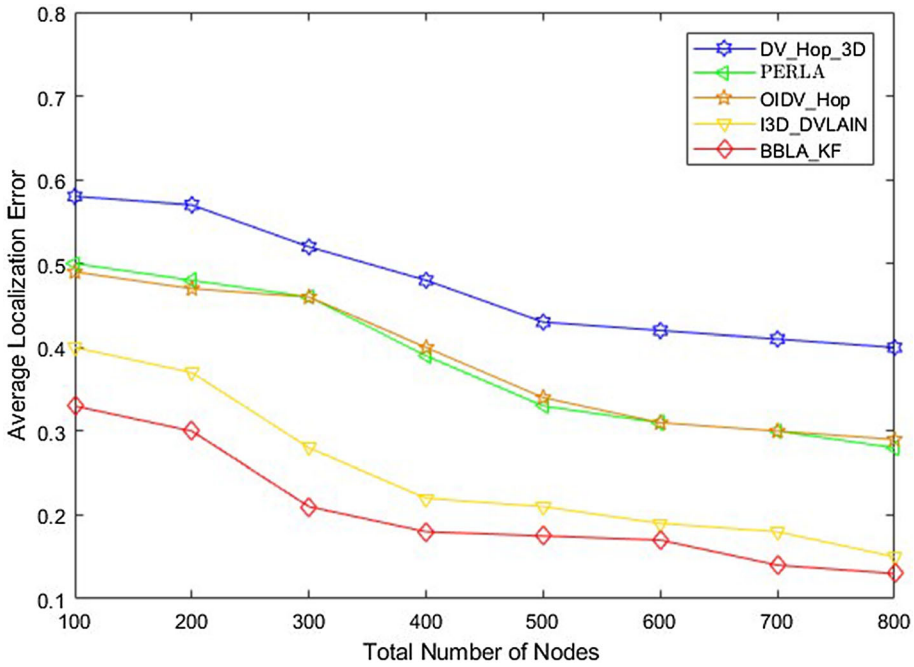


Fig. 11 Average localization error versus the total number of sensor nodes

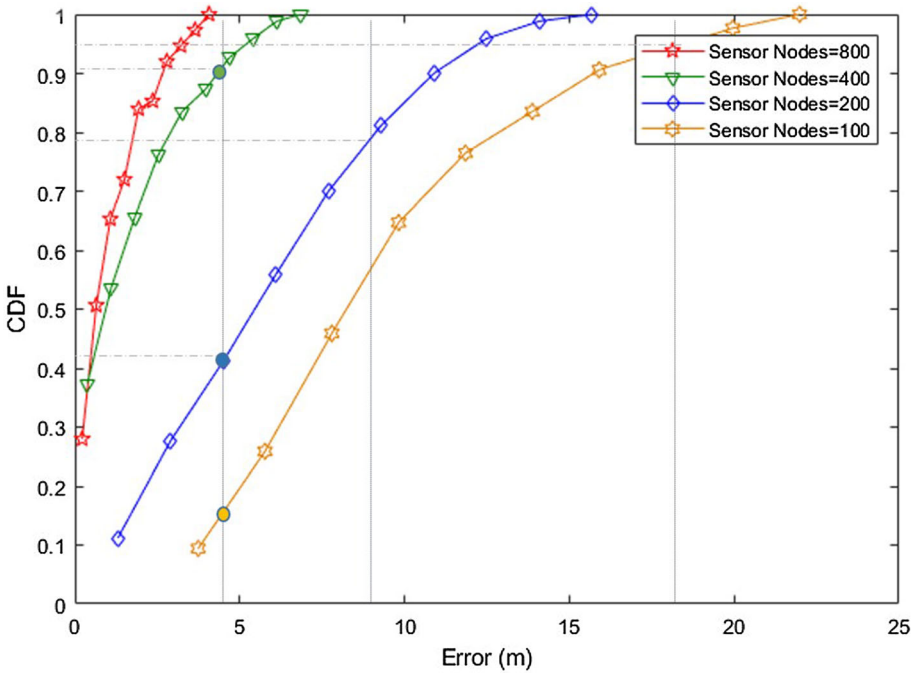


Fig. 12 Cumulative distribution function (CDF) plots of different number of sensor nodes

Table 3 Localization error CDF's for different number of nodes

	Localization erorr		
Number of nodes	0.25R	0.5R	R
800	100%	100%	100%
400	91%	100%	100%
200	42%	78%	100%
100	14%	57%	95%

on these results, we are able to confirm that our algorithm performs better than other algorithms.

5.6 Effect of anisotropy

BBLA-KF is evaluated in 3D fields. The impact of anisotropy is evaluated by studying the localization results in anisotropic fields with isotropic fields. Thus, The results of localization of 3D cubic field and 3D C-shaped field are compared for our proposed algorithm. Using the MATLAB simulation experiment, we obtain the experimental results in Figs. 15 and 16, where the red lines represent the location error, the blue circles are the reel positions of the unknown nodes and the red circles are the estimated positions of the unknown nodes by our algorithm. Figure 16, the C-shaped anisotropic field and Fig. 15 the isotropic field give a good ALE which means a good localization accuracy. The BBLA-KF shows good results in both isotropy and anisotropic fields, as mentioned in Table 4, which proves the efficiency of our algorithm.

6 Conclusion

The precision of localization algorithms is a problem, especially in irregular areas. The majority of existing algorithms in the literature have been developed for locating nodes in regular fields, whereas the real-world applications of WSNs are in irregular fields. In order to help overcome this gap, we proposed a localization algorithm called BBLA-KF. This hop-based algorithm uses properties of the Bounding-Box and Kalman-Filter methods to enhance localization accuracy in a variety of field types. The results of this algorithm show better localization accuracy in various 3D fields of different shapes compared to its counterparts recently published. In future work, we are planning to implement this algorithm on an actual WSN to solve the localization problem. Also, We are planning to

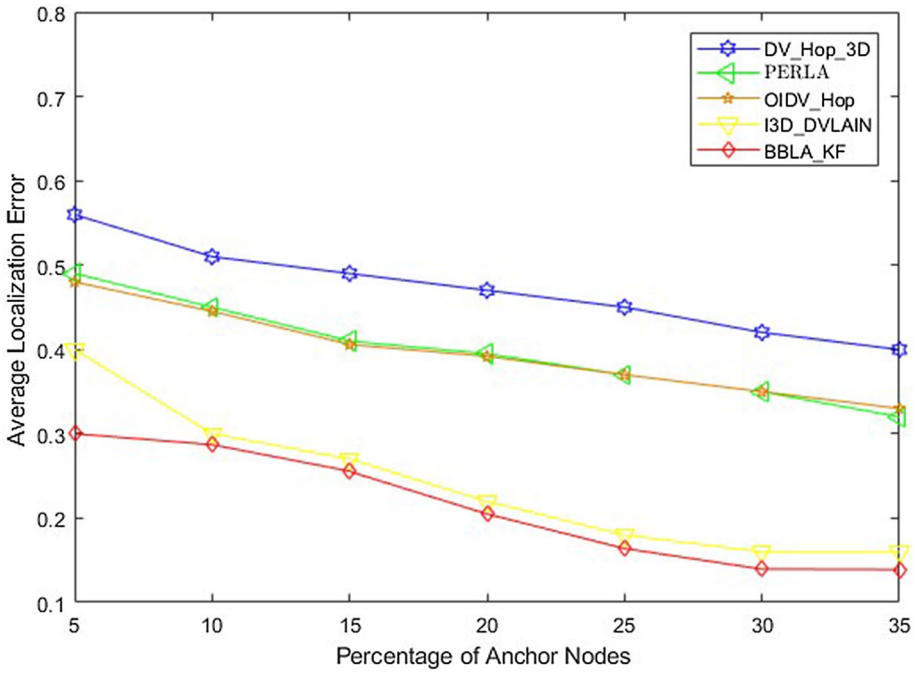


Fig. 13 Localization error versus percentage of anchor nodes

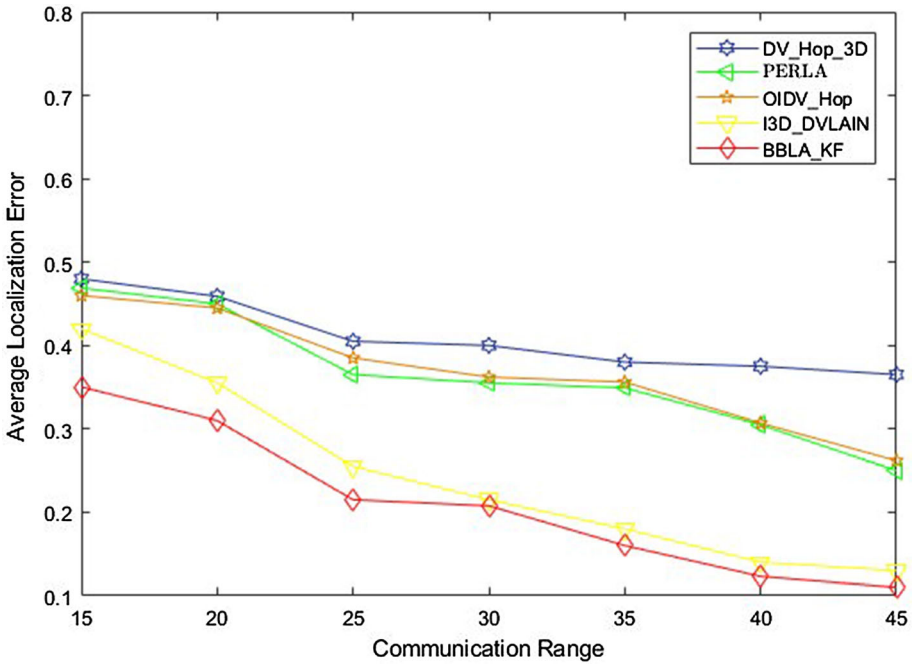


Fig. 14 Localization error versus percentage of communication range

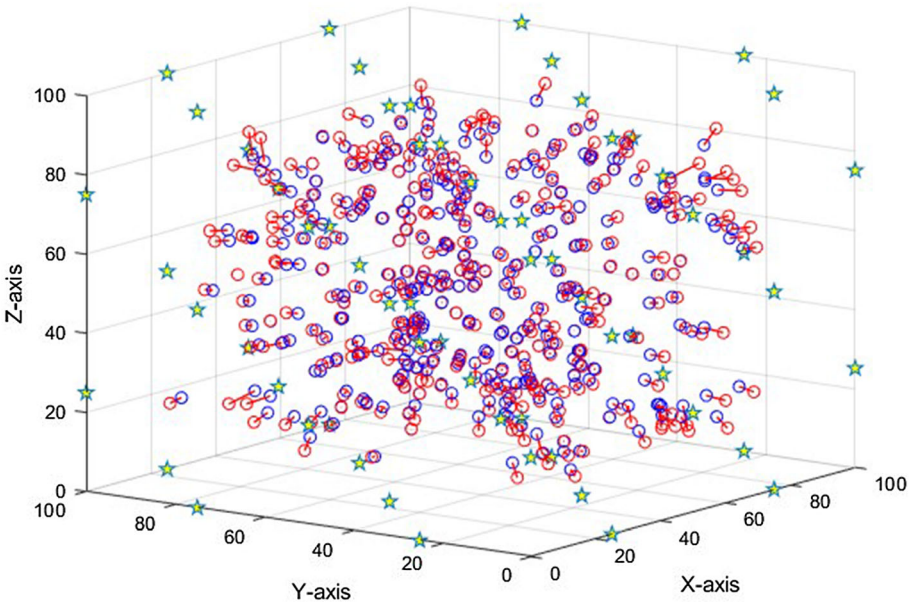


Fig. 15 Location results for unknown nodes in 3D cubic field

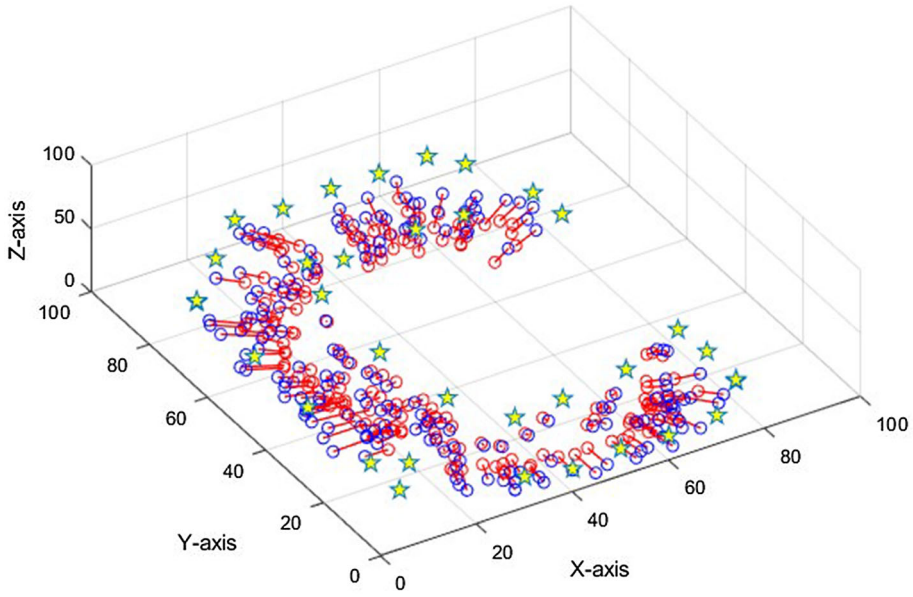


Fig. 16 Location results for unknown nodes in 3D C-shaped field

Table 4 ALE results for isotropic and anisotropic 3D fields

3D-WSN fields	Average localization error
Isotropy	0.21
Anisotropy	0.26

design and develop our model on a configurable field programmable gate array (FPGA).

Author Contributions RM made the concept and design of the article, collected the data, performed the experiments and conducted the analysis. NL, ARN and JLSR reviewed and corrected the design, writing and experiments. All authors have read and agreed to the published version of the manuscript.

Funding Funding for open access publishing: Universidad de Sevilla/CBUA. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability The dataset is available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no confict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

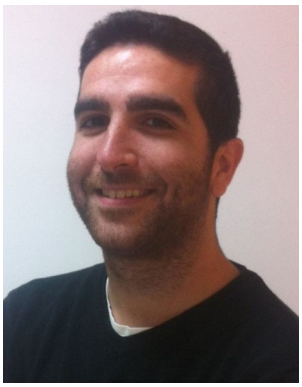
1. Liu, X., Sun, C., Zhou, M., Wu, C., Peng, B., & Li, P. (2021). reinforcement learning-based multislot double-threshold spectrum sensing with Bayesian fusion for industrial big spectrum data. *IEEE Transactions on Industrial Informatics, 17*(5), 3391–3400.
2. Liu, X., Sun, Q., Lu, W., Wu, C., & Ding, H. (2020). Big-data-based intelligent spectrum sensing for heterogeneous spectrum communications in 5G. *IEEE Wireless Communications, 27*(5), 67–73.
3. Liu, X., Ding, H., & Hu, S. (2021). Uplink resource allocation for NOMA-based hybrid spectrum access in 6G-enabled cognitive Internet of Things. *IEEE Internet of Things Journal, 8*(20), 15049–15058.
4. Bhat, S. J., & Santhosh, K. V. (2020). Is localization of wireless sensor networks in irregular fields a challenge? *Wireless Personal Communications, 114*, 2017–2042.
5. Fan, J., Hu, Y., Luan, T. H., & Dong, M. (2017). DisLoc: A convex partitioning based approach for distributed 3-D localization in wireless sensor networks. *IEEE Sensors Journal, 17*(24), 8412–8423.
6. Kumar, S., Kumar, S., & Batra, N. (2020). Optimized distance range-free localization algorithm for WSN. *Wireless Personal Communications, 117*, 1879–1907.
7. Wang, Z., Zhang, B., Wang, X., Jin, X., & Bai, Y. (2018). Improvements of multihop localization algorithm for wireless sensor networks. *IEEE Systems Journal, 13*, 1–12.
8. Singha, A. K., Singla, A., & Gupta, S. (2016). Localization system for wireless sensor networks. *IJRDO Journal of Computer Science and Engineering, 2*, 7.
9. Hu, L., & Evans, D. (2004). Localization for mobile sensor networks. In *MobiCom 04*.

10. Hu, Z., Gu, D., Song, Z., & Li, H. (2008). Localization in wireless sensor networks using a mobile anchor node. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (pp. 602–607).
11. Han, G., Zhang, C., Lloret, J., Shu, L., & Rodrigues, J. J. P. C. (2014). A mobile anchor assisted localization algorithm based on regular hexagon in wireless sensor networks. *The Scientific World Journal, 2014*, 219371.
12. Han, G., Xu, H., Jiang, J., Shu, L., Hara, T., & Nishio, S. (2013). Path planning using a mobile anchor node based on trilateration. *Wireless Communications and Mobile Computing, 13*, 1324–1336.
13. Wang, Z., Zhang, B., Wang, X., Chai, S., & Bai, Y. (2018). Cooperative localization with bounding constraints in mobile wireless sensor networks. *IEEE Access, 6*, 47011–47025.
14. Baggio, A., & Langendoen, K. (2008). Monte Carlo localization for mobile wireless sensor networks. *Ad Hoc Networks, 6*(5), 718–733.
15. Qiao, Y., Chunyue, Z., Baitong, Z., & Hui, T. (2019) An improved MCB localization algorithm based on received signal strength indicator. In *20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)* (pp. 421–426).
16. Zhou, C., Tian, H., & Zhong, B. (2020). An improved MCB localization algorithm based on weighted RSSI and motion prediction. *Computer Science and Information Systems, 17*, 779–794.
17. Hua, W., Liu, J., Dong, Z, & Liu, Y. (2020). A hybrid mobile node localization algorithm based on adaptive MCB-PSO approach in wireless sensor networks. *Wireless Communications and Mobile Computing*.
18. Ullah, I., Shen, Y., Su, X., Esposito, C., & Choi, C. (2020). A localization based on unscented Kalman filter and particle filter localization algorithms. *IEEE Access, 8*, 2233–2246.
19. Bhat, S. J., & Santhosh, K. V. (2022). Localization of isotropic and anisotropic wireless sensor networks in 2D and 3D fields. *Telecommunication Systems, 79*, 309–321.
20. Shen, S., Yang, B., Qian, K., She, Y., & Wang, W. (2019). On improved DV-hop localization algorithm for accurate node localization in wireless sensor networks. *Chinese Journal of Electronics, 28*, 658–666.
21. Kumar, S., & Lobiyal, D. K. (2014). Power efficient range-free localization algorithm for wireless sensor networks. *Wireless Networks, 20*, 681–694.
22. Kaushik, A., Lobiyal, D. K., & Kumar, S. (2021). Improved 3-dimensional DV-hop localization algorithm based on information of nearby nodes. *Wireless Networks, 27*, 1801–1819.
23. Niculescu, D., & Nath, B. (2001) Ad hoc positioning system (APS). In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)* (Vol. 5, pp. 2926–2931).
24. Mani, R., Sevillano, J. L., & Liouane, N. (2022). Improved least-square DV-Hop algorithm for localization in large scale wireless sensor network. In *2022 International Conference on Control, Automation and Diagnosis (ICCAD)*.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Rahma Mani received the Diploma degree in electronical engineering from the National Engineering School of Monastir (ENIM), Tunisia, in 2018. Currently Ph.D. student at the National Engineering School of Monastir Tunisia (ENIM), University of Monastir, Tunisia and the Escuela Politécnica Superior, University of Seville, Spain. Her research interest includes localization algorithms, optimization in wireless sensor networks and embedded and systems design.



Antonio Rios-Navarro received the B.S. degree in computer science engineering, the M.S. degree in computer engineering, and the Ph.D. degree in neuromorphic engineering from the University of Seville, Seville, Spain, in 2010, 2011, and 2017, respectively. He currently holds the assistant professor position at the Computer Architecture and Technology Department, University of Seville. His current research interests include neuromorphic computing, Artificial Intelligence, real-time spikes signal processing, field-programmable gate array design, and deep learning.



Jose-Luis Sevillano-Ramos is Professor of Computer Technology and Architecture at Universidad de Sevilla (Spain). In the period 2014–2022 he has been Director of the School of Computer Engineering at the Universidad de Sevilla, and since February 2015 he has been President of the Board of Trustees of the Foundation for Research and Development of Information Technologies in Andalusia (FIDETIA). His research interests include real-time communications and architectures, Mobile Robots and eHealth and Rehabilitation Systems.



Noureddine Liouane is professor in the Engineering Department at the ENIM-Monastir. He received a MSEE degree in 1988 from the ENSET at Tunis University and the Ph.D. degree from Ecole Centrale de Lille, France in 1998. His research interests include evolutionary optimization methods for events systems, computer science and operational research.