

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)



---

# Gradient Descent Localization in Wireless Sensor Networks

---

Nuha A.S. Alwan and Zahir M. Hussain

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.69949>

---

## Abstract

Meaningful information sharing between the sensors of a wireless sensor network (WSN) necessitates node localization, especially if the information to be shared is the location itself, such as in warehousing and information logistics. Trilateration and multilateration positioning methods can be employed in two-dimensional and three-dimensional space respectively. These methods use distance measurements and analytically estimate the target location; they suffer from decreased accuracy and computational complexity especially in the three-dimensional case. Iterative optimization methods, such as gradient descent (GD), offer an attractive alternative and enable moving target tracking as well. This chapter focuses on positioning in three dimensions using time-of-arrival (TOA) distance measurements between the target and a number of anchor nodes. For centralized localization, a GD-based algorithm is presented for localization of moving sensors in a WSN. Our proposed algorithm is based on systematically replacing anchor nodes to avoid local minima positions which result from the moving target deviating from the convex hull of the anchors. We also propose a GD-based distributed algorithm to localize a fixed target by allowing gossip between anchor nodes. Promising results are obtained in the presence of noise and link failures compared to centralized localization. Convergence factor issues are discussed, and future work is outlined.

**Keywords:** gradient descent, node localization, tracking, distributed averaging, push-sum algorithm, link failure, step size

---

## 1. Introduction

Wireless sensor networks are used in a wide range of monitoring and control applications such as traffic monitoring, environmental monitoring of air, water, soil quality or temperature, smart factory instrumentation, and intelligent transportation. The nodes are usually small

radio-equipped low-power sensors scattered over an area or volume of a few tens of square or cubic meters, respectively. There is information sharing between sensors and for this information to be meaningful, the nodes or sensors have to be located. Although global positioning systems (GPS) achieve powerful localization, it is costly and impractical to equip each sensor in a WSN with a GPS device. Besides, in many environments such as indoors and forested zones, the GPS signal may be weak or even unavailable. This explains the vast on-going research devoted to efficient localization for WSNs.

Node information may be processed either centrally or in a distributed manner. In centralized localization, distance measurements are collected by a central processor prior to calculation. In distributed algorithms, the sensors share their information only with neighbors but possibly iteratively. Both methods face the high cost of communication, but, in general, centralized localization produces more accurate location information, whereas distributed localization offers more scalability and robustness to link failures.

Node localization relies on the measurements of distances between the nodes to be localized and a number of reference or anchor nodes. The distance measurements can be via radio frequency (RF), acoustic, or ultra-wideband (UWB) signals. Measurements that indicate distance can be time of arrival (TOA), angle of arrival (AOA), or received signal strength (RSS). TOA measurements seem to be most useful especially in low-density networks, since they are not as sensitive to inter-device distances as AOA or RSS. The TOA distance measurements usually correspond to line-of-sight (LOS) arrivals that are hampered by additive noise. The consequent measurement errors can be adequately modeled by zero-mean Gaussian noise with variance  $\sigma^2$ . The inclusion of a mean  $\mu$  in this Gaussian model may be necessary to account for possible non-line-of-sight (NLOS) arrivals.

Accurate location information is important in almost all real-world applications of WSNs. In particular, localization in a three-dimensional (3D) space is necessary as it yields more accurate results. Trilateration and multilateration positioning methods [1] are analytical methods employed in two-dimensional (2D) and three-dimensional (3D) spaces, respectively. These methods use distance measurements to estimate the target location analytically, and suffer from poor performance, decreased accuracy, and computational complexity especially in the 3D case. More specifically, trilateration is the estimation of node location through distance measurements from three reference nodes such that the intersection of three circles is computed, thereby locating the node as shown in **Figure 1**. Multilateration is concerned with localization in a 3D space in which more than three reference nodes are used [2].

Practically, when distance measurements are noisy and fluctuating, localization becomes difficult. The intersection point in **Figure 1** becomes an overlapped region. With this uncertainty, analytical methods become almost useless and we resort to optimization methods. Iterative optimization methods offer an attractive alternative solution to this problem. The Kalman filter, which is an iterative state estimator, can be used for node localization in case of noisy measurements. However, its computational and memory requirements may not be met adequately by the limited resources of a sensor system, subsequently resulting in poor

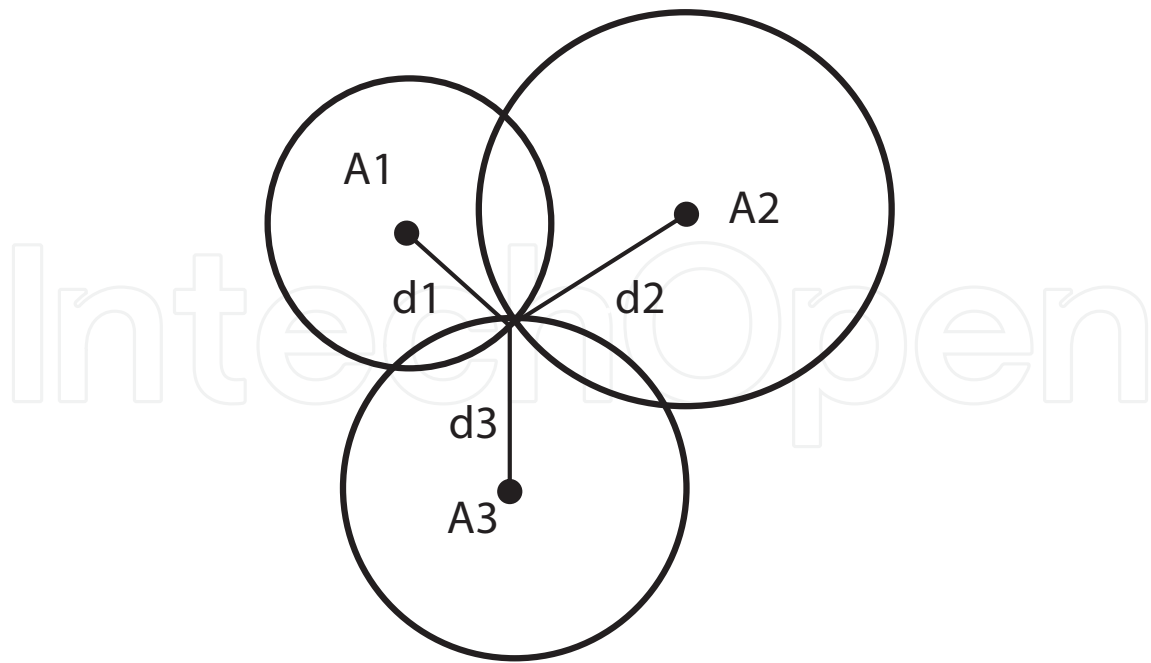


Figure 1. Trilateration.

performance [3]. Thus, the most common iterative optimization method is the computationally efficient gradient descent algorithm, which has been widely dealt with in the literature for the 2D case [4, 5].

This chapter addresses localization in a three-dimensional space of stationary and moving wireless sensor network nodes by gradient descent methods. First, it is assumed that a central processor collects the data from the nodes. TOA measurements will be assumed throughout. An evaluation analysis of the performance of the localization algorithm considered is performed. The effect of measurement noise has also been studied. The work also investigates tracking of moving sensors and proposes a method to counteract some associated problems such as falling into local minima [6]. Second, distributed GD localization will be handled using a proposed gossip-based technique in which anchor nodes exchange data to iteratively compute the positions and gradients locally in each anchor [7]. This distributed method serves to mitigate the effects of noise and link failures.

## 2. Centralized gradient descent (GD) localization in 3D wireless sensor networks

### 2.1. Stationary node localization

Localization in 3D space is particularly important in practical applications of WSNs, but many of its aspects remain unexplored as the typical scenario for WSN localization is set up in a 2D plane [8]. In a 3D space, at least four anchor nodes are needed whose locations are known. An

estimate of the  $i$ th distance  $d_i$ ,  $i = 1, 2, 3, 4$ , between the  $i$ th anchor node  $(x_i, y_i, z_i)$  and the node to be localized  $(x, y, z)$  is needed.

The TOA distance measurement technique is assumed. TOA is the time delay between transmission at the node to be localized and reception at an anchor node. This is equal to the distance  $d_i$  divided by the speed of light if either RF or UWB signals are used. The backbone of the TOA distance measurement technique is the accuracy of the arrival time estimates. This accuracy is hampered by additive noise and NLOS arrivals. The measurement errors are modeled as additive zero-mean Gaussian noise. The total additive Gaussian measurement noise will be modeled as  $N(\mu, \sigma_{\text{NLOS}}^2)$ , where the letter  $N$  denotes the normal or Gaussian distribution,  $\mu$  is the mean, and  $\sigma_{\text{NLOS}}^2$  is the variance, taking into account NLOS as well as LOS arrivals. The occasional inclusion of a mean accounts for the biased location estimate resulting from NLOS errors [9, 10].

To determine the TOA in asynchronous WSNs, two-way TOA measurements are used. In this method, one sensor sends a signal to another that immediately replies. The first sensor will then determine TOA as the delay between its transmission and reception divided by two [10].

Gradient descent iterative optimization in three dimensions results in slower convergence when compared to the 2D case due to tracking along an extra dimension. This is true for all iterative optimization methods. Due to limited exploration of 3D scenarios in the literature, the present work presents practical results relating to the GD localization problem in three-dimensional WSNs. The definition of an objective or error function is normally required for optimization methods whose purpose is to minimize this function to produce the optimal solution. In GD localization, the objective error function is usually defined as the sum of squared distance errors from all anchor nodes. As such, we may write the objective error function as:

$$f(p) = \sum_{i=1}^N \{ [(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2]^{1/2} - d_i \}^2 \quad (1)$$

and

$$d_i = c(t_i - t_o) \quad (2)$$

where  $p = [x, y, z]^T$  is the vector of unknown position coordinates  $(x, y, z)$ ,  $t_i$  is the receive time of the  $i$ th anchor node,  $t_o$  is the transmit time of the node to be localized,  $c$  is the speed of light ( $= 3 \times 10^8$  m/s), and  $N$  is the number of anchor nodes. The difference  $(t_i - t_o)$  is the TOA that can be measured (with measurement noise) in asynchronous WSNs as explained.

Minimization of the objective function produces the optimal solution that is the position estimate of the node to be localized. This problem is solved iteratively using GD as follows:

$$p_{k+1} = p_k - \alpha \cdot g_k \quad (3)$$

where  $p_k$  is the vector of the estimated position coordinates,  $\alpha$  is the step size, and  $g_k$  is the gradient of the objective function given by:

$$g_k = \nabla f(x, y, z) = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right]^T \quad (4)$$

If we define the term  $B_{k,i}$  as:

$$B_{k,i} = [(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2]^{\frac{1}{2}} \quad (5)$$

then the three components of the gradient vector at the  $k$ th iteration will be:

$$\left. \frac{\partial f}{\partial x} \right|_k = \sum_{i=1}^N 2\{B_{k,i} - d_i\} \cdot \frac{(x_k - x_i)}{B_{k,i}} \quad (6)$$

$$\left. \frac{\partial f}{\partial y} \right|_k = \sum_{i=1}^N 2\{B_{k,i} - d_i\} \cdot \frac{(y_k - y_i)}{B_{k,i}} \quad (7)$$

$$\left. \frac{\partial f}{\partial z} \right|_k = \sum_{i=1}^N 2\{B_{k,i} - d_i\} \cdot \frac{(z_k - z_i)}{B_{k,i}} \quad (8)$$

The initial position coordinates may be chosen to be the mean position of all anchor nodes. The required number of iterations for convergence is a tradeoff between energy consumption, which is critical to WSNs, and the degree of accuracy.

A minimum of four anchor nodes are needed to estimate position in a 3D space. The estimation accuracy increases as a function of the number of anchor nodes. Since the objective function is the sum of the squares of the differences between estimated distances and measured distances, distance measurement errors are squared, too. This problem is countered by weighting distance measurements according to their confidence to limit the effect of measurement errors on localization results [11]. So the objective function accommodating different weights may be expressed as:

$$f(p) = \sum_{i=1}^N w_i \{ [(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2]^{1/2} - d_i \}^2 \quad (9)$$

Weighting, however, may result in sub-optimal solutions if only four anchor nodes are used. Since usually there are only a few anchors in a real WSN [12], the use of five anchor nodes is a good choice to achieve better accuracy without undue deviation from real settings.

In a 3D WSN, the error function of Eq. (1) is a 4D performance surface with a global minimum and several local minima. To avoid local minima, the gradient descent must run several times with different starting points, which is expensive computationally. To better visualize the local minima problem, localization in a 2D space is considered to enable performance surface plotting in a 3D space. Three anchors (30, 45), (80, 65), and (10, 80) are chosen with  $d_i = 32.0156, 83.2166,$  and  $60.0000$  corresponding to a point  $p = (10, 20)$ . Then, plotting the following objective function

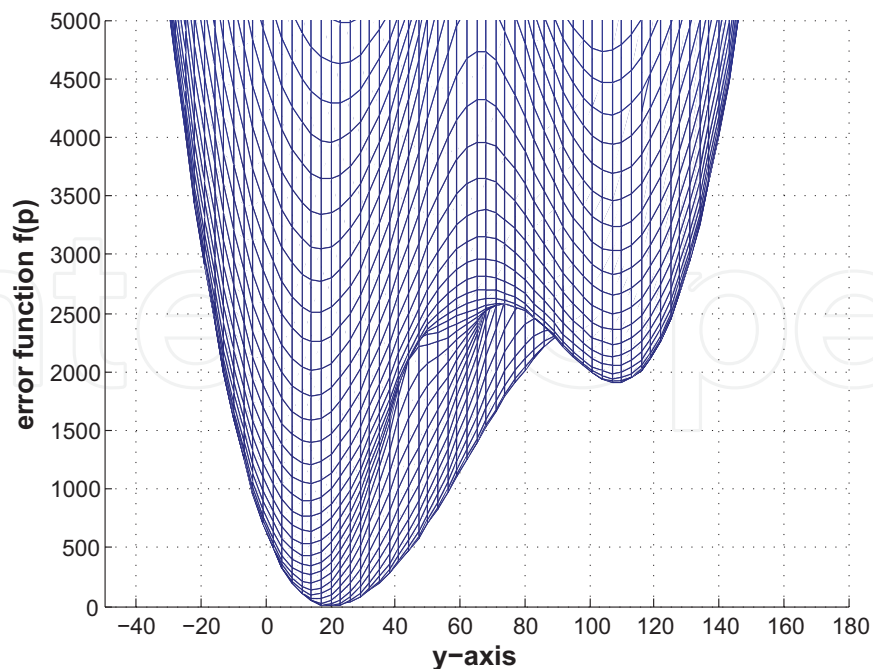
$$f(p) = \sum_{i=1}^3 \{[(x - x_i)^2 + (y - y_i)^2]^{1/2} - d_i\}^2 \quad (10)$$

results in **Figure 2** with azimuth = 90° and elevation = 0°.

The presence of a global minimum at  $p$  and a neighboring local minimum can be discerned from **Figure 2**. Therefore, GD search of the minimum along the performance surface often gets trapped in a local minimum especially when tracking a moving node. In the following section, a solution will be presented to solve the local minima problem in a moving sensor localization setting.

### 2.1.1. Simulation scenario

GD localization in a 3D WSN is simulated in MATLAB. The anchor node locations are chosen at random in a volume of  $200 \times 200 \times 200 \text{ m}^3$ . It is assumed that the target node to be localized (whether stationary or moving) has all anchor nodes within its radio range, and that the target node lies within the convex hull of the anchors. The LOS and NLOS measurement noise is assumed to obey a normal distribution  $N(\mu, \sigma^2)$ . In subsequent simulations, noisy TOA measurements are simulated by adding a random component to the exact value of the time measurement. The latter is readily computed for simulation purposes from knowledge of the exact node position to be localized, the anchor positions, and the speed of light  $c$ .



**Figure 2.** Error function  $f(p)$  as a 3D performance surface with 2D anchor nodes (30, 45), (80, 65), and (10, 80) and a global minimum at  $p = (10, 20)$ . Azimuth = 90° and elevation = 0°.

### 2.1.2. Simulation results

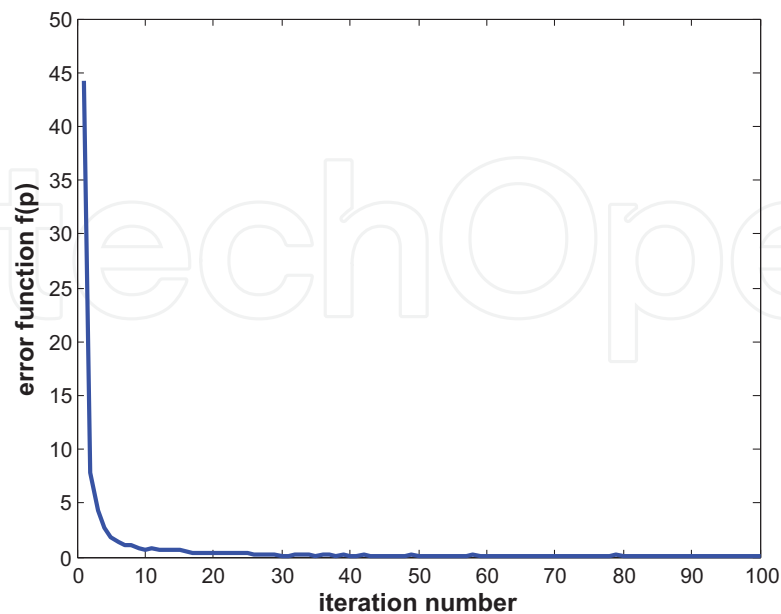
We first consider four anchor nodes to localize a node of position (60, 90, 60) in the 3D space assuming that the standard deviation (SD) of the zero-mean Gaussian TOA measurement noise, the convergence factor or step size, and the number of iterations to be  $SD = 0.001 \mu\text{s}$ ,  $\alpha = 0.25$ , and  $j = 100$ , respectively. The anchor positions are (10, 100, 10), (100, 90, 10), (10, 70, 100), and (100, 80, 100). Simulation results localized the target node as (60.28, 84.02, 58.65). When five anchor nodes are used, they provide an almost ideal target localization of (60.16, 89.64, 60.09). The fifth anchor position is (90, 90, 150). **Figure 3** is a plot of the error function versus the number of iterations for this last case of five anchor nodes. Retaining this scenario, another node (70, 45, 60) is localized as (70.03, 45.16, 59.85). Obviously, any node within the convex hull of the anchor nodes will be almost exactly localized with five anchors.

The results of **Figure 3** are repeated in **Figure 4** taking into account the presence of NLOS arrivals and a greater noise standard deviation. In **Figure 4**,  $SD = 0.002 \mu\text{s}$ , and  $\mu_{\text{NLOS}} = 0.006 \mu\text{s}$ . A reduction in the localization process accuracy is readily noticed: The point (60, 90, 60) results in a localization of (60.35, 88.97, 59.40). It is also clear from the figure that the solution is biased due to NLOS arrivals.

The issue of energy consumption may appear to disfavor iterative methods compared to analytical methods. This is not the case, however, when the target is moving, since updating would then be must whether iterative or other methods are employed.

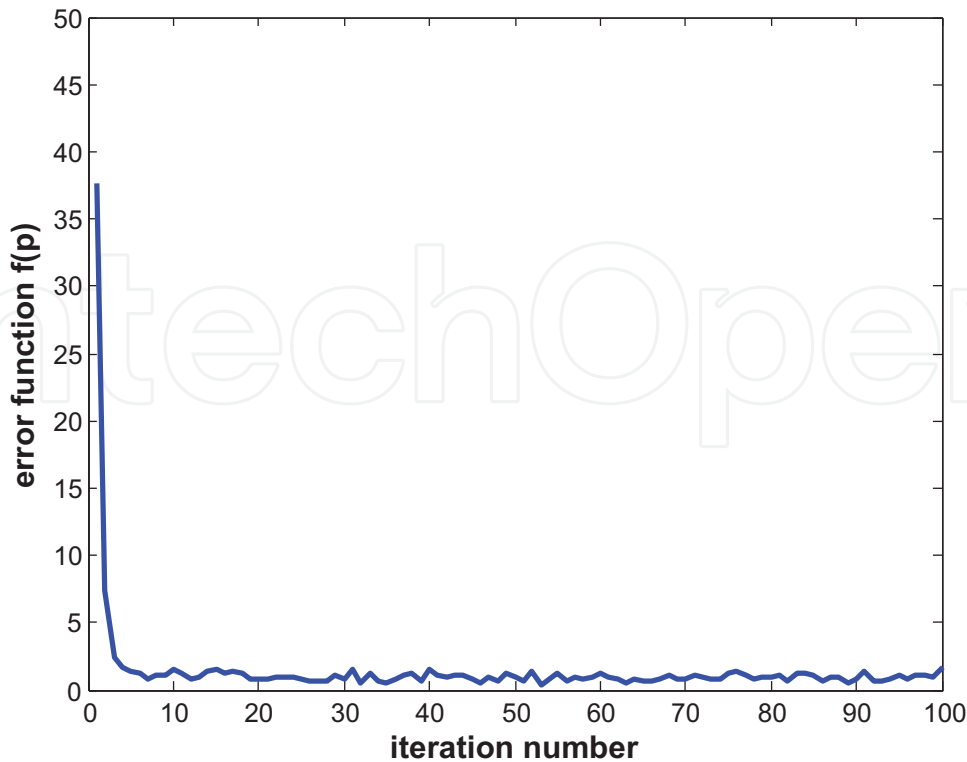
## 2.2. Moving node localization and tracking

GD can be used to track a moving target in real time. The measurement sample interval determines the measurement update rate. A bit of care is required in adjusting the sample



**Figure 3.** Error function versus the number of iterations when GD localization of a stationary target in 3D space is performed using five anchor nodes. Convergence factor = 0.25, TOA measurement noise  $SD = 0.001 \mu\text{s}$ .





**Figure 4.** Error function versus the number of iterations when GD localization of a stationary target in 3D space is performed using five anchor nodes. Convergence factor = 0.25, TOA measurement noise SD = 0.002  $\mu\text{s}$  and  $\mu_{\text{NLOS}} = 0.006 \mu\text{s}$ .

interval to avoid conflict with moving sensor velocity and motion models which may be completely unknown [9]. The moving node must provide multiple measurements to the anchors as it moves across space. It has the opportunity to reduce environment-dependent errors as it averages over space. Many computational aspects of this problem remain to be explored [10].

In Refs. [13, 14], the problem of avoiding local minima for moving sensor localization is handled by smart use of available anchors and good initialization. Although these works are also based on minimizing cost functions, they are not general GD algorithms. Moreover, these works require good initial estimation of the target location. It is therefore worthwhile to attempt achieving moving sensor localization without the need to estimate the initial moving target location. As a solution to this problem, we introduce the concept of diversity in the iterative GD localization problem.

The algorithm below is proposed in Ref. [6] to localize a moving sensor in a 3D space with the provision of local minima avoidance. The foreseen success of the proposed method is based on the idea that, as the updated position begins to wander away from the global minimum in the direction of a local minimum, it is highly probable that it will return to the right track if some anchor nodes are replaced. Anchor node replacement results in a consequent change in the performance surface shape and hence local minima positions.

Algorithm 1: Proposed GD localization of a moving sensor [6]

1. Estimate a suitable measurement sample interval or update rate.
2. Cluster available anchor nodes into sets of five nodes each. The number of resulting sets  $P$  will be:

$$P = \binom{N}{5} = \frac{N!}{5!(N-5)!}$$

where  $N$  is the total number of heard anchor nodes.

3. Randomly draw  $M$  sets from  $P$  obeying a uniform distribution.
4. Perform  $M$  independent gradient descent localization procedures on the moving sensor using these  $M$  sets.
5. Iterate the gradient descent algorithm up to the  $L$ -th update, and calculate the final  $f(p)$  for each of the  $M$  sets. Discard the sets that produce  $f(p)$  greater than a certain threshold  $\gamma$ . Find the point  $p$  with the minimum  $f(p)$ .
6. Stop the algorithm if the moving sensor tracking halts.
7. Complete the  $M$  sets by randomly choosing other sets from  $P$ , and repeat steps 4–6 starting with the final position of  $p$  that corresponds to the minimum  $f(p)$ .

The different parameters appearing in Algorithm 1 should be properly chosen. These are  $M$ ,  $N$ ,  $L$ , and the threshold  $\gamma$ . As discussed in the problem description,  $N$  should not be unduly large in practical settings. Assuming that five anchors per set are involved in localization,  $N$  must not be much greater especially when the WSN area or volume is limited. As for  $M$ , it naturally determines the computational overhead; GD localization must run  $M$  times in each round of position estimation. To reduce the amount of computation to a minimum, the choice of  $M$  must achieve a tradeoff between computational complexity and sufficient diversity of anchor sets in order to cancel unsuitable candidates and retain functional ones. The threshold  $\gamma$  depends on the specific application and how tolerant the latter is to the final value of the error function  $f(p)$ . In the simulations, the moderate value of  $7 \text{ m}^2$  is used as a default setting. This means that the estimated squared distance error associated with each anchor is  $(7/5) \text{ m}^2$  on average according to Eq. (1).

As for  $L$ , it has been assigned the value 150 iterations in the present simulation settings, which is, however, an ad hoc value that worked for the particular settings under consideration. To ensure accurate tracking, a check on the error function of all running estimations can be performed after each certain interval (e.g. 30 iterations) and then the decision is made whether to proceed or replace the diverging sets.

Applying an iterative optimization algorithm for  $M$  subsets of heard anchors, when  $M$  can be as large as 20, has been implemented in Ref. [12], albeit without diversity, in the context of least median square (LMS) secure node localization in WSNs to combat localization attacks. Algorithm 1 has been inspired from Ref. [12] by adapting it to:

1. Suit the simpler iterative GD localization algorithm with the aim of local minima avoidance rather than secure localization.

2. Repeat itself with diversity to avoid divergence due to local minima as the target moves along its path.

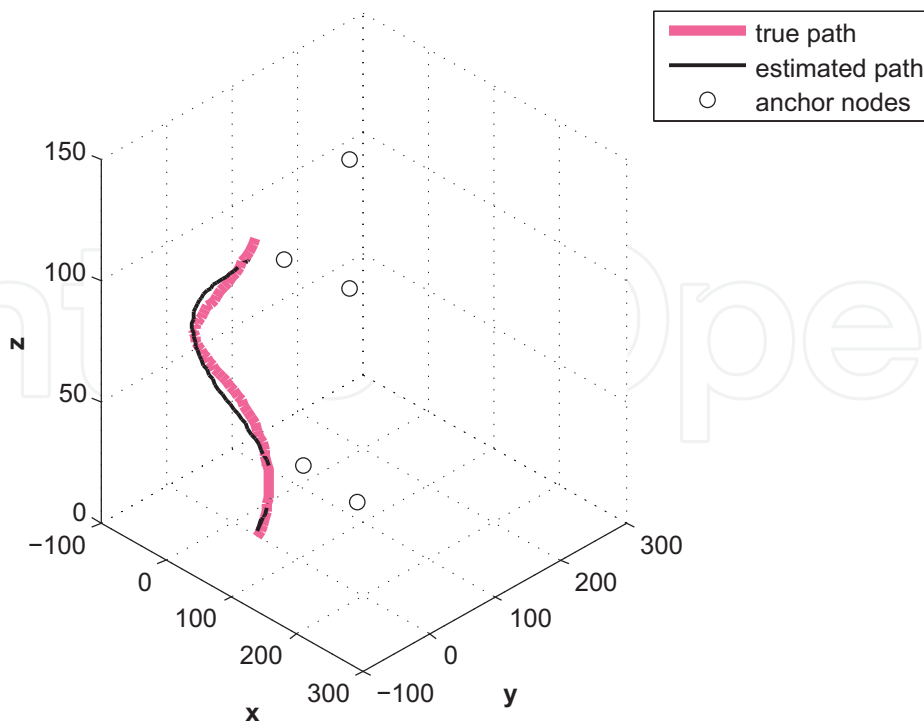
A final remark concerns the communication overhead; the proposed algorithm does not add to the communication complexity. With each iteration, and after the sensing has been achieved, only one broadcast (communication) of the distance measurement is enough from each of the  $N$  anchors. It is in the fusion center that the various combinations of  $P$  are sorted out and their associated computations performed.

### 2.2.1. Simulation results

In the following scenarios, a moving node is tracked and localized. We assume five anchor nodes since this offers the best estimation accuracy. We first illustrate GD tracking of a node moving along a helical path (**Figure 5**). The three dimensions representing the moving target location are given by:

$$\begin{aligned} x &= r \cos \theta \\ y &= r \sin \theta \\ z &= k \theta \end{aligned} \quad (11)$$

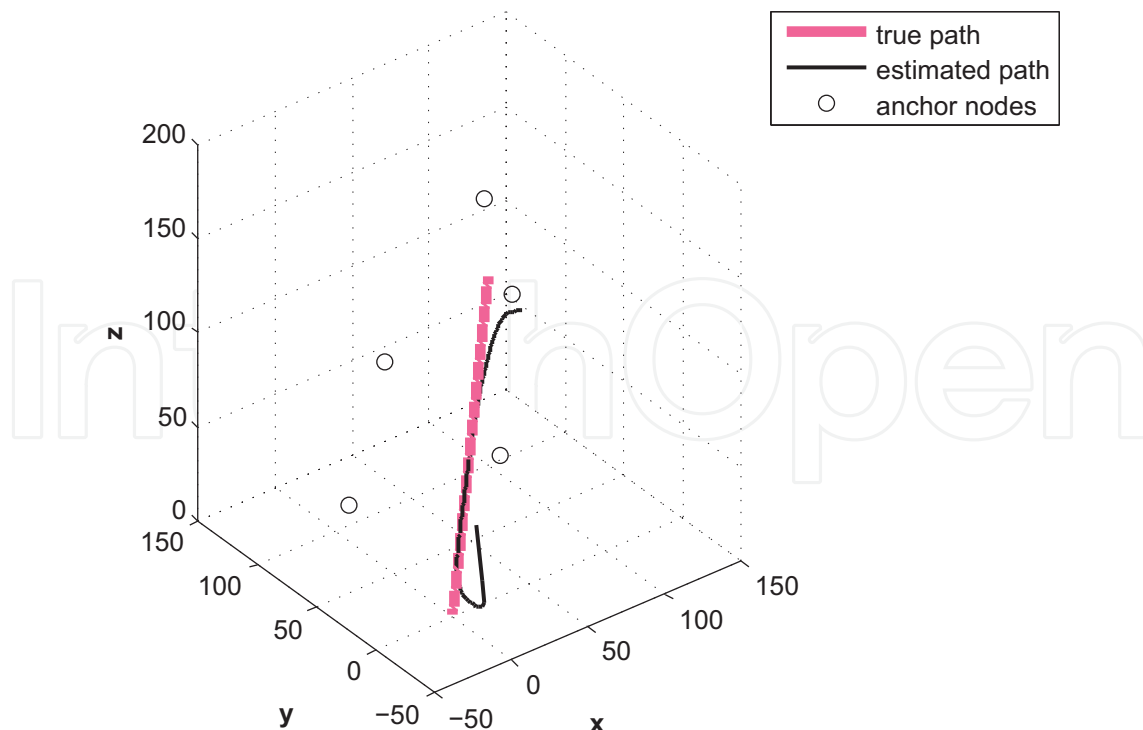
The angle  $\theta$  is continuously increasing and  $r$  and  $k$  are constants. **Figure 5** shows the moving node helical path and its GD track for values of  $\theta$  varying from zero to  $2\pi$ , together with the anchor positions shown as small circles. The anchors are assumed to be in the radio range of the helical trajectory. The constant values  $r$  and  $k$  are 40 and 20, respectively. Noise-free distance measurements are assumed throughout.



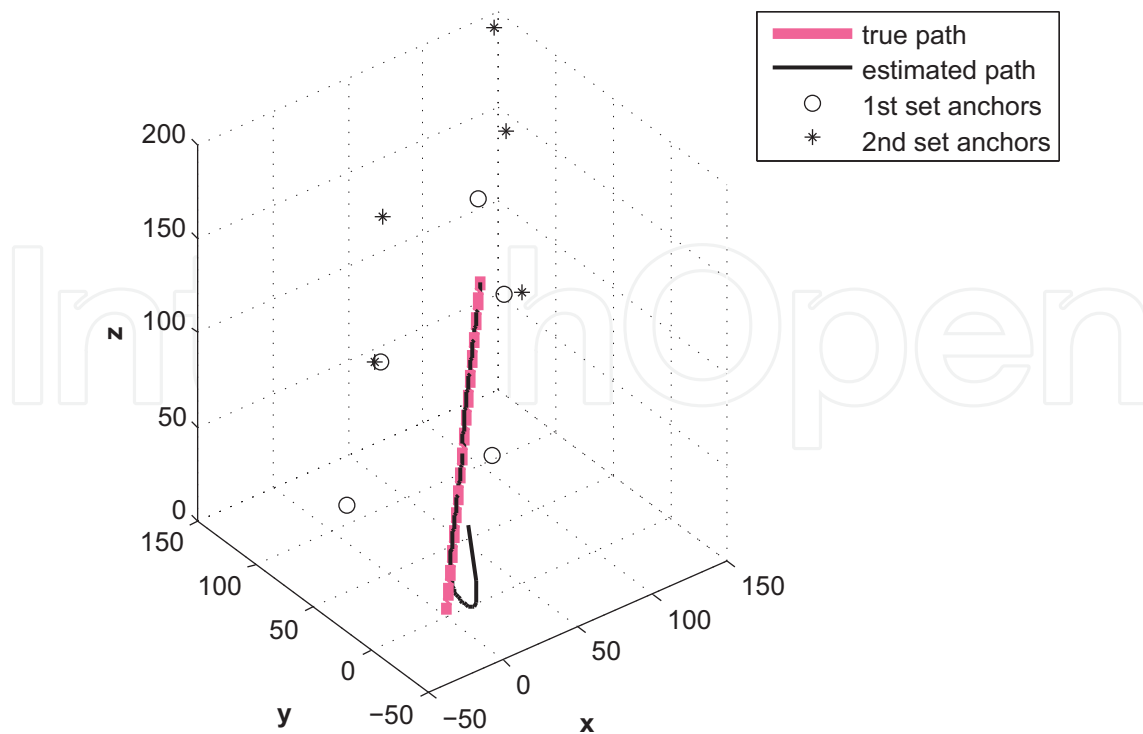
**Figure 5.** Target node tracking along a helical path.

Next, and to better illustrate the proposed algorithm in Ref. [6] for moving target tracking, and the effect of the various inherent parameter values, a straight line path segment is considered. The details are outlined in the following steps:

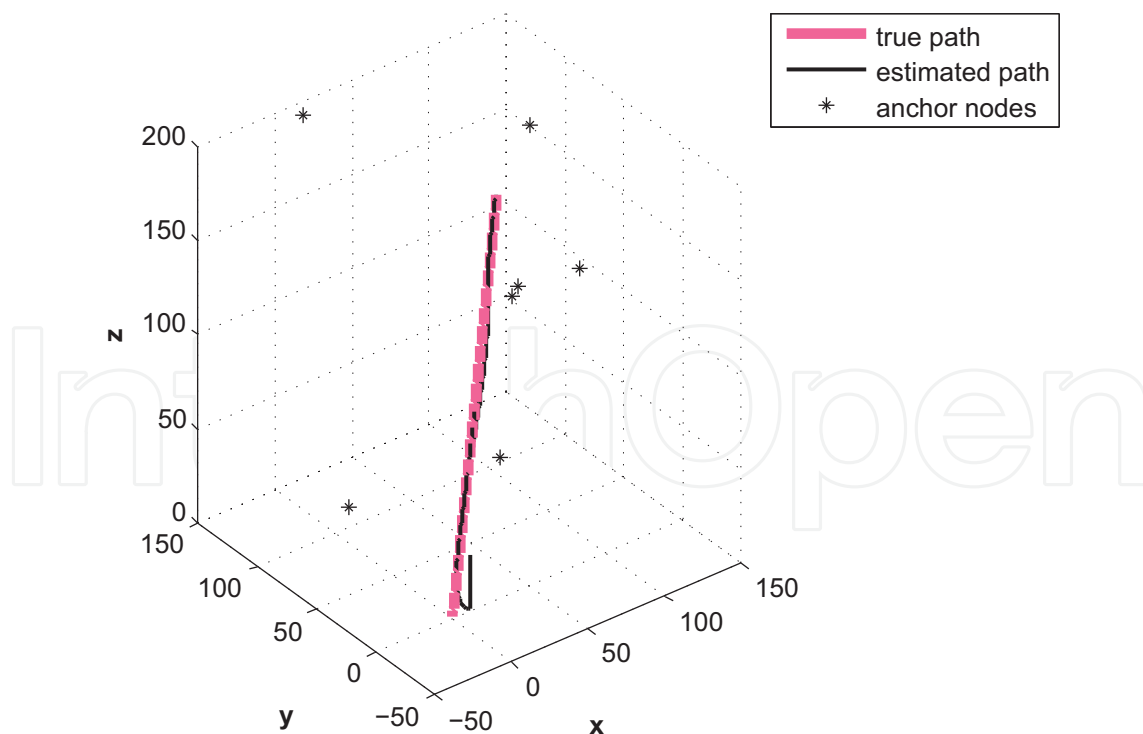
- a. A target node is moving 0.5 m in each of the three  $x$ ,  $y$ , and  $z$  axes in each of 200 steps, which gives a true track distance of 100 m/dimension. The true track is illustrated by the straight line in **Figure 6**. The estimated track begins with an initial point of (50, 50, 50) and converges to the true track for a while but then deviates from it due to the local minima associated with this problem. This deviation is shown clearly in **Figure 6**.
- b. The same scenario is repeated except that the track is divided into two segments. The first segment uses the same previous anchor nodes. In the second segment, the anchor nodes have been changed in an attempt to avoid the local minimum and resume tracking the true path. **Figure 7** shows the corrected tracking behavior and the new set of anchor nodes.
- c. The proposed method of Algorithm 1 is applied with  $N = 7$  resulting in  $P = 21$ , that is, seven anchor nodes are clustered in 21 sets of five anchor nodes each.  $M$  is chosen to be equal to 10 and  $L$  equal to 150. The threshold is chosen as  $\gamma = 7$ . At the 150th update, the final  $f(p)$  is calculated for each of the 10 sets. The sets that produce an error function greater than 7 are discarded, and other sets from the remaining 11 sets are chosen to complete the 10 sets starting with the final position of  $p$  that corresponds to the minimum  $f(p)$ . Iterative computations are continued for another 150 updates and the optimum set is



**Figure 6.** Tracking of a moving sensor in 3D space using iterative GD with initial point (50, 50, 50) and a fixed set of anchor nodes. Convergence factor = 0.1.



**Figure 7.** Two-segment true path and track of a moving sensor in 3D space using iterative GD. Initial point is (50, 50, 50). Convergence factor = 0.1.



**Figure 8.** GD tracking of a moving sensor using the proposed algorithm. Initial point is (50, 50, 50). Convergence factor = 0.1.

also found by inspecting the localized point that results in the minimum final  $f(p)$ . The true and estimated tracks are shown in **Figure 8**. Simulations show that the optimum set of anchor nodes in the first segment (150 iterations) is different from that of the second segment and no local minimum deviation is noticed.

It is worth noting that in the second segment, the first segment unsuccessful sets can be replaced in a deterministic manner rather than randomly, since one would by then have an idea of the location of the moving target. This is especially convenient for WSNs with widely scattered sensors, where sets with nodes that are distant from the moving target and that are likely to contribute to poor localization can be discarded.

Future work may consider introducing distance-measurement noise and studying its effect on the performance of the proposed algorithm. In that case, the final  $f(p)$  may not be enough indication of the validity of any certain set of anchors due to noisy measurements. So averaging  $f(p)$  of the last 10 iterations of each segment of the estimated path, and for all  $M$  running sets, may be considered to obtain a more accurate comparison and a judicious subsequent selection of sets.

### 3. Distributed gradient descent (GD) localization in 3D wireless sensor networks

In Ref. [7], the authors propose a distributed GD localization method that is robust against node and link failures. The computation of sums is inherent in the GD localization problem and can therefore be made distributed by applying gossip-based distributed summing or averaging algorithms.

It can be seen from Eqs. ((1), (6)–(8)) that, there are four  $N$ -term sums that have to be computed in each iteration of the GD localization algorithm. For each of the four sums, each set of variables that constitute each of the  $N$  terms is resident in one of the  $N$  anchor nodes. This set of variables includes the current tracked or estimated position, the corresponding distance measurement and the location of the anchor node itself. This readily implies the possibility of computing each of the four sums in a distributed manner by sharing information (gossiping) among the anchor nodes. Upon completion of the distributed averaging or summing task, each anchor will possess an estimated value of all four sums, and then Eq. (3) can be computed in each anchor to obtain the estimated position of the node to be localized. This whole process is repeated in each iteration of the GD localization algorithm.

The averaging or summing problem is the building block for solving many complex problems in signal processing. Gossip algorithms [15] are a class of randomized algorithms that solve the averaging problem through a sequence of pairwise averages. In our case, the communicating or gossiping nodes are the anchors, and we assume they are within transmission range of each other. Therefore, a simple gossip-based synchronous averaging protocol called the push-sum (PS) distributed algorithm [15, 16] is used for this application.

### 3.1. The push-sum gossip-based distributed averaging algorithm

The PS algorithm is iterative and not exact. Therefore, every anchor node will obtain an estimate of the sums that differ slightly from that of the other anchors. The gossiping anchor nodes are assumed to work synchronously. The term “iteration” will be preserved for the GD time step, whereas the term “round” or “PS round” will be used to indicate the PS time step. The total number of rounds will be designated as  $T$ . With every round  $t$ , a weight  $\omega(i)$  is assigned to each node  $i$ , and initialized to  $\omega(i) = 1/N$ , where  $N$  is the number of anchors. Likewise, a sum  $s(i)$  is initialized to  $s(i) = x(i)$ , where  $x(i)$  is the resident summation element in node  $i$ . For round  $t = 0$ , each node  $i$  sends the pair  $[s(i), \omega(i)]$  to itself, and in each of the remaining rounds  $t = 1, \dots, T$ , node  $i$  follows the protocol of Algorithm 2:

Algorithm 2: The push-sum algorithm {Pushsum( $x_i$ )} [15, 16]

**Input:**  $N$  and  $T$

1. Initialization:  $t = 0$ ,  $s(i) = x(i)$  and  $\omega(i) = 1/N$  for  $i = 1, \dots, N$ .
2. **Repeat.**
3. Designate  $\{\hat{s}(r), \hat{\omega}(r)\}$  as the set of all pairs sent to node  $i$  at round  $t-1$ .
4. Compute  $s(i) \equiv \sum_r \hat{s}(r)$  and  $\omega(i) \equiv \sum_r \hat{\omega}(r)$ .
5. At each node  $i$ , a target node  $f(i)$  is chosen uniformly at random.
6. The pair  $[0.5 s(i), 0.5 \omega(i)]$  is sent to target nodes  $f(i)$  and node  $i$  (the sending node itself).
7.  $[s(i)/\omega(i)]$  is the estimate of the sum at round  $t$  and node  $i$ .
8.  $t = t + 1$ .
9. **until**  $t = T$ .

**Output:**

$[s(i)/\omega(i)]$  is the sum at round  $t$  and node  $i$ .

$$\sum_{i=1}^N \omega(i) = 1 \text{ and } \sum_{i=1}^N s(i) = \text{the sum, at all rounds } t.$$

The number of steps  $T$  needed such that the relative error in Algorithm 2 is less than  $\varepsilon$  with probability at least  $(1 - \delta)$  is of order:

$$T(\delta, N, \varepsilon) = O\left(\log_2 N + \log_2 \frac{1}{\varepsilon} + \log_2 \frac{1}{\delta}\right) \quad (12)$$

where  $T$  is also referred to as the diffusion speed of the uniform gossip algorithm [15].

### 3.2. Distributed GD localization in WSNs

The PS distributed averaging method of Algorithm 2 is considered as scalar version. It can be extended to a vector version [17] where nodes (anchors) exchange vector messages that are

summed up element-wise. This concept readily conforms to our proposed distributed GD localization method in which we have to compute four sums in each iteration as in Eqs. ((1), (6)–(8)).

At the  $k$ th iteration and in the  $i$ th anchor node, there reside  $f(p_k)|_i$ ,  $\frac{\partial f}{\partial x}|_{k,i}$ ,  $\frac{\partial f}{\partial y}|_{k,i}$  and  $\frac{\partial f}{\partial z}|_{k,i}$  which can be considered the four elements of the vector.

The core idea of our distributed GD localization algorithm is that, for each outer gradient iteration, a series of inner rounds reach consensus on each of the four  $N$ -term sums.

### 3.2.1. Simulation results

The GD localization problem in a 3D space is simulated in MATLAB as in Ref. [7]. Four anchor node locations are chosen in a volume of  $100 \times 100 \times 100 \text{ m}^3$ . It is assumed that the target node to be localized has all anchors within its radio range. The same four anchors given in the simulation results of Section 2 are used. The targeted node is (60, 90, 60). Error-free TOA measurements are assumed, and centralized localization is first performed with  $N = 4$ ,  $\alpha = 0.25$  and  $p_o = (50, 50, 50)$ . After 100 iterations, it is found that the error function is 0.748 and the localized point is (60.1, 84.1, 58.8) which is very close to the targeted node.

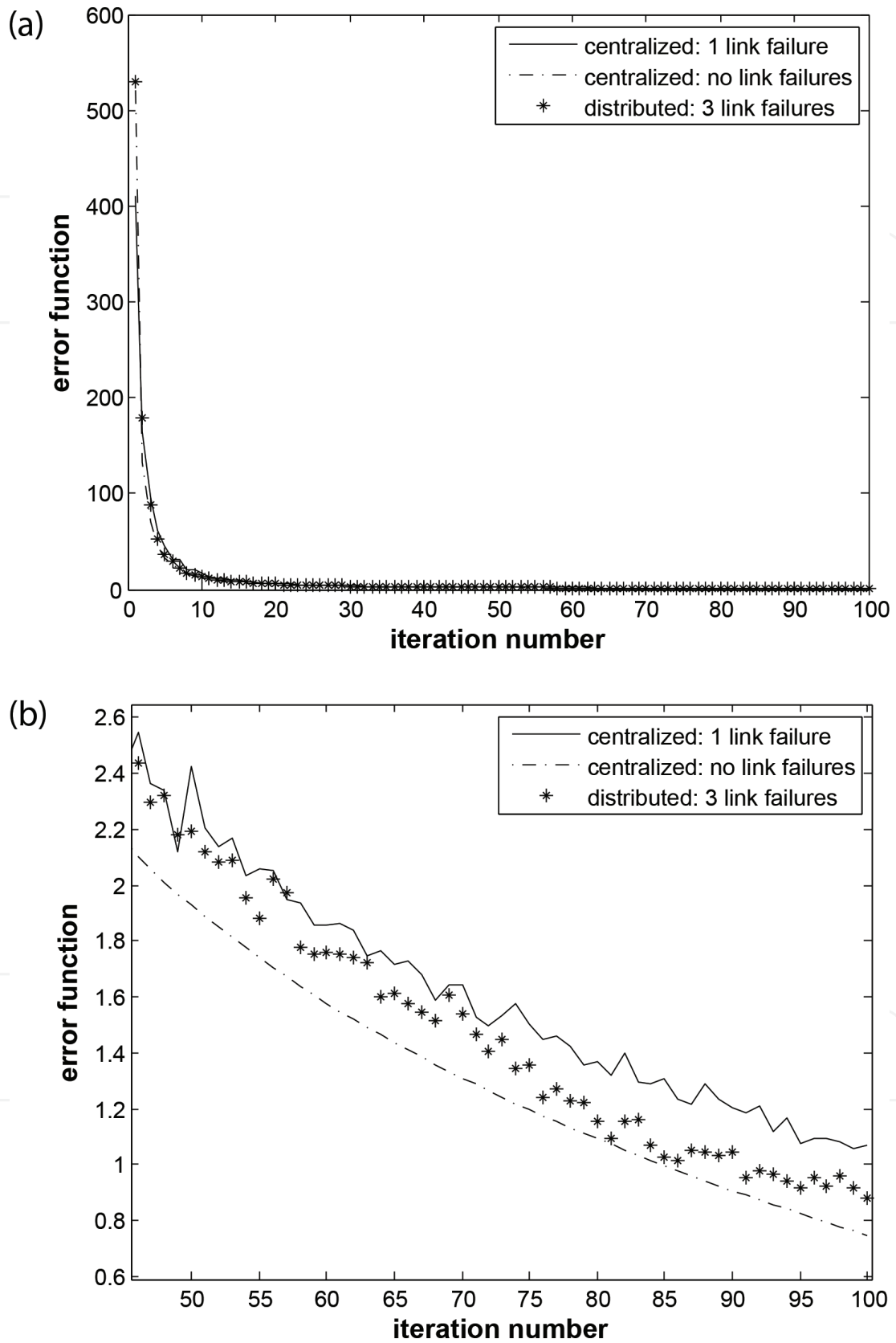
Treating the order in Eq. (12) as an exact value, we set the number of rounds of the PS algorithm (Algorithm 2),  $T$ , for a number of nodes  $N$ , equal to

$$T = \log_2 \left( \frac{N}{\delta \varepsilon} \right) \quad (13)$$

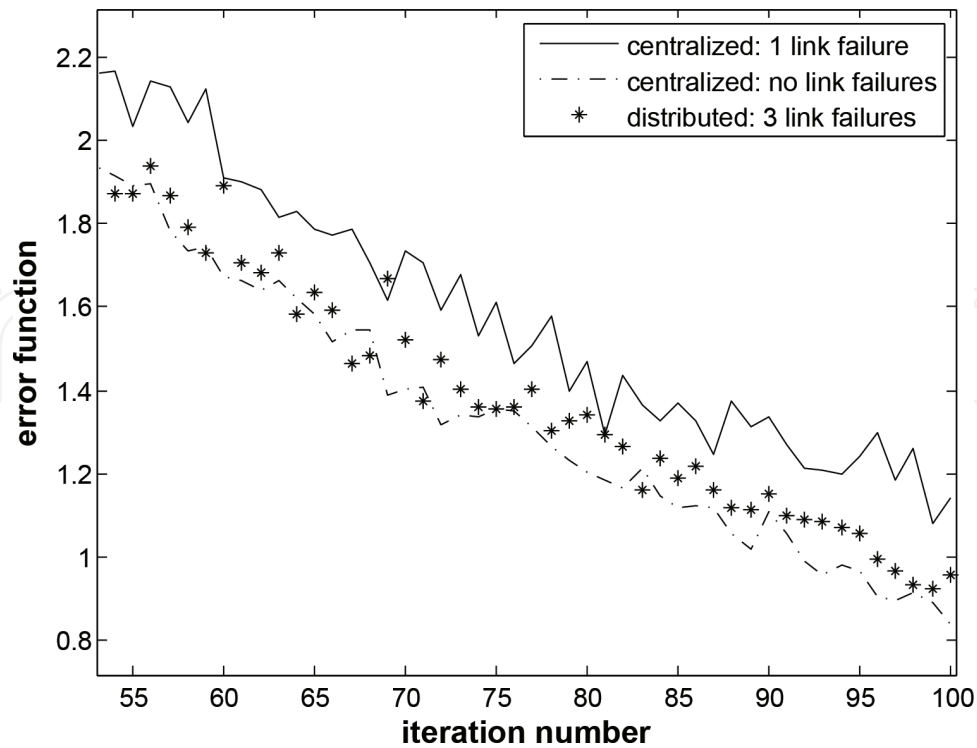
Note that  $\delta \varepsilon = 2^{-12}$  is obtained when we set  $\delta = \varepsilon = 2^{-6} \approx 0.0157$ . Substituting these values in Eq. (13), we find that  $T = 14$  PS rounds when  $N = 4$ . Clearly, this implies that we may expect a relative error  $\varepsilon \leq 0.0157$  with probability higher than 0.9843 in the PS algorithm. The final accuracy in the estimated localization corresponds to the accuracy level  $\varepsilon$  set in the PS algorithm [16]. Thus, from such estimated values of  $\varepsilon$  and  $(1 - \delta)$ , it can be deduced that the accuracy of our distributed localization algorithm is almost equivalent to that of centralized GD localization in the absence of noise and link failures.

Distributed algorithms are robust against network failures, or, typically, link failures. The latter arise due to many reasons such as channel congestions, message collisions, moving nodes, or dynamic topology [18]. Link failures can be modeled by the absence of a bidirectional connection between two nodes. All nodes operate in synchronism. At each time step, some percentage of the links between anchor nodes is randomly removed. The missing links may differ every time step since they are programmed to be randomly chosen, but their number remains fixed for each run of the code, and ensemble averaging over 100 trials is performed in each run. **Figure 9** demonstrates the robustness of the proposed distributed algorithm. Even if we lose up to 50% of the links in every time step, the algorithm is still comparatively accurate. This is illustrated by **Figure 9a** and **b** which are plots of the error function versus iteration number in the presence of link failures. For  $N = 4$ , the number of available links is 6, and losing three (50%) of which results in a final localized target point of (59.8, 82.8, 58.4) with an error function of 0.9 when  $\alpha = 0.25$  [7].





**Figure 9.** (a) Error function versus iteration number for centralized and proposed distributed GD localization for different cases of link failure conditions,  $\alpha = 0.25$ . (b) A close view of **Figure 9a** demonstrating the comparative performance of the different centralized and proposed distributed localization algorithms.



**Figure 10.** Comparative performance of the different centralized and proposed distributed localization schemes.  $\alpha = 0.25$ . TOA measurement noise SD = 0.5 ns.

For the purpose of comparison with centralized GD localization, we find that one link failure (25% of available links) isolates the corresponding node from the fusion center, and we have only three anchors to compute the target position, though randomly chosen in every time step. After ensemble averaging, the localized point is (60.1, 82.4, 58.6) and the error function is 1.0, again when  $\alpha = 0.25$ . This accuracy and in fact, even slightly better is achieved with the distributed scenario of four anchors and three link failures (50% of available links), which clearly shows the advantage of our proposed distributed localization algorithm over its centralized counterpart. The only disadvantage is that in every iteration, we must allow for a delay of 14 PS rounds ( $T$ ).

The simulations are repeated for noisy TOA measurements as shown in **Figure 10**. Gaussian measurement noise with zero-mean accounting for LOS arrivals only is assumed, and the SD is chosen to be 0.5 ns. This results in a distance error of 15 cm when UWB signals are used for sensing. The resulting plots are noticeably noisier than those of **Figure 9**, but are obviously interpreted in the same way as the noise-free cases. That is, the proposed distributed algorithm with three link failures (50% of links) performs better than the centralized algorithm with one link failure (25% of links) [7].

#### 4. Step size considerations

The fixed step size in this work should be chosen carefully; a too large step size would affect the performance advantage of the proposed distributed localization algorithm as well as the

centralized one, whereas a small step size would increase the error function. It is worth mentioning that there are instances in the literature on distributed GD localization algorithms where only the optimal step size is computed in a distributed manner [19, 20] rather than the GD sums in the present work. In Refs. [19, 20], the optimization of the step size in each iteration depends on the node positions and gradients. The optimization method is called the Barzilai-Borwein or simply BB method [21], in which the step size is updated at each iteration using the estimated target position and gradient vectors of the current and past time iterations.

The BB method cannot be applied successfully to our distributed GD localization under consideration [7], that is, by updating  $\alpha$  at each iteration and in each anchor. Applying the BB method yields favorable results that are superior to those with fixed step size only in the cases of centralized localization, and distributed localization in the absence of link failures which is an ideal situation not found in practice. The reason is obvious since, in our work, the gradient components are found through gossiping among anchors and become, therefore, greatly affected in case of link failures causing the BB method to result in pronounced sub-optimality in the computation of  $\alpha$  at each iteration and in each anchor. This conclusion was arrived at in Ref. [22], where the above situation was simulated and the BB method tested when applied to GD localization in WSNs. Linearly-varying step sizes are shown in Ref. [22] to have the best performance, as they do not involve gradient computations.

## 5. Recapitulation and future trends

The problem of sensor localization in a 3D space by the method of gradient descent has been investigated and solutions are presented to some impediments that are associated with the moving sensor case, namely, the local minima problem [6]. The proposed method considers all possible combinations of a certain chosen number of anchor nodes from a larger set of available anchors. The foreseen success of the proposed method stems from the fact that a deviating estimated path toward a local minimum is almost certain to return to the right track if some anchor nodes are replaced. This is true since anchor node replacement entails a change of the shape of the performance surface along with different local minima positions. The anchor nodes placement is made uniformly random as the true track of the moving sensor to be localized is unpredictable, and it is performed periodically. The simulation results demonstrate the success of this method. The advantage gained is at the expense of increased computational requirements, and the proposed method also necessitates faster data processing in order to perform accurate moving sensor localization in real time.

In Ref. [7], the GD localization algorithm in WSNs in a 3D space was combined with PS gossip-based algorithms to implement a distributed GD localization algorithm. The main idea is to compute the necessary sums by inter-anchor gossip. The method compared favorably with the centralized version as regards convergence, accuracy, and resilience against noise and link failures. Our simulation results demonstrate that centralized processing with four anchors

and one link failure (25% of the links) introduce a localization error comparable to (and even slightly greater than) that introduced by the proposed distributed processing method with three link failures (50% of the links). This is achieved when the number of PS rounds is suitably selected.

Despite the inevitable degradation of performance in case of noisy TOA measurements, the proposed distributed method retains its advantages over centralized processing with proper selection of the GD step size and number of PS rounds. It is therefore evident that resort to distributed techniques such as the proposed distributed GD localization algorithm [7] ensures robustness against link failures even in the presence of noisy TOA measurements, eliminates the need for a computationally-demanding central processor, and avoids a possible communication bottleneck at or near the fusion center [10].

As a future trend, compressive sensing (CS) or random sampling can be implemented to track a moving node in a centralized WSN using the iterative GD algorithm resulting in remarkable energy efficiency with tolerable error [23]. Moreover, an efficient approach for (pseudo-) random sampling via chaotic sequences that has first appeared in Ref. [24] could initiate further investigation of CS concepts via chaos theory and the possibility of their application to WSN moving node tracking.

## Author details

Nuha A.S. Alwan<sup>1\*</sup> and Zahir M. Hussain<sup>2,3</sup>

\*Address all correspondence to: n.alwan@ieee.org

1 College of Engineering, University of Baghdad, Baghdad, Iraq

2 College of Computer Science and Mathematics, University of Kufa, Najaf, Iraq

3 School of Engineering, Edith Cowan University, Joondalup, Australia

## References

- [1] Zhang L, Tao C, Yang G. Wireless positioning: Fundamentals, systems and state of the art signal processing techniques. In: Melikov A, editor. Cellular Networks—Positioning, Performance Analysis, Reliability. InTech; Rijeka, Croatia. 2011. pp. 3–50. ISBN: 978-953-307-246-3
- [2] Alrajeh NA, Bashir M, Shams B. Localization techniques in wireless sensor networks. International Journal of Distributed Sensor Networks. 2013;2013:9. DOI: 10.1155/2013/304628
- [3] Shareef A, Zhu Y. Localization using extended Kalman filters in wireless sensor networks. In: Moreno M, Pigazo A, editors. Kalman Filter: Recent Advances and Applications. I-Tech; Rijeka, Croatia. 2009. pp. 297–320. ISBN: 978-953-307-000-1

- [4] Qiao D, Pang GKH. Localization in wireless sensor networks with gradient descent. In: Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (Pac-Rim); 23 August 2011; Victoria, BC, Canada. IEEE; 2011. pp. 91–96
- [5] Garg R, Varna AL, Wu M. Gradient descent approach for secure localization in resource constrained wireless sensor networks. In: International Conference Acoustics, Speech and Signal Processing (ICASSP); 14 March 2010; Dallas, TX, USA. IEEE; 2010. pp. 1854–1857
- [6] Alwan N AS, Mahmood AS. On gradient descent localization in 3-D wireless sensor networks. *Journal of Engineering*. 2015;**21**:85–97
- [7] Alwan NAS, Mahmood AS. Distributed gradient descent localization in wireless sensor networks. *Arabian Journal for Science and Engineering*. 2015;**40**:893–899. DOI: 10.1007/s13369-014-1552-2
- [8] Wang J, Ghosh RK, Das SK. A survey on sensor localization. *Journal of Control Theory Applications*. 2010;**8**:2–11. DOI: 10.1007/s11768-010-9187-7
- [9] Gustafsson F, Gunnarsson F. Mobile positioning using wireless networks. *IEEE Signal Processing Magazine*. 2005;**22**:41–53. DOI: 10.1109/MSP.2005.1458284
- [10] Patwari N, Ash JN, Kyperoutas S, Hero AOIII, Moses RL, Correal NS. Locating the nodes. *IEEE Signal Processing Magazine*. 2005;**22**:54–69. DOI: 10.1109/MSP.2005.1458287
- [11] Kwon YM, Mechtov K, Sundresh S, Kim W, Aga G. Resilient localization for sensor networks in outdoor environments. In: The 25th IEEE International Conference on Distributed Computing Systems (ICDCS 2005); 6-10 June 2005; Columbus, OH, USA. IEEE; 2005. pp. 643–652
- [12] Li X, Hua B, Shang Y, Xiong Y. A robust localization algorithm in wireless sensor networks. *Frontiers of Computer Science China*. 2008;**2**:438–450. DOI: 10.1007/s11704-008-0018-7
- [13] Agarwal A, Daume HIII, Phillips JM, Venkatasubramanian S. Sensor network localization for moving sensors. In: Proceedings of the 12th International Conference on Data Mining Workshops (ICDMW 2012); 10 December 2012; Brussels: IEEE; 2012. pp. 202–209
- [14] Agarwal A, Phillips JM, Venkatasubramanian S. Universal multidimensional scaling. In: Proceedings of the 16th International Conference on knowledge discovery and data mining (ACM SIGKDD 2010); 25-28 July 2010; Washington, DC. ACM; 2010. pp. 1149–1158
- [15] Kempe D, Dobra A, Gehrki J. Gossip-based computation of aggregate information. In: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science; 11-14 October 2003; Cambridge, MA. IEEE; 2003. pp. 482–491
- [16] Dumard C, Riegler E. Distributed sphere decoding. In: International Conference on Telecommunications (ICT'09); 25-27 May 2009; Marrakech. IEEE; 2009. pp. 172–177
- [17] Strakova H, Gansterer WN. A distributed Eigensolver for loosely coupled networks. In: The 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP); 27 Feb-1 March 2013; Belfast. IEEE; 2013. pp. 51–57

- [18] Sluciak O, Strakova H, Rupp M, Gansterer WN. Distributed Gram-Schmidt orthogonalization based on dynamic consensus. In: The 46th Asilomar Conference on Signals, Systems and Computers; 4-7 November 2012; Pacific Grove CA. IEEE; 2012. pp. 1207–1211
- [19] Calafiori GC, Carlone L, Wei M. A distributed technique for localization of agent formations from relative range measurements. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*. 2012;**42**:1065–1076. DOI: 10.1109/TSMCA.2012.2185045
- [20] Calafiori G, Carlone L, Wei M. A distributed gradient method for localization of formations using relative range measurements. In: Proceedings of the 2010 IEEE International Symposium on Computer-Aided Control System Design (CACSD'10); 8-10 September 2010; Yokohama. IEEE; 2010. pp. 1146–1151
- [21] Barzilai J, Borwein JM. Two point step size gradient methods. *IMA Journal of Numerical Analysis*. 1988;**8**:141–148. DOI: 10.1093/imanum/8.1.141
- [22] Alwan NAS. Adaptive step sizes for gradient descent localization in wireless sensor networks. *International Journal of Information and Communication Technology Research*. 2016;**6**:1–7
- [23] Alwan NAS, Hussain ZM. Compressive sensing for localization in wireless sensor networks: An approach for energy and error control. Submitted for publication. 2017
- [24] Nguyen LT, Phong DV, Hussain ZM, Huynh HT, Morgan VL, Gore JC. Compressed sensing using chaos filters. In: Australasian Telecommunication Networks and Applications Conference (ATNAC 2008); 7-10 December 2008; Adelaide, SA. IEEE; 2008

