

A Comprehensive Investigation on Range-free Localization Algorithms with Mobile Anchors at Different Altitudes

Francesco Betti Sorbelli^a, Sajal K. Das^a, Cristina M. Pinotti^b, Giulio Rigoni^c

^a*Dept. of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA*

^b*Dept. of Computer Science and Math., University of Perugia, Italy*

^c*Dept. of Computer Science and Math., University of Florence, Italy*

Abstract

In this work, the problem of localizing ground devices (GDs) is studied comparing the performance of four range-free (RF) localization algorithms that use a mobile anchor (MA). All the investigated algorithms are based on the so-called heard/not-heard (HnH) method, which allows the GDs to detect the MA at the border of their antenna communication radius. Despite the simplicity of this method, its efficacy in terms of accuracy is poor because it relies on the antenna radius that continuously varies under different conditions. Usually, the antenna radius declared by the manufacturer does not fully characterize the actual antenna radiation pattern. In this paper, the radiation pattern of the commercial DecaWave DWM1001 Ultra-Wide-Band (UWB) antennas is observed in a real test-bed at different altitudes for collecting more information and insights on the antenna radius. The compared algorithms are then tested using both the observed and the manufacturer radii. The experimental accuracy is close to the expected theoretical one only when the antenna pattern is actually omnidirectional. However, typical antennas have strong pattern irregularities that decrease the accuracy. For improving the performance, we propose range-based (RB) variants of the compared algorithms in which, instead of using the observed or the manufacturer radii, the actual measured distances between the MA and the GD are used. The localization accuracy tremendously improves confirming that the knowledge of the exact antenna pattern is essential for any RF algorithm.

Keywords: Drone, rover, localization, range-free, range-based, antenna model

1. Introduction

Considering the recent interest in the Internet of Things (IoT), and in general in the Wireless Sensor Networks (WSNs), the problem of localizing sparse wireless sensors is more and more appealing to researchers. Due to their limited size and low cost, those sensors can be installed everywhere for any specific task. For instance, localization can help to search and rescue people after natural disasters, to monitor the structural health of building, or even to track people during the COVID-19 pandemic in order to decrease the circulation of the virus.

Technically speaking, the localization problem aims at estimating the position of *ground devices* (GDs) deployed on the WSN. For this important task, special devices whose positions are known a-priori, called *anchors*, are in charge of localizing the GDs. In the literature, localization can be classified in respect to the type of anchors deployed, i.e., *static anchors* (SAs) or *mobile anchors* (MAs). The former scenario requires a discrete amount of SAs with fixed positions, while the latter

one requires a single MA (e.g., rover, drone) with GPS capabilities (need of fresh coordinates while moving). Notice that the usage of SAs is not affordable in terms of costs when the WSN is large; analogously, such a fixed infrastructure cannot be quickly reused elsewhere. Hence, in this paper we consider only the localization with a single MA. Localization algorithms can be also broadly categorized as *range-free* (RF) or *range-based* (RB) approaches [1]. In the previous, the position is estimated only by discovering if the GD and MA are one in the range of the other. In the latter, the position of the GD is estimated by taking measurements (e.g., distance, angle, signal strength) between it and the MA. Usually, RB algorithms are known to be more accurate than RF ones but at the cost of additional specialized hardware. Finally, among the RF algorithms, the *radius-based* algorithms assume the knowledge of the transmission radius, while the *radius-free* ones do not.

In this paper, we evaluate on a test-bed the accuracy of many RF algorithms. They exploit the *heard/not-heard* (HnH) method, consisting of detecting two consecutive messages transmitted by the MA, one heard and one not-heard from the GD, called *endpoints*. All the algorithms, but one which is new, have been described in the literature under the *ideal model*. The ideal model assumes that both MA and GD are equipped with an isotropic antenna whose radiation pattern is a perfect sphere. Consequently, the transmission and receiving areas, which are given by the intersection of the sphere with the earth surface, are perfect circles. The ideal model can be emulated in reality only with expensive hardware on noiseless areas, which are not the usual operating conditions of most localization applications that occur, for example, in precision agriculture or search-and-rescue context. To reproduce the typical operating conditions, we adopt for our test-bed a set of inexpensive DecaWave DWM1001 Ultra-Wide-Band (UWB) antennas and an off-the-shelf 3DR Solo drone as the flying MA. Our pursued goals can be summarized as follows:

- We experimentally evaluate the performance of many localization algorithms, with very little knowledge on the antenna radiation pattern;
- We study how the MA's altitude can affect the localization performance.

The experimental results raised difficulties, doubts, and questions. Firstly, collecting a large set of endpoints on the field has proven to be challenging. So, we search for the statistical distribution that best fits the already observed endpoints for generating a large new set of synthetic endpoints. Such a set is used for testing the compared algorithms. To mitigate the poor observed performance in the real scenario, we decide to exploit the DWM1001's capability of taking distance measurements to gain awareness of the antenna radiation pattern. Despite this strategy makes the RF algorithms actually RB, the original algorithmic rules are kept, and the awareness of the exact antenna radiation pattern, though incomplete, significantly improves the performance.

The paper, which is an extension on [2], is structured as follows: Sec. 2 reviews the literature on localization. Sec. 3 presents the compared algorithms. Sec. 4 describes the test-bed. Sec. 5 analyzes the antenna's performance. Sec. 6 evaluates the algorithms under different scenarios. Sec. 7 shows how the algorithms can incorporate the distance measurements. Sec. 8 offers conclusions.

2. Related Work

In this section we cover the state-of-the-art about localization in WSNs, giving first a brief overview about known techniques with SAs, and then surveying more in detail techniques using a MA. Finally, we describe a few works that implemented real test-beds about localization.

The Static Anchors Scenario. In literature, different algorithms exist to tackle the issue of localization of GDs using SAs. DV-HOP based techniques [3] and Amorphous algorithms [4] approximate the distance between GDs using the number of hops between them and estimating the average hop distance inside the WSN. Then, using the trilateration method, each GD computes its own position. The Centroid algorithm used in [5] is another technique where each SA broadcasts its position to the surrounding, and each GD computes its own position calculating the average of all the coordinates of the SAs that it can hear. Other works propose solutions based on the Approximate Point in Triangulation (APIT) [6] whose goal is to divide the area in triangles in which unknown GDs reside. In each overlapped section (i.e., a polygon) of triangle area, the position of the unknown GD is computed calculating the centroid. The main issue for these solutions is the relative high number of SAs required for an acceptable localization error.

The Mobile Anchor Scenario. All these algorithms emulate multiple SAs with a single MA that continuously broadcasts its position. For localizing GDs, a MA has to plan a route (static path) in advance inside the WSN. On that route, the MA estimates the distance between itself and the GDs in range, and eventually the GDs' positions are computed performing trilateration. In [7] three different 2D movement trajectories have been studied, i.e., SCAN, DOUBLE-SCAN, and HILBERT. The distance between two consecutive segments of the trajectories is defined as the resolution. The simplest algorithm is SCAN, in which the MA follows a path formed by vertical straight lines interconnected by horizontal lines. Essentially, the MA sweeps the area along the y -axis. The main drawback is that it provides a large amount of collinear endpoints. In order to resolve the collinearity problem, DOUBLE-SCAN sweeps the sensing area along both the x -axis and the y -axis. However, in this way the path length is doubled compared with SCAN. Finally, a level- n HILBERT curve divides the area into 4^n square cells and connects the centers of those cells using 4^n line segments, each of length equal to the length of the side of a square cell. Generally, HILBERT provides more non-collinear endpoints but the path length can be very long if the resolution increases. All the above techniques are based on straight lines and suffer from collinearity problem. In order to heavily reduce collinearity, S-CURVES [8] has been introduced, which is similar to SCAN except that it uses curves rather than straight lines. Even though the collinearity problem is almost resolved, the main problem is that it does not properly cover the four corners of the squared sensing area. One of the best techniques is LMAT [9]. The main idea is to plan a series of MAs in such a way as to form equilateral triangles, avoiding collinearity problems. Each GD inside a triangle is localized by a trilateration procedure using the three vertices.

All the previous algorithms are range-based since they rely on measurements (signal strength, distance, etc.). Only a few range-free techniques have been proposed in the literature. Among them, the ones proposed by Xiao [10] and Lee [11] rely on a ground MA, while the one proposed by Betti [12], relies on a flying MA. We will give more details and insights about those algorithms in Sec. 3.

Implemented Test-beds. To the best of our knowledge, only a few test-bed implementations have been done aimed at comparing the performance among RF algorithms with a MA. Recently in [13], a test-bed using inexpensive UWB antennas and a drone as MA evaluates the localization accuracy of the RF DRF algorithm proposed in [12]. Such an algorithm strictly relies on the good quality of the antenna radiation pattern and requires simple geometrical rules for estimating the GD's position. Unfortunately, in practice, the experimental localization error obtained by the implemented DRF algorithm is large. Another recent study in [14] shows that the irregularities of the hard-

ware antenna radiation pattern can heavily affect the air-to-ground (A2G) link quality between a MA (drone) and a GD. The authors study the A2G link quality of BroadSpec UWB antennas from Time Domain Inc by observing the Received Signal Strength Indication (RSSI). They show a dependency of the link quality on the antennas' orientations, their elevation, and their distance. The authors in [15] report the UWB A2G propagation channel measurements in an open field using a drone. Three scenarios with different obstacles are considered while a drone was orbiting above a GD at different altitudes and ground distances. Also, different antenna orientations are considered for the drone. Experimental results show that the received power is highly dependent on the antenna gain of the line-of-sight (LoS) component in the elevation plane when the antennas are aligned (same orientation). In a work proposed in [16], authors investigate a time difference of arrival (TDoA) based approach for localizing drones taking into account a simple A2G 3D antenna radiation pattern. Experimental results show that accounting for antenna effects makes a significant difference and reveals many important relationships between the localization accuracy and the altitude of the drone. Most importantly, they finally show that the localization performance varies in a non-monotonic pattern with respect to the drone altitude.

Motivations. Considering that the RF algorithms have been around for a long time, that the algorithm tested in [13] on the field using a drone (i.e., flying MA) resulted in bad performance, and that the irregularities showed in [14, 15, 16] are for A2G links, we started to think that altitude could be one of the main cause of poor results for the algorithm tested in [13] and in general for RF algorithms. Therefore, these considerations motivate us to investigate more concerning the accuracy of RF algorithms at different altitudes and the quality of antennas.

3. The Range-free Algorithms

In this section, we describe the RF algorithms that we compare. We start defining a *rover* as a MA that moves on the ground, and a *drone* as a MA that flies in the sky. During the localization procedure, called *mission*, the MA visits specific points in its path, called *waypoints*. Along this path, the MA continuously transmits a *beacon* that can be used by any GD within the communication range for estimating its position. The beacon includes the current MA's GPS position and it is sent at regular intervals of time. The distance among any two consecutive transmitted beacons is called *inter-waypoint* distance I_w and depends on the MA's current speed. All the algorithms that we study exploit the HnH method relying on the detection of endpoints. Performing HnH, the GD learns that the MA is currently transmitting at its transmission area border. Moreover, the GD learns that the last (first) not-heard beacon is at a distance I_w from the first (last) heard beacon (endpoint), and hence at a distance no more I_w from the edge of the transmission area. In general, three applications of HnH are required for localizing the GD.

In the following, we describe three RF algorithms that we evaluate, namely, DRF [12], XIAO [10], and LEE [11]. We also introduce a new variant of DRF, called DRFE. In these algorithms, the MA travels along a static path for localizing the GD. The static path is formed by a sequence of consecutive straight segments in which the MA regularly broadcasts message beacons (its current position) that the GD is listening to. Once the GD has collected enough information, it can finally compute and estimate its position according to the performed algorithm.

3.1. The DRF Algorithm

DRF [12] is a lightweight RF radius-free algorithm designed for drones. This algorithm is based on the notion of *chord*. In geometry, the perpendicular bisector of any circle's chord passes

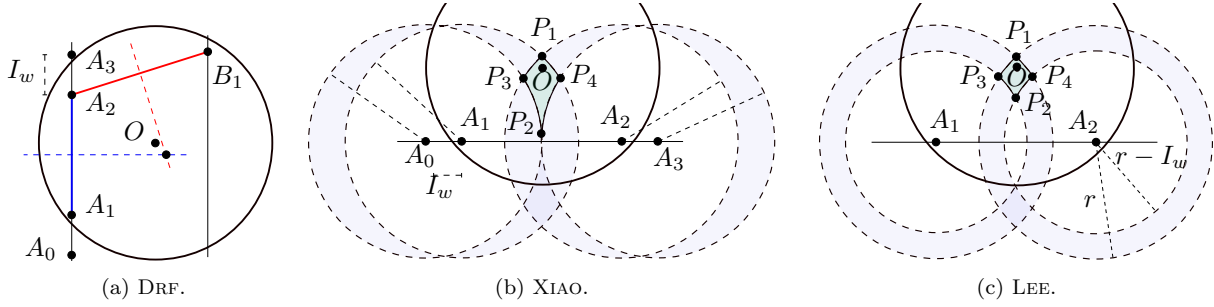


Figure 1: The DRF, XIAO, and LEE localization algorithms. In XIAO e LEE there are two symmetric intersection areas: a third point (not illustrated) is required to find and disambiguate the intersection area where GD resides.

through the center O of the circle itself. So, the bisector of another non-parallel chord and the previous one intersect at O point. In Fig. 1(a), the GD is located at the point O , while initially the MA travels along the segment that intersects the points in sequence A_0 first and then A_3 . The radio receiving area of the GD is identified by the circle centered at O , so if the MA transmits a message outside that circle (e.g., in A_0), the GD cannot hear any message. However, when the MA crosses such a circle and transmits in A_1 , the GD can now receive and record messages, because the relative distance between them is less than or equal to the transmitting/receiving radius. The same reasoning can be applied for the points A_2 (heard) and A_3 (not heard). Accordingly, the first chord is denoted by the segment with endpoints A_1A_2 . It is easy to understand that when the MA crosses the GD's receiving area along another segment (e.g., the one that intersects the point B_1), another endpoint is detected, and eventually two chords are identified by the pairs A_1A_2 and A_2B_1 . Finally, the GD starts to estimate its position once it has detected these two chords computing first the associated perpendicular bisectors (dashed lines) and then their intersection point (which can be different from point O). The detection of chords incurs several problems that eventually affect the localization accuracy. Recalling that the MA regularly broadcasts its current position (waypoint) at discrete intervals of time and that two consecutive waypoints are at distance I_w , the endpoints of the chords may not exactly fall on the circumference of the receiving disk, even if the receiving disk is a perfect circle (e.g., A_2 and A_3).

3.2. The XIAO Algorithm

XIAO [10] is a RF radius-based localization algorithm initially developed for ground MAs. Like DRF, the XIAO algorithm exploits the HnH method in order to detect special points used for building a constrained area that bounds the GD's position. Unlike DRF, XIAO also uses the value of the communication radius r . In Fig. 1(b), the GD is located at the point O while the MA travels along the segment that intersects first the point A_0 and then A_3 . Once applied the HnH method, the GD initially detects the first pair of heard endpoints A_1 and A_2 . However, since the segment lies on a straight line and the value of I_w is known, the GD can also compute two additional non-heard beacons, i.e., A_0 and A_3 , associated with A_1 and A_2 , called pre-arrival and post-departure. Then, four circles of radius r centered at each of these four points are drawn. Those circles create two symmetrical intersection areas (e.g., the first one is bounded by the points P_1, P_4, P_2, P_3) where the GD may reside. Hence, the GD's position can be at the "center" of one of the two intersection areas. In order to disambiguate in which intersection area the GD resides, a third HnH beacon is required, and the final estimated position is the one which has the closest distance, from that third

point, to the radius. The definition of center varies depending on whether the intersection area is delimited by four or five vertices [10].

3.3. The LEE Algorithm

LEE [11] is a RF radius-based algorithm very similar to XIAO. It builds a constrained area using the HnH method and the knowledge of both r and I_w , similarly to XIAO. In Fig. 1(c), once the GD has detected the two extreme endpoints (A_1 and A_2), it traces two circles of radius r and $r - I_w$ on both the points, which create two annuli, intersecting in two distinct and symmetrical intersection areas (e.g., one is bounded by the points P_1, P_4, P_2, P_3). Finally, GD resides at the center of one of such areas, and a third endpoint is used to disambiguate the correct one.

3.4. The Proposed DRFE Algorithm

Now we present a new RF algorithm, called DRFE, that shares the “chord” idea with the DRF algorithm and the radius information r with XIAO and LEE. In the special case that the two endpoints A_1 and A_2 that delimit the chord exactly lie on the circumference, the GD resides on the point O of the perpendicular bisector that is distant r from the two endpoints. Precisely, there are two possible locations for O , one on the left and one on the right of the chord. Usually, using a third endpoint non-collinear with A_1 and A_2 , it is possible to disambiguate the correct intersection.

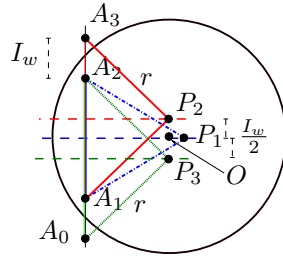


Figure 2: The points P_1 , P_2 , and P_3 of the DRFE algorithm.

In general, since the MA’s path is sampled with discrete beacons at distance I_w among them, A_1 and A_2 may not lie on the circumference and thus GD may not exactly reside on the perpendicular bisector of the chord $\overline{A_1A_2}$, but in its vicinity. So, to find the GD’s position, DRFE repeats the above construction for the three chords $\overline{A_1A_2}$, $\overline{A_0A_2}$, and $\overline{A_1A_3}$, where A_0 and A_3 are the two non-heard beacons associated with the endpoints A_1 and A_2 , as illustrated in Fig. 2. From the three chords $\overline{A_1A_2}$, $\overline{A_1A_3}$, and $\overline{A_0A_2}$, three intersection points P_1 , P_2 , and P_3 are obtained at distance r from the endpoints of their chord. In other words, P_1 , P_2 , and P_3 form three isosceles triangles $\triangle(A_1A_2P_1)$, $\triangle(A_1A_3P_2)$, and $\triangle(A_0A_2P_3)$ with two oblique sides of equal length r . As before, this construction finds three vertices on the left of the chord and three vertices on the right that will be disambiguated using another non-collinear endpoint (not illustrated). Let us suppose that GD is on the right of the chord $\overline{A_1A_2}$. DRFE places GD at the centroid of P_1 , P_2 , and P_3 on the right of the chord $\overline{A_1A_2}$. Since the RF radius-based algorithms require the knowledge of the transmission radius, in the next next we describe how to collect such information in a real test-bed.

4. Test-bed Setup

In this section, we describe the test-bed and the used hardware. We fix a Cartesian coordinate system with origin at the special position HOME $(0, 0, h_0)$, with $h_0 = 1$ m. At HOME, we place

the GD’s antenna at the top of a tripod of height h_0 . Also, the MA is equipped with an antenna. When we set $h = 0$ m, we refer to a rover mission where the MA and the GD are placed at h_0 , while when we set $h > 0$ m, we refer to a drone mission flown at an altitude $h_0 + h$.

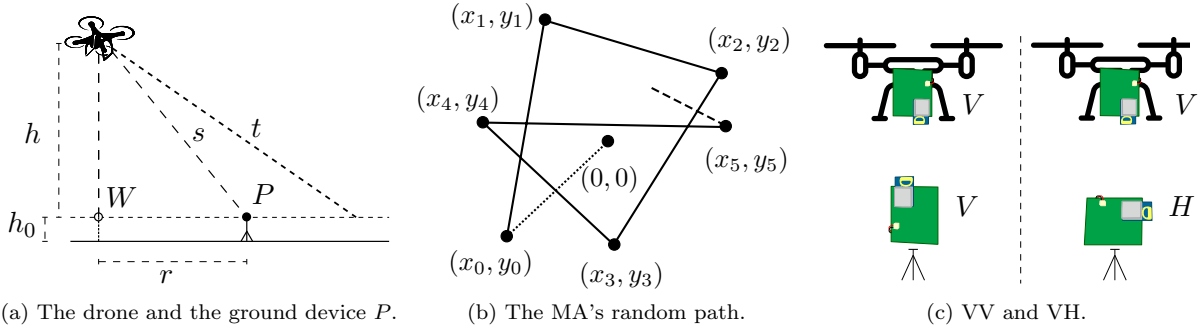


Figure 3: The test-bed setup.

In our test-bed, we employ different hardware components: a few DecaWave DWM1001 UWB antennas [17], a Raspberry Pi, and a 3DR Solo drone [18]. According to DecaWave, the transmission radius is 60 m and hence, in this paper, we set the manufacturer radius to $r_0 = 60$ m. The main component that pilots the MA and sends commands to the GD is the Raspberry Pi. The Raspberry can be used for the experiments on the ground using a rover, or together with the drone for the aerial ones. In the former case, i.e., MA as a rover, we simulate the rover’s behavior by just walking in the field at a regular walking speed (about 3 km/h as measured by a smart-watch) keeping the Raspberry on the hands at $h_0 = 1$ m above the ground. Moreover, since the rover has to send GPS positions, we rely on a cheap USB GPS module connected to the Raspberry. Finally, in the latter case, i.e., MA as a drone, we use the 3DR Solo drone which is able to fly up to 25 min [18].

A mission is a static path Π that consists of n segments S_i , $i = 0, \dots, n - 1$. Such a list of n segments is made by generating $n + 1$ random points in the deployment area. Each segment is delimited by two random points, and any two consecutive segments share one random point by setting the waypoints’ coordinates (x_{W_i}, y_{W_i}, h) for each W_i , $i = 0, \dots, n - 1$ (see Fig. 3(b)). When all the waypoints are generated, the mission starts. Once the first waypoint of Π is reached, the drone/rover starts to send message beacons according to its current position by converting GPS coordinates in local Cartesian (x, y) positions. This process continues until the MA reaches the last waypoint of Π . When the mission is accomplished, the MA comes back to HOME.

Regarding the GD, in our experiments, we set its antenna on the tripod placed at HOME, lying on two different planes, as sketched in Fig. 3(c) (left and right). According to the DecaWave’s Datasheet document [19], there are three planes in the spherical space with respect to the antenna’s center, i.e., xz , xy , and yz . Each plane experiences a different radiation pattern. In Fig. 3(c), we show the GD’s antenna when it is in the xz plane (bottom left) and in the yz plane (bottom right). We denote as *vertically placed* an antenna that lies in the xz plane, and as *horizontally placed* an antenna that lies in the yz plane. The drone’s antenna is always vertically placed in the xz plane (Fig. 3(c), top left and top right) keeping the UWB transceiver at the bottom for guaranteeing the most available free space. For simplicity, we indicate the first configuration (Fig. 3(c), left), where the two antennas lie on the same side, but in opposite direction, with vertical-vertical (VV); whereas we refer to the other configuration (Fig. 3(c), right) with vertical-horizontal (VH).

5. Antenna Analysis

In this section, we recap the UWB technology, report the DecaWave’s technical datasheet information, and analyze the experimental data.

5.1. The Ultra Wide Band Technology

UWB is a promising radio technology that can use a very low energy level for short-range, high-bandwidth communications over a large portion of the radio spectrum. Nowadays, its primary purpose is in the field of location discovery and device ranging. Differently from both Wi-Fi and Bluetooth, UWB is natively more precise and accurate, uses less power and, as production of UWB chips blows up over time, holds the promise of a lower price point. Moreover, UWB offers relative immunity to multipath fading.

In this paper we rely on a kit of DWM1001 UWB antennas produced by DecaWave. According to DecaWave’s datasheet document [17], those antennas provide 10 cm accuracy for the measurements. Moreover, those chips have a 6.5 GHz center frequency, and have a point-to-point range up to 60 m in a line-of-sight (LoS) scenario and up to 35 m in a non line-of-sight (NLoS) scenario. Although the DWM1001 chip transmitting power is set to -41.3 dBm/MHz, and the typical receiver sensitivity is -93 dBm/500MHz [17], the received power is influenced by the antenna polarization.

5.2. Datasheet Antenna Information

Fig. 4 shows the antenna radiation patterns of the UWB antennas according to DecaWave’s document (*Datasheet for the DWM1001C*, Tab. 12 [19]) for different configurations. The solid dark line of Fig. 4(a) shows that it is possible to obtain the same gain in all the directions in the xz -plane when an antenna vertically placed (i.e., on xz -plane) is observed by another antenna which shares the same *vertical* orientation (Φ polarization), i.e., they are concordant. We recreate this situation by implementing the antennas as VV. Thus, we expect that the VV configuration experiences the same gain at different angles, at least when the two antennas are at the same height.

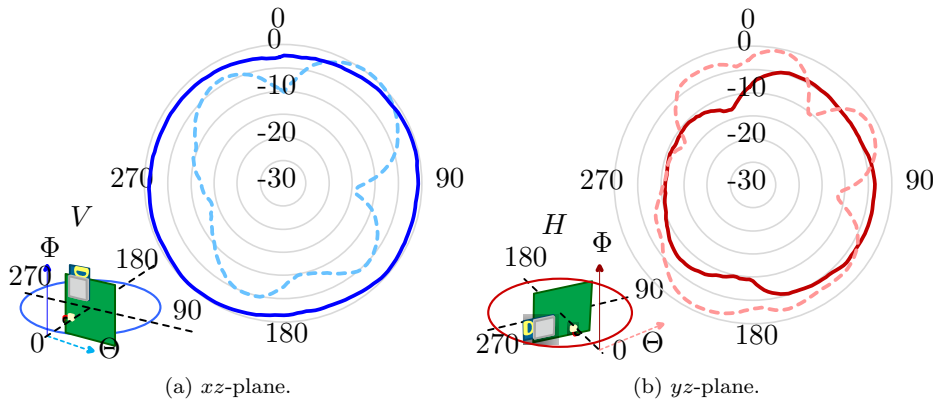


Figure 4: The DWM1001 radio pattern: dBm vs angle.

Both the dashed lines of Fig. 4(a) and Fig. 4(b) refer to the VH configuration because they show the gain when an antenna is observed by another antenna *perpendicularly* oriented (Θ polarization), i.e., they are discordant. Both the dashed lines have nulls at certain angles that can limit the gain and can introduce “holes” and “bubbles” in the pattern. Thus, we expect that the VH configuration experiences different gains at different angles, and we also expect a relevant variability given that the

dashed lines of Fig. 4(b) and Fig. 4(a) are different, although the relative position of the antennas seems to be the same. Analyzing these technical data, it seems that the gain is omnidirectional at least when the two antennas are placed as VV.

Table 1: The DW1000 elevation gains (in dBm) at 6.2 GHz.

		Θ (discordant)	Φ (concordant)
yz -plane	peak	0.30	2.92
	average	-6.99	-3.04
xz -plane	peak	0.26	1.39
	average	-5.74	-3.90

We have not found, for DWM1001, any data which correlates the gain and the polar angle. However, in a document of a former antenna model called DW1000 [17], DecaWave gives the gain values (reported in Tab. 1) for an antenna vertically placed (as in Fig. 4(a)) in an anechoic chamber. We report these values just to confirm the not negligible impact of the elevation¹: the 3D radiation pattern is far from being a sphere, with the same gain in all the directions.

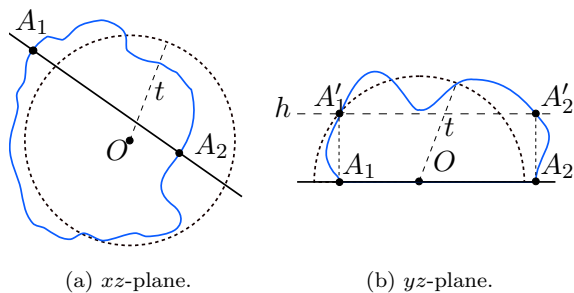


Figure 5: The ideal (dashed) and actual (solid) antenna radiation profile in xz -plane and yz -plane.

Then, we conjecture that, when the MA is a drone, the 3D antenna pattern is highly irregular and it can be sketched as a nibbled apple (Fig. 5). Therefore, when such an antenna shape is projected on the ground, holes and bubbles can be found. We speculate that the altitude is the main cause of the pattern irregularity. So, we conjecture that when the two antennas are both on the ground (rover as MA) in VV configuration, the gain is the same in almost all the directions.

5.3. Experiments for Antenna Radius

In our experiments, we wish to characterize the 2D antenna pattern observing the range of values of its radius. The MA starts at HOME in $(0,0)$ at different altitudes $h = \{0, 10, 20\}$ m. As explained in Sec. 4, the MA traverses the deployment area with n segments that aim to cross the receiving shape of the GD. Moving along each random segment, the MA continuously broadcasts its current (x, y) position, and the GD registers the first and the last heard endpoints sent by the MA. Since we know in advance the HOME position of the GD, we can compute the actual 2D radius for each detected endpoint. Note that we observe the radius on the ground. It is important to recall that the beacons are sent at regular intervals of time, and so the observed radii have an intrinsic error of at most I_w . In this paper, we fix I_w to 0.40 m since we have experimentally observed this

¹Note also that the 3D antenna pattern is completely defined if we know its behavior in 3 planes: xz , xy , and yz .

value, which clearly depends on the speed of the MA. From the collected endpoints during the same experiment, we compute the mean μ and the standard deviation σ of the set of observed radii. Then, we apply the goodness-of-fit method in order to assess whether a given distribution is suitable to the built data-set². We repeat the experiments for the two antenna configurations (VV and VH) and for different altitudes.

We start reporting in Tab. 2 the results of the first experiment with the rover (i.e., $h = 0$ m) and VV configuration. According to DecaWave’s datasheet (solid line in Fig. 4(a)), we expect an almost uniform radius in the experiments, at least when the two antennas are at the same height. We observed 38 different endpoints, with mean $\mu = 97.10$ m and $\sigma = 39.74$ m, and also with $\min = 32.14$ m and $\max = 162.40$ m. According to the Pearson χ^2 -test, at $h = 0$ m, the radii of the VV configuration have a uniform distribution. As we will see, this is the only configuration with uniform distribution of the radii. Therefore, we agree with DecaWave that this configuration is somehow special. However, the radius cannot be considered really constant.

Table 2: $h = 0$ m, VV

class #	radii (in m)		frequencies		
	from	to	observed	uniform (U)	normal (N)
1	32.14	58.34	8	0.08	3.14
2	58.34	84.54	7	0.01	0.13
3	84.54	110.74	8	0.08	0.34
4	110.74	136.94	5	0.69	1.06
5	136.94	163.14	10	1.06	8.15
likelihood				0.75	0.01

Then, in Tab. 3 we report a second experiment at altitude $h = 10$ m, with VH configuration. We observed 28 different endpoints out of 38, thus confirming several null angles. The observed radii have mean $\mu = 66.34$ m, $\sigma = 22.81$ m, $\min = 16.52$ m, and $\max = 121.66$ m. According to the Pearson χ^2 -test, at $h = 10$ m, the radii of the VH configuration have a normal distribution.

Table 3: $h = 10$ m, VH

class #	radii (in m)		frequencies		
	from	to	observed	uniform (U)	normal (N)
1	16.52	41.52	5	1.68	0.68
2	41.52	66.52	9	0.00	0.14
3	66.52	91.52	11	0.52	0.07
4	91.52	141.52	3	12.23	0.15
likelihood				0.01	0.90

Finally, in Tab. 4 we summarize the statistic distributions that fit the observed experimental radii. For each distribution, i.e., Uniform (U) and Normal (N), we give the observed μ and σ , and the likelihood. Except for the rover in VV, all the experiments show that the radius most likely follows a normal distribution, but with a large σ . It is worthy to note that increasing h , the mean

²Fixed a distribution and a set of categories, we determine if there is a significant difference between the expected and observed frequencies in one or more categories by using the chi-squared test.

of the radii decreases. The mean decreases faster with VV while with VH it remains quite stable (see Tab. 4). The values of the radii are generally more concentrate with VH than VV.

Table 4: The radii (in m) distribution with its parameters $D(\mu, \sigma)$ and its likelihood p .

	VV	VH
$h = 0$ m	$U(97.10, 39.74); 0.75$	$N(63.58, 33.01); 0.40$
$h = 10$ m	$N(84.97, 31.70); 0.81$	$N(66.34, 22.81); 0.90$
$h = 20$ m	$N(62.91, 34.06); 0.83$	$N(57.69, 24.91); 0.82$

We conclude that, oppositely to our conjecture, VH seems better than VV and the radii obtained with a drone seem more concentrated than those obtained with a rover. Marginally, let us point out that organizing a localization mission is easier with a drone than with a rover because the drone is faster and less attention has to be paid to the terrain. Although the results are different from what we expected, we continue our investigation in localization algorithms accuracy. Thus, we use the results reported in Tab. 4 to generate a large synthetic set of endpoints that fit the estimated parameters of the radii distributions for testing the different algorithms surveyed in Sec. 3.

From now on, we refer to the radius reported in Tab. 4 as the *observed radius* $r = \mu$, while to the DecaWave declared radius as the *manufacturer radius* $r = r_0$.

6. Range-free Comparative Evaluation

In this section, we compare all the RF localization algorithms using first the set of synthetic endpoints, and then the set of real endpoints collected during the experiments.

Our goal is to analyze the localization error and the percentage of unsuccessful localizations of DRF, XIAO, LEE, and DRFE. From now on, with height $h = \{0, 10, 20\}$ m and antenna configurations $\{VV, VH\}$ we refer to a particular scenario. For each simulated scenario, we run 200 localizations generating at random three endpoints, B_1 , B_2 , and B_3 , with the distribution and the parameters of the simulated scenario derived in Sec. 5.3 from the observed endpoints (see Tab. 4). For each triple, we invoke the four algorithms using either the observed radius $r = \mu$ used to generate the endpoints or the manufacturer radius $r = r_0$. In the former case, we test the performance of algorithms when they receive in input the actual radius, but still, the endpoints can be affected by the antenna irregularity (i.e., σ). In the latter case, we test the performance of algorithms when they receive in input a completely different radius from the actual one.

Reinterpreting the constraints to improve the accuracy given in [20], the three selected endpoints, i.e., B_1 , B_2 , and B_3 we use for localizing the GD should satisfy two constraints: the minimum distance $r_{\min} = 60$ m and the minimum angle $\alpha_{\min} = 20$ deg between them. The constraint r_{\min} means that the distances $d(B_1, B_2)$, $d(B_2, B_3)$, and $d(B_3, B_1)$, must be at least as long as r_{\min} . The α_{\min} constraint means that the three angles $\alpha_1 = \angle B_3 B_1 B_2$, $\alpha_2 = \angle B_1 B_2 B_3$, and $\alpha_3 = \angle B_2 B_3 B_1$, must be at least as large as α_{\min} . From a geometrical point of view, these two constraints guarantee that the three selected endpoints are sufficiently apart each other, thus avoiding the construction of degenerated triangles. Therefore, in the experiments, we discard any triple of endpoints that does not satisfy $r_{\min} = 60$ m and $\alpha_{\min} = 20$ deg. We repeat the endpoint extraction until we find three suitable endpoints.

We compare the RF algorithms under two metrics; the *localization error*, defined as the Euclidean distance between the actual GD's position and the estimated one outputted by the algorithms, and the *percentage of unlocalized*. Concerning the first metric, we report the localization

error resumed into a boxplot that highlights the median (horizontal line), the average value (solid circle), and the data between the first Q_1 and the third quartile Q_3 (box). Additionally, the extremes of the whiskers represent the $Q_1 - 1.5 \text{ IQR}$ and $Q_3 + 1.5 \text{ IQR}$, respectively, where the interquartile range (IQR) is defined as $\text{IQR} = Q_3 - Q_1$. Lastly, about the second metric, an unsuccessful localization is an application of the algorithm which does not return any constrained area or in general any geometrical intersection where the GD can reside, i.e., the GD remains unlocalized. This mainly happens for the RF radius-based algorithms when the radius is under-estimated.

6.1. Algorithms Comparison Results

In this section, we evaluate the performance of the RF algorithms. We start considering a synthetic set of endpoints with an average radius equal to the observed radius μ , but very small standard deviation, emulating an ideal model. This experiment is to support the observation that a high accuracy is possible when the antenna is almost isotropic. Moreover, we discuss the performance of the RF algorithms on the synthetic set of endpoints generated according to the observed distributions in Tab. 4. These experiments, as those that use the manufacturer radius, show poor accuracy. In all the experiments, the impact of using a rover or a drone is considered. Finally, we report the comparison between the performance of DRF and DRFE.

Under the Ideal Model. Let us start considering the nearly ideal model in Fig. 6 in which the endpoints are generated with the observed radius μ given in Tab. 4 for the VV configuration, but selecting σ equal to 1. We evaluate the algorithms simulating an almost omnidirectional antenna. Since the dispersion is low, the radius can be considered almost constant. As expected, all the radius-based algorithms perform well when they use the observed radius $r = \mu$, which is the same radius used to generate the endpoints. In such a case, the localization error is on the order of a couple of meters or less for all the algorithms³. DRF is slightly better than the other algorithms at any altitude, while DRFE is worse than DRF. The number of unlocalized GDs is very small. However, the performance of the radius-based algorithms drastically drops down when the algorithms run using the manufacturer radius $r_0 = 60 \text{ m}$, while the endpoints have been generated with the observed radius. The percentage of unsuccessful localizations is extremely large.

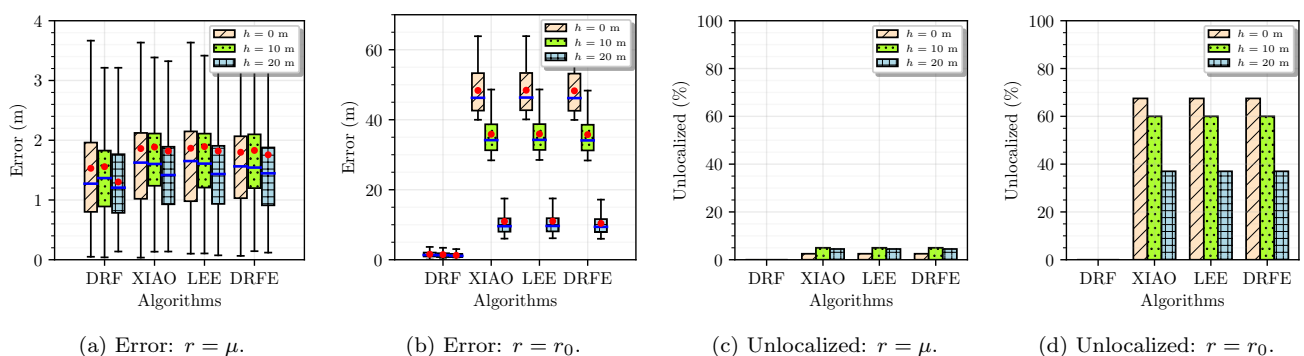


Figure 6: The impact of the radius on the ideal model.

These results confirm what we said in Sec. 3: to obtain an accurate localization, not only the antenna must be of good quality (i.e., with small σ), but also the observed radius μ must be exactly

³Please note that the scale of y axis in Fig. 6(a) is zoomed with respect to the scale of y axis in Fig. 6(b)

known by the algorithm. The results show that the error due to the use of the radius r_0 decreases when h increases because it decreases the difference between the observed μ and the manufacturer r_0 radii. Alongside, note that since DRF is radius-free, the performance of DRF is not influenced by the radius selection, as shown in Fig. 6(b). DRF only requires an omnidirectional antenna.

Using the Synthetic Endpoints Set. Figs. 7(a), 7(b), 7(e), and 7(f) show the results when both the endpoints and the algorithms use the observed radius in Tab. 4. Figs. 7(c), 7(d), 7(g), and 7(h) instead show the results when the endpoints are generated using the observed radius and the algorithms run with the manufacturer radius. This is what happens in practice whenever the end-user implements the RF radius-based algorithms using the DecaWave datasheet radius on our test-bed. The errors of the algorithms are compared at different altitudes, antenna configurations, and radii. For all the algorithms, the average error is large. The worst error occurs with the VV configuration of the antennas and $h = 0$ m: in such a scenario, the endpoints follow a uniform distribution. In general, the VH makes an error smaller than the VV probably because the radius that generated the endpoints for VH is less dispersed (i.e., σ is smaller) than that for VV. The three radius-based algorithms, XIAO, LEE, and DRFE, exhibit the same performance. They are inspired by slightly different ideas, but they actually act the same.

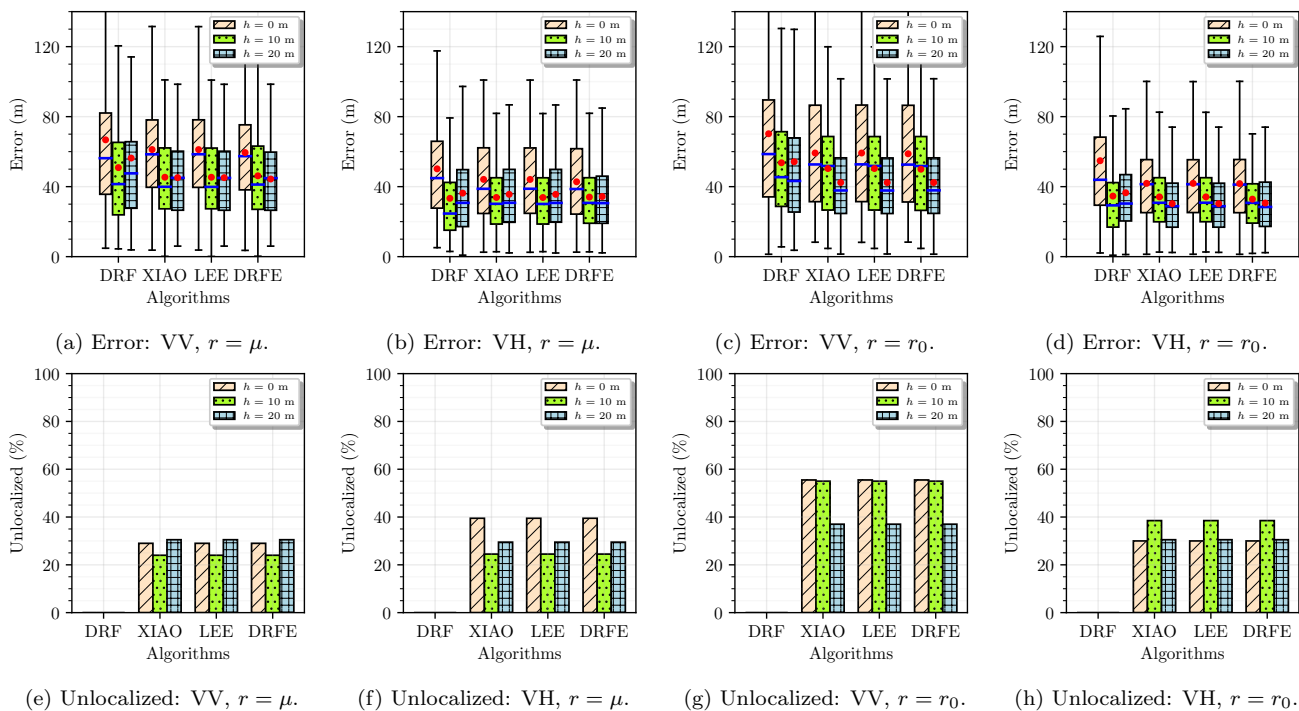


Figure 7: Comparisons between all the algorithms in the studied scenarios.

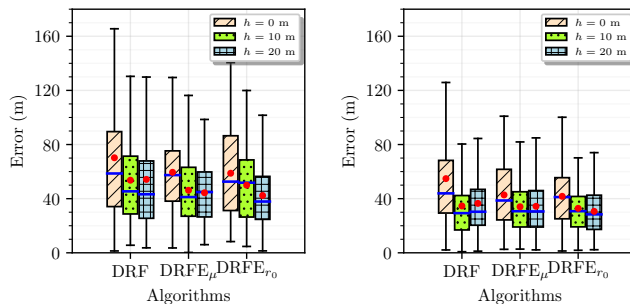
The DRF algorithm experiences the worst average error, but the error is only slightly more than that of the radius-based algorithms. The whisker of the largest error of DRF is the longest whisker among all the algorithms probably because DRF finds a localization also in extreme cases when other algorithms return an unsuccessful localization. The error in DRF decreases when the altitude increases, moreover the error of the VV configuration is worse than that of the VH one.

The localization error is almost the same regardless of the adopted radius (observed or man-

ufacturer). As witnessed by comparing the whiskers of the boxplots in Figs. 7(a) and 7(c) (resp., by Figs. 7(b) and 7(d)), the localization error of the radius-based algorithms, when they use the manufacturer radius r_0 , is slightly worse than when they use the observed radius μ . Instead, the knowledge of μ reduces percentage of unlocalized GDs as illustrated in Figs. 7(e) and 7(g) for VV. The improvement in Figs. 7(f) and 7(h) is weaker for VH than for VV because for VH the difference between the observed radii $\mu = \{63.58, 66.34, 57.60\}$ m and $r_0 = 60$ m is smaller than VV.

In conclusion, the average localization error for VV and VH is 50 m and 30 m, respectively, regardless of if the algorithm knows the true radius used to generate the endpoints or not. The knowledge of the radius used to generate the endpoints by the algorithm only matters when the radius dispersion is small. The take-away lesson here is that to apply a RF radius-based algorithm the antenna must be omnidirectional and its radius must be known by the algorithm.

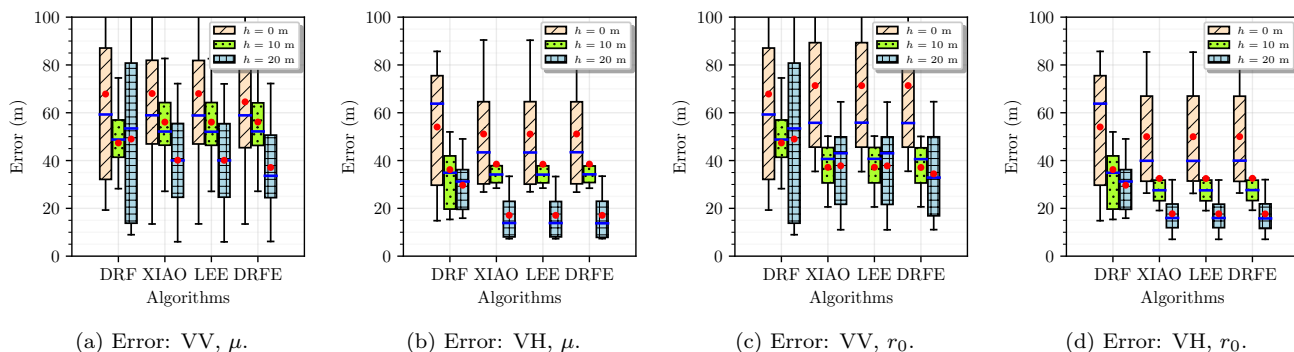
Between DRF and DRFE. In Fig. 8, we compare the performance of DRF, DRFE_μ (DRFE with radius μ), and DRFE_{r_0} (DRFE with radius r_0). In general, DRFE_μ performs better than DRFE_{r_0} , but DRFE_μ has always an error much greater than the error of the ideal model (see Fig. 6).



(a) Error: VV, DRF vs DRFE. (b) Error: VH, DRF vs DRFE.

Figure 8: DRF vs DRFE.

Using Real Endpoints. In Fig. 9, we show the results using the real endpoints collected during our test-bed. For the radius-based algorithms, in VV, the results of the localization error seem to follow



(a) Error: VV, μ .

(b) Error: VH, μ .

(c) Error: VV, r_0 .

(d) Error: VH, r_0 .

Figure 9: Error using real data.

the trend already seen on the synthetic generated set. In general, VV has a larger error compared to that of VH. In VH, the results are slightly better than that obtained on the synthetic generated

set, although the number of valid triples is small. Indeed, the average error at $h = 20$ m is about 15 m although σ is greater than 20 m when the radius μ is used. The average error increases to 20 m when the radius r_0 is used. This can be explained with the fact that there is a much stronger correlation between the endpoints that are not fully captured by the two constraints r_{\min} and α_{\min} that we imposed for the selection of the synthetic endpoints. Finally, the error of DRF is worse with real endpoints than with synthetic endpoints, especially for the VV configuration.

So far, we learned the performance of all the RF algorithms strongly relies on the radiation antenna pattern. The simplest way to discover the radiation antenna pattern is to measure for each direction up to which distance the MA and the GD are one in the range of the other. So, in the next section, we significantly improve the localization accuracy exploiting the capability of the UWB antennas used in our test-bed of taking distance measurements. Even though there are bubbles and holes in the antenna pattern, it is always possible to discover them if the distance from the GD and the MA is taken (see Fig. 10).

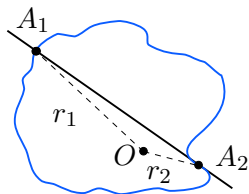


Figure 10: The two radii measure by the GD when the antenna pattern is irregular.

7. Range-based Comparative Evaluation

Our interest in RF algorithms lay in the fact that they are simple to implement, do not require specialized hardware, and are scalable with respect to the number of GDs. Also, they are immune to problems that come from the measurement of the 3D distance. Indeed, distance measurements are affected by several errors that depend on the adopted technology and GPS. Technologies like WiFi or Bluetooth are much less accurate than UWB and may have measurement errors of the order of tens of meters, whereas GPS and barometer inaccuracies together with bad weather conditions can seriously impact the drone's position. Such combined 3D slant errors are then reflected on the ground, leading to 2D ground errors. Although range measurements come with their troubles, given the results of the previous section, we decided to include distance measurements in our algorithms. Since in our test-bed both the MA and GD are equipped with UWB antennas that are able to take distance measurements, we do not need to heavily modify our test-bed.

In the algorithm variants that we are going to propose, the actual measured distance between MA and GD is used to run the algorithms instead of using the observed or manufacturer radius. This adaption makes the RF algorithms actually RB algorithms, but we maintain the original algorithmic rules (e.g., intersection of annuli or circumferences). It is worth noting that we can apply modifications only to the RF algorithms that actually make use of the radius in their localization rules, and hence only for the radius-based ones, i.e., XIAO, LEE, and DRFE. Accordingly, the simplest radius-free algorithm DRF cannot be adapted to measurements.

Range-based version of XIAO. Fig. 11(a) shows how the XIAO algorithm is modified. When the GD hears for the first time the beacon in A_1 , the GD measures the distance $r'_1 = d(A_1, O)$ between its position O and A_1 . Whenever the MA is inside the receiving area of the GD, the GD still takes

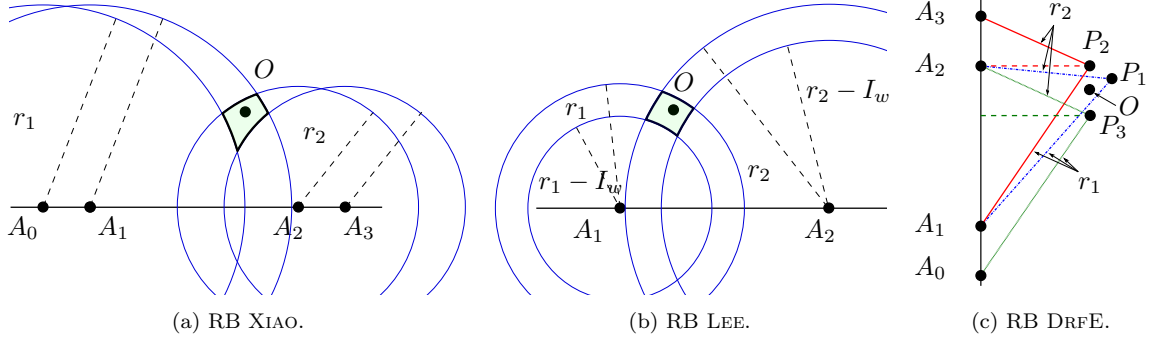


Figure 11: The XIAO, LEE, and DRFE localization algorithms using two radii.

distance measurements neglecting the intermediate measurements. When the last beacon A_2 is heard, the GD saves the last distance $r'_2 = d(A_2, O)$ between its position O and A_2 . As in the original version of XIAO, the GD computes the other two points A_0 and A_3 knowing the line that interconnects A_1 and A_2 and the value of I_w . Then, the algorithm proceeds as before, except that it draws two circumferences of radius $r_1 = r'_1 + \frac{I_w}{2}$ centered in A_0 and A_1 , and two circumferences of radius $r_2 = r'_2 + \frac{I_w}{2}$ centered in A_2 and A_3 . That is, the predefined observed or manufacturer radii are substituted by the distances from the endpoints and the GD. Note that the radius r_1 and r_2 are used instead of $r'_1 = d(A_1, O)$ and $r'_2 = d(A_2, O)$ to avoid that O falls outside the intersection area in case of measurement error that underestimated the distances (see Fig. 11(a)).

Range-based version of LEE. As seen for XIAO, also for LEE, the two distances r'_1 and r'_2 between the endpoints and the GD are ranged. The algorithm proceeds by drawing two annuli centered at A_1 and A_2 , with outer radius equal to $r_1 = r'_1 + \frac{I_w}{2}$ and $r_2 = r'_2 + \frac{I_w}{2}$, and inner radius equal to $r_1 - I_w$ and $r_2 - I_w$, respectively. The main difference with the previous version is that the radii are replaced by the distances between the GD and the endpoints. As in XIAO, we use the radii $r_1 = r'_1 + \frac{I_w}{2}$ and $r_2 = r'_2 + \frac{I_w}{2}$ instead of r'_1 or r'_2 to limit the risk that, due to measurement inaccuracy, O falls outside the intersection area. The width I_w of the annuli is preserved.

Range-based version of DRFE. We replace r with $r_1 = r'_1 = d(A_1, O)$ and $r_2 = r'_2 = d(A_2, O)$. The intersection of the two circles of radius r_1 and r_2 centered, respectively, at A_1 and A_2 , returns P_1 which coincides with O (assuming no measurement errors). To repeat the construction of the RF DRFE, the points P_3 and P_2 are drawn in a way similarly to P_1 . Precisely, P_3 is at the intersection between a circumference of radius r_1 centered in A_0 and a circumference of radius r_2 centered in A_2 . P_2 is at the intersection between a circumference of radius r_1 centered in A_1 and a circumference of radius r_2 centered in A_3 . As a result, differently from the original version where there were three isosceles triangles lying on the same line passing through A_0 and A_3 , here there are three scalene triangles, i.e., $\triangle(A_1A_2P_1)$, $\triangle(A_1A_3P_2)$, and $\triangle(A_0A_2P_3)$. Eventually, the centroid resulting from the vertices P_1 , P_2 , and P_3 is selected as the estimation point, as shown in Fig. 11(c).

7.1. Error Analysis with Different Radii

In this section, we argue concerning the localization error that can be obtained once the original RF versions of the algorithms have been adapted to be actually RB. We focus on XIAO, but a very similar analysis applies to LEE. Some words will be finally spent for DRFE.

Without loss of generality, we assume that the MA moves along a straight line along the x -axis (see Fig. 11(a)). Let $O = (x_O, y_O)$ be the actual GD's location with respect to Cartesian coordinate system with origin in $A_0 = (0, 0)$, and let $P = (x_P, y_P)$ be the estimated GD's location by the algorithm. We fix A_1 and A_2 so as $r_1 = d(A_1, O) \leq d(A_2, O) = r_2$ and let $A_1 = (I_w, 0)$ and $A_2 = (kI_w, 0)$, where k is an integer number that represents the number of times the MA has sent the beacons from A_1 to A_2 . Finally, let $A_3 = ((k+1)I_w, 0)$.

The RB XIAO relies on two different radii, i.e., r_1 applied to A_0 and A_1 , and r_2 applied to A_2 and A_3 , for constructing four circumferences, i.e., (a) $x^2 + y^2 > r_1^2$, centered in A_0 , (b) $(x - I_w)^2 + y^2 \leq r_1^2$, centered in A_1 , (c) $(x - kI_w)^2 + y^2 > r_2^2$, centered in A_2 , and (d) $(x - (k+1)I_w)^2 + y^2 \leq r_2^2$, centered in A_3 . We also denote as L_1 the *lune*⁴ delimited by Eqs. (a)–(b), and as L_2 the lune delimited by Eqs. (c)–(d), also in accordance with the original XIAO algorithm.

Note that r_1 and r_2 are affected by the measurement errors [20] and for this reason we do not directly conclude that the estimated position P is at the intersection of Eqs. (b)–(c). Moreover, the quantity $kI_w = d(A_1, A_2)$ is influenced by the MA's speed, and hence it cannot be considered exact as well. Therefore, we prefer to select P inside the intersection among L_1 and L_2 .

The correctness of the RB XIAO algorithm is completely different from that of the original XIAO. Although our implementation only memorizes the first and the last measured radius, i.e., r_1 and r_2 , the RB XIAO algorithm performs measuring, at each i^{th} beacon $A_i = (iI_w, 0)$, the current distance $r_2(i) = d(A_i, O)$. Therefore, for each intermediate beacon A_i and corresponding measurement $r_2(i)$, with $1 \leq i \leq k$, RB XIAO knows that $O \in (L_1 \cap L_2(i))$, i.e., where $L_2(i)$ is the lune created by the two circumferences of radius $r_2(i)$ centered in A_i and A_{i+1} , respectively. Hence, although we do not implement this feature in our algorithm, after each measurement $r_2(i)$ RB XIAO could stop. As we will see, the best moment to stop would be when $r_2(i)$ is minimum. This is the main reason that makes RB XIAO robust to hole and bubbles: whenever it stops, the intersection area is limited. However, a discussion about the intersection of lunes while A_i varies along the x -axis is needed to complete the error analysis.

Assuming an omnidirectional antenna pattern, as MA moves along the x -axis, it crosses the GD's receiving area disk. The distances $d(A_i, O)$ decrease until MA reaches the closest position to O in $A_{k^*} = (k^*I_w, 0)$ (precisely, $x_O \leq k^*I_w < x_O + I_w$). Then, $d(A_{k^*}, O) \approx y_O$. After A_{k^*} , the distances $d(A_i, O)$ start to increase again up to $A_3 = ((k+1)I_w, 0)$, where the MA is no longer reachable from the GD, and whose last measured distance taken from $A_2 = (kI_w, 0)$ is $d(A_2, O) = r_2$. The observed distances while MA moves in A_i , with $i = 1, \dots, k^*, \dots, k$, form a unimodal sequence R with minimum in k^*I_w . As long as the radii in R decrease, i.e., $x_{A_i} < x_{A_{k^*}}$, the lunes L_1 and $L_2(i)$ have the same *curvature*⁵ (see Fig. 12(a)), while when the radii in R increase, i.e., $x_{A_i} > x_{A_{k^*}}$, L_1 and $L_2(i)$ have opposite curvature (see Fig. 12(c)). In A_{k^*} , the tangent to the circumferences of the lune $L_2(k^*)$ is parallel to the x -axis (see Fig. 12(b)).

In a general scenario in presence of holes and bubbles, the MA stops at A_2 in one of the three cases depicted in Fig. 12: (i) the curvatures of A_1 and A_2 are concordant, i.e., $1 < k < k^*$ or $k^* < 1 < k$; (ii) the tangent at A_2 is parallel to x -axis, i.e., $1 < k = k^*$; (iii) the curvature A_1 and A_2 are discordant, i.e., $1 < k^* < k$. To evaluate the size of the intersection among L_1 and L_2 in these three cases, let observe that they cross each other making at most four intersections, i.e., A is the intersection among Eqs. (b)–(c), B among Eqs. (b)–(d), C among Eqs. (a)–(d), and D among Eqs. (a)–(c) (see Fig. 12), where:

⁴In plane geometry, a lune is the concave-convex region bounded by two circular arcs.

⁵The curvature is the sign of the tangent to the curve.

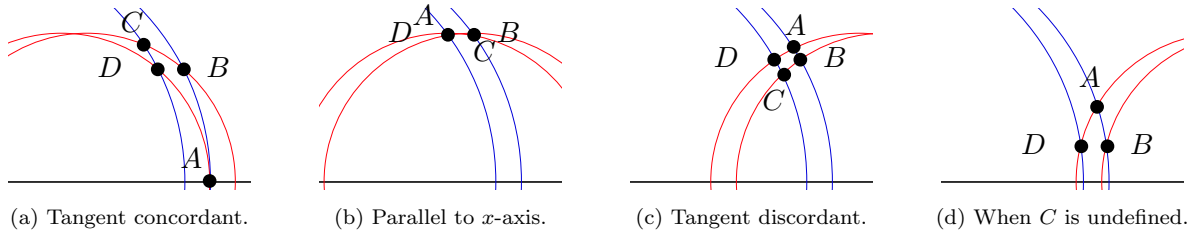


Figure 12: The possible intersection areas depending on the lunes: the solid line is the x -axis.

$$\begin{aligned}
 A &= \left(\frac{r_1^2 - r_2^2}{2(k-1)I_w} + \frac{k+1}{2}I_w, \sqrt{r_1^2 - \left(\frac{r_1^2 - r_2^2 + (k-1)^2 I_w^2}{2(k-1)I_w} \right)^2} \right) \\
 B &= \left(\frac{r_1^2 - r_2^2}{2kI_w} + \frac{k+2}{2}I_w, \sqrt{r_1^2 - \left(\frac{r_1^2 - r_2^2 + k^2 I_w^2}{2kI_w} \right)^2} \right) \\
 C &= \left(\frac{r_1^2 - r_2^2}{2(k+1)I_w} + \frac{k+1}{2}I_w, \sqrt{r_1^2 - \left(\frac{r_1^2 - r_2^2 + (k+1)^2 I_w^2}{2(k+1)I_w} \right)^2} \right) \\
 D &= \left(\frac{r_1^2 - r_2^2}{2kI_w} + \frac{k}{2}I_w, \sqrt{r_1^2 - \left(\frac{r_1^2 - r_2^2 + k^2 I_w^2}{2kI_w} \right)^2} \right)
 \end{aligned}$$

Note that A is stable when A_2 moves because it is at the intersection of the two measurements of O . If there were no errors, $O \equiv A$. It is worth noting that $y_B = y_D$ and $x_B - x_D = I_w = d(D, B)$ regardless of k . Now we are in the position of clarifying the three previous cases. Recalling that $A_1 = (I_w, 0)$, depending on which is the last heard beacon $A_2 = (kI_w, 0)$, it may occur:

Case (i): The lunes have the same curvature, as illustrated in Fig. 12(a). The area where P can be selected has width I_w and height $y_C - y_A$. In particular, if k is very small, $r_2 \rightarrow r_1$, and thus $y_C \rightarrow r_1$. If $y_A \rightarrow 0$, the error $y_C - y_A \approx r_1$. As k approaches k^* , $y_C - y_A$ quickly decreases. A similar error occurs when $k^* < 1 < k$.

Case (ii): By intersecting the coordinates $x_C = x_A$, and all the four points A , B , C , and D are very close. The area where P can be selected is very small.

Case (iii): Observe that x_C can be re-written as $x_C \approx (x_A + I_w) \frac{k-1}{k+1}$ for $k \geq 2$. Thus, $x_C < x_A + I_w$ for any k . Due to this and the fact that the curvatures of L_1 and L_2 are opposite, their intersection area is quite limited. Unfortunately, we could not find a simple formula for describing $|y_A - y_C|$ here, but it can be computed by approximating the curves with their tangent in A . Hence, the intersection of the two lunes can be inscribed in a rectangle with two sides parallel to the x -axis of length I_w and two vertical sides whose length depends on the angular coefficient of the tangents in A . We can only add that their lengths decrease below $2I_w$ when $\arctan \frac{y_A}{x_A} > 1$. So, the area where P can be selected only depends on I_w . When $\arctan \frac{y_A}{x_A} \leq 1$, the trivial bound is y_A , which cannot be very large however. So, the size of the area where P can be selected depends on I_w and y_A . When $r_1 \leq x_A < r_1 + I_w$, C is undefined (see Fig. 12(d)). Hence, the vertical side of the rectangle that contain P has length at most $y_A \leq \sqrt{r_1^2 - (r_1 - I_w)^2}$. This leads to the same

error described in XIAO when the lunes have only 3 intersection points [10].

In presence of irregularities, RB XIAO, as in our implementation, stops at the last heard beacon. The error can be large only when A_1 and A_2 fall on the same side with respect to x_O , and thus the associated lunes have concordant curvatures. In order to limit the occurrences of Case (i), in our implementation we have forced B_1 , B_2 , and B_3 , i.e., the three selected endpoints, to respect the r_{\min} and α_{\min} constraints, putting them sufficiently apart, so as the two lunes will have opposite curvature. Moreover, in Cases (ii) and (iii) the error is quite limited and only depends on I_w . Without irregularities, RB XIAO always falls in Cases (ii) or (iii). The same happens for the original XIAO which stops when $r_1 = r_2$ and thus the curvatures are opposite.

To conclude, the above analysis also applies for LEE. In LEE, the GD belongs to the intersection of two annuli of different radii but with the same width I_w , and the error depends on the curvature of the lunes created by the intersection of annuli. As regard to DRFE, P_1 , P_2 , and P_3 are the intersections of pair of circumferences, and here the lunes coincide with portions of circumferences. Differently from XIAO and LEE, the position uncertainty depends only on the measurement errors and on I_w , and it is bounded according to [20].

7.2. Range-based Versions Results

In this section, we evaluate the performance of the RB variants, and we compare these results with those of the RF original implementation.

In reality, in our previous experiments, since the goal was to investigate the performance of RF algorithms, we did not take any distance measurement even though our UWB antennas were able to range measurements with good accuracy. Indeed, we only knew in advance the GD's HOME position (0,0) and the MA's position with respect to HOME. Instead, the new RB variants rely on the 3D slant distances between the MA and the GD, which are then converted in 2D ground distances. However, it is important to recall that, due to the pandemic COVID-19, it is forbidden to take new distance measurements in the open field. According to those restrictions and based on our previous research experience, we decided to estimate the 2D ground distances (see Fig. 3(a)) between the MA and the GD just calculating the Euclidean distance from the endpoints to the HOME in (0,0) plus a random error (overestimation or underestimation) computed as proposed in [20]⁶. In other words, we perturb any Euclidean distance r adding the error E_r :

$$E_r = \gamma_d + \frac{h}{r}\gamma_h + e_s \cdot \sqrt{1 + \frac{h^2}{r^2}} = 1.2 + \frac{h}{r}0.2 + e_s \cdot \sqrt{1 + \frac{h^2}{r^2}} \quad (1)$$

where h is the drone's altitude, $\gamma_d = 1.2$ m is the drone's rolling error, $\gamma_h = 0.2$ is the drone's altitude error, and e_s is a random number in the range $[-\epsilon_s, +\epsilon_s]$, where $\epsilon_s = 0.1$ m is the UWB instrumental accuracy [20]. From Eq. (1), when the ground distance $r = 30$ m, the maximum ground error E_r is approximately 1.3 m, 1.4 m, and 1.5 m, for h is 0 m, 10 m, and 20 m, respectively. First, we compare the accuracy of the new RB variant algorithms toward that of their RF implementation. Then, we make remarks on the accuracy of the new RB variants of the three RB algorithms.

Range-free vs Range-based. In Tab. 5 we report the localization accuracy of XIAO, LEE, and DRFE showing the comparison between the original RF and adapted RB versions of them. Tab. 5 also reports the number of times (in %) that an algorithm does not localize the GD. The experimental

⁶Our decision is supported by our previous experience in converting slant from/to ground measurements [20, 21].

results take into account the constraints $r_{\min} = 60$ m and $\alpha_{\min} = 20$ deg, and only VV. In particular, for the original RF versions we report the average error for both the radii, i.e., the observed radius $r = \mu$ and the manufacturer radius $r = r_0$. For the adapted RB versions we report the average error for the measured radius, denoted as $r = d(A_i, O)$, where A_i is an endpoint.

Table 5: Error (in m) and unlocalized (in %) between RF and RB algorithms in VV.

Algorithm	h	RF				RB	
		$r = \mu$ error	unloc	$r = r_0$ error	unloc	$r = d(A_i, O)$ error	unloc
XIAO	0 m	68	30%	72	55%	4	0%
	10 m	56	25%	38	54%	4	0%
	20 m	40	31%	39	37%	3	0%
LEE	0 m	67	30%	73	55%	3.8	0%
	10 m	55	25%	38	53%	3.8	0%
	20 m	40	32%	38	37%	2.7	0%
DRFE	0 m	64	29%	74	55%	3.2	0%
	10 m	55	26%	37	53%	3.2	0%
	20 m	37	31%	33	36%	2.2	0%

It is clear that the average error, using measurements, is one order of magnitude smaller. We have seen that the knowledge of the exact distance significantly improves the accuracy and practically clear the number of unlocalized devices. This also enforces our original statement that the knowledge of the radiation pattern is fundamental for the success of the localization. Bubbles and holes in the antenna radiation pattern are no longer a problem because we range the effective distances between the endpoints and the GD. So to avoid measurements, any manufacturer should give as much information as possible on the antenna radiation pattern as possible.

Range-based Comparison. Here, we present performance-wise the RB versions of the XIAO, LEE, and DRFE algorithms. As previously said, DRF is not present because it is not radius-based. For this simulation evaluation, we consider different cases varying the constraints of minimum distance $r_{\min} = \{0, 60\}$ m and minimum angle $\alpha_{\min} = \{20, 40\}$ deg, as explained in Sec. 6.

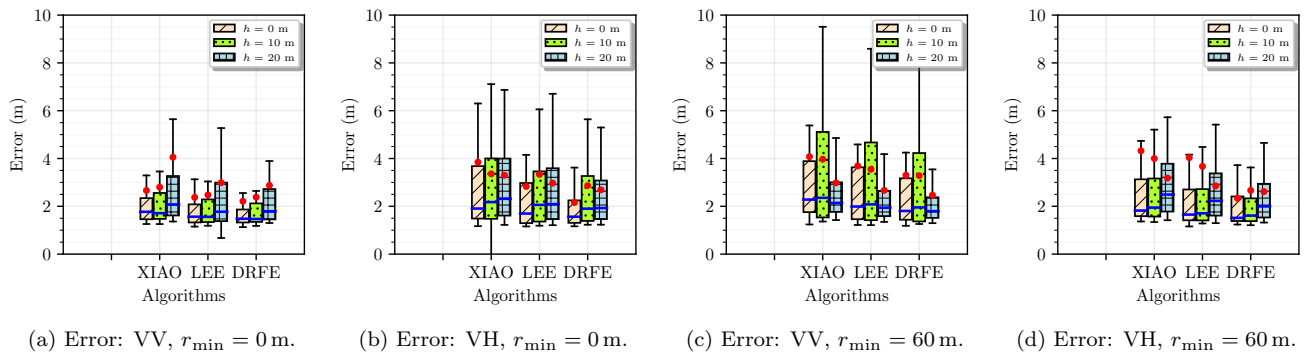


Figure 13: Error using distance measurements when $\alpha_{\min} = 20$ deg and r_{\min} varies.

In Fig. 13 and Fig. 14, we evaluate the performance of the RB algorithms. In particular, in Fig. 13 we compare the algorithms fixing the minimum angle $\alpha_{\min} = 20$ deg, while in Fig. 14 such

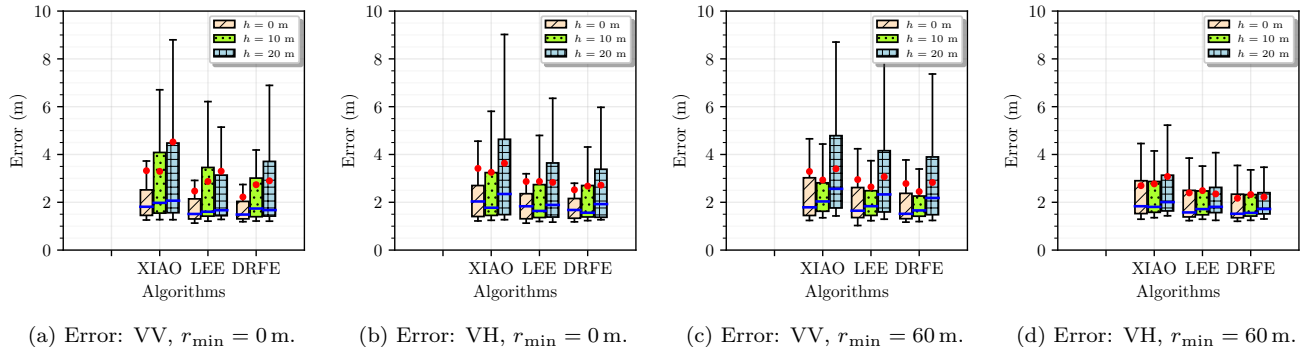


Figure 14: Error using distance measurements when $\alpha_{\min} = 40$ deg and r_{\min} varies.

constraint is kept to $\alpha_{\min} = 40$ deg. Overall, we can see that DRFE works better than the other two algorithms with an average error smaller across all the tests, and that XIAO works better than LEE with those settings. Also, a better localization accuracy can be obtained for the experiments carried on the ground at $h = 0$ m, exception made for Fig. 13(c) in which, oddly, the error is smaller at $h = 20$ m. More specifically, in both Fig. 13 and Fig. 14 is also evident that VH benefits slightly more from the r_{\min} constraint of 60 m, while it is the opposite for VV which behaves better showing smaller error with no r_{\min} constraint. Finally, there is very little improvement while testing with larger α_{\min} . It is extremely important to note that using measurements, all the selected endpoints are suitable for localizing the GD in \mathcal{O} . This means that the number of unlocalized GDs is zero, but also this leads to the growth of the outliers number. This can be seen by the presence of long whiskers and a considerable difference between the average (i.e., solid circle) and median (i.e., horizontal line) values. However, even though such whiskers are pretty long, they are really short with respect to the ones seen in the previous experiments in Sec. 6, Fig. 9.

8. Conclusion

In this paper, we compared the performance of four RF algorithms based on HnH on a real test-bed using the DecaWave DWM1001 UWB antennas as MA and GDs. We implemented and simulated the algorithms on a large data-set of endpoints collected in the field. We analyzed the antenna radiation pattern of the GD at different altitudes and configurations, i.e., VV and VH. We have shown how such algorithms actually perform assuming (i) first the datasheet radius of the antenna equal to that released by the manufacturer, (ii) then the experimental observed radius, and (iii) finally the actual radius obtained via distance measurements. The manufacturer datasheet radius poorly performs because usually it does not characterize the antenna radiation pattern very well. The observed radius can help only if it is almost constant in all the directions. When the antenna is irregular, only the knowledge of the distances between the MA and the GD can alleviate the localization error. We conclude that the RF algorithms are simple and elegant, but they can be very inaccurate and can leave a high percentage of unlocalized GDs if the antenna is not omnidirectional and measurements are not allowed. However, the exact knowledge of the irregular antenna can make accurate the RF algorithms.

References

- [1] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, G. K. Karag., A survey on mobile anchor node assisted localization in wireless sensor networks, *IEEE Communications Surveys & Tutorials* 18 (3) (2016) 2220–2243.
- [2] F. Betti Sorbelli, C. M. Pinotti, G. Rigoni, Range-free localization algorithms with mobile anchors at different altitudes: A comparative study, in: *Proceedings of the 21st ICDCN*, 2020, pp. 1–10.
- [3] H. Xiao, H. Zhang, Z. Wang, T. A. Gulliver, An rssi based dv-hop algorithm for wireless sensor networks, in: *2017 IEEE PACRIM*, IEEE, 2017, pp. 1–6.
- [4] L. Zhao, X. Wen, Dan Li, Amorphous localization algorithm based on bp artificial neural network, in: *International Conference on Frontiers of Internet of Things 2014*, 2014, pp. 178–183.
- [5] J. Wang, P. Urriza, Y. Han, D. Cabric, Weighted centroid localization algorithm: theoretical analysis and distributed implementation, *IEEE Transactions on wireless communications* 10 (10) (2011) 3403–3413.
- [6] J. zeng Wang, H. Jin, Improvement on APIT localization algorithms for wireless sensor networks, in: *2009 Int. Conf. on Networks Security, Wireless Comm. and Trusted Computing*, Vol. 1, IEEE, 2009, pp. 719–723.
- [7] D. Koutsonikolas, S. M. Das, Y. C. Hu, Path planning of mobile landmarks for localization in wireless sensor networks, *Computer Communications* 30 (13) (2007) 2577–2592.
- [8] R. Huang, G. V. Zaruba, Static path planning for mobile beacons to localize sensor networks, in: *Pervasive Computing and Communications Workshops*, 2007, IEEE, 2007, pp. 323–330.
- [9] J. Jiang, G. Han, H. Xu, L. Shu, M. Guizani, Lmat: Localization with a mobile anchor node based on trilateration in wireless sensor networks, in: *IEEE GLOBECOM*, IEEE, 2011, pp. 1–6.
- [10] B. Xiao, H. Chen, S. Zhou, Distributed localization using a moving beacon in wireless sensor networks, *IEEE Transactions on Parallel and Distributed Systems* 19 (5) (2008) 587–600.
- [11] S. Lee, E. Kim, C. Kim, K. Kim, Localization with a mobile beacon based on geometric constraints in wireless sensor networks, *IEEE Transactions on Wireless Communications* 8 (12) (2009) 5801–5805.
- [12] F. Betti Sorbelli, C. M. Pinotti, V. Ravelomanana, Range-Free Localization Algorithm Using a Customary Drone: Towards a Realistic Scenario, *Pervasive and Mobile Computing* 54 (2019) 1–15.
- [13] F. Betti Sorbelli, C. M. Pinotti, Ground localization with a drone and uwb antennas: Experiments on the field, in: *2019 IEEE 20th International Symposium on WoWMoM*, IEEE, 2019, pp. 1–7.
- [14] J. Chen, D. Raye, W. A. G. Khawaja, P. Sinha, I. Güvenc, Impact of 3d uwb antenna radiation pattern on air-to-ground drone connectivity, *2018 IEEE 88th Vehicular Technology Conference (2018)* 1–5.
- [15] W. A. G. Khawaja, O. Ozdemir, F. Erden, I. Guvenc, D. Matolak, Ultra-wideband air-to-ground propagation channel characterization in an open area, *arXiv preprint arXiv:1906.04013* (2019).
- [16] P. Sinha, Y. Yapici, I. Guvenc, Impact of 3d antenna radiation patterns on tdoa-based wireless localization of uavs, in: *IEEE Conference on Computer Communications Workshops*, IEEE, 2019, pp. 614–619.
- [17] DecaWave, Product Documentation, <https://www.decawave.com/product-documentation/> (2020).
- [18] 3D Robotics, Solo Specs: Just the facts, <https://www.mydronelab.com/reviews/3dr-solo.html> (2020).
- [19] DecaWave, Datasheet for the DWM1001C, <https://www.decawave.com/dwm1001/datasheet/> (2021).
- [20] F. Betti Sorbelli, S. K. Das, C. M. Pinotti, S. Silvestri, On the accuracy of localizing terrestrial objects using drones, in: *2018 IEEE International Conference on Communications (ICC)*, IEEE, 2018, pp. 1–7.
- [21] F. Betti Sorbelli, S. K. Das, C. M. Pinotti, S. Silvestri, Range based Algorithms for Precise Localization of Terrestrial Objects using a Drone, *Pervasive and Mobile Computing* 48 (2018) 20–42.