Range-Free Localization Algorithm Using a Customary Drone: Towards a Realistic Scenario ☆

Francesco Betti Sorbelli[a,b], Cristina M. Pinotti[b], Vlady Ravelomanana[c]

[a] *Dept. of Computer Science and Math., University of Florence, Italy*
[b] *Dept. of Computer Science and Math., University of Perugia, Italy*
[c] *Dept. of Computer Science, University of Denis Diderot, Paris, France*

Abstract

The localization of devices is a key ingredient of Internet of Things (IoT) which may require extra cost for deploying anchor nodes aware of their position. In this work, we propose a range-free localization algorithm, called Drone Range-Free (briefly, Drf), that replaces the anchor nodes with an off-the-shelf drone. When the radio communications are isotropic, our algorithm localizes the IoT devices ensuring the level of precision required by the final user. However, since an isotropic antenna that propagates its signal in a perfect sphere does not exist in reality, we face a more realistic scenario by proposing an enhanced version of our localization algorithm, called Drone Range-Free Enhanced (briefly, DrfE), that well tolerates high degrees of irregularity of the antenna signal. Under the anisotropic model, Drf and DrfE exceed in precision all the computational-light previous known range-free localization algorithms based on a mobile rover.

Keywords: Drone, range-free, localization, design of algorithm, radio propagation model.

1. Introduction

Localization is a substantial obstacle in the soon to come deployment of Internet of Things smart devices [1]. Due to limitations in form factor, cost per unit, and energy budget, not all the individual devices are expected to be GPS-enabled [2]. Also, the deployment of few anchor nodes aware of their position to train the remaining devices may be an unsatisfactory solution for the time and the cost that it requires [3]. Thus, a localization solution could involve a mobile rover to be reused for different deployments. However, a rover requires to be in advance aware of the terrain conditions: for example, it is not easy for a rover to overtake a river. Recently, Unmanned Aerial Vehicles (UAVs), have received increasing attention from research and industry community [4, 5]. In this work, we explore the possibility of using a customary drone in order to localize the IoT devices/sensors exploiting a range-free technique that relies only on radio communications, which may be irregular.

☆Part of this work has been accepted to the 4th IEEE International Conference on Smart Computing. This manuscript has been published into Elsevier Pervasive and Mobile Computing Digital Object Identifier no. http://dx.doi.org/10.1016/j.pmcj.2019.01.005

Email addresses: francesco.bettisorbelli@unifi.it (Francesco Betti Sorbelli), francesco.bettisorbelli@unipg.it (Francesco Betti Sorbelli), cristina.pinotti@unipg.it (Cristina M. Pinotti), vlad@liafa.univ-paris-diderot.fr (Vlady Ravelomanana)

1.1. Related Works

A localization algorithm could be categorized into two different classes, i.e., *range-free* and *range-based* localization algorithms [1, 6]. In the former, the sensor's position estimation is done without using any type of measurement, but only using information between the sensors/anchors. In the latter, the estimations are done exploiting several properties of the communication signals, e.g., Received Signal Strength Indicator (RSSI) or Time of Arrival (ToA). A typical range-based approach has a higher localization accuracy, but it requires additional hardware because it needs to take distance or angle measurements. On the other hand, a range-free approach is preferable to keep lightweight the cost of the infrastructure, but implies a lack of precision.

In this section, we first survey the range-based localization algorithms proposed in the literature that use a drone to localize the IoT devices/sensors. Then, we survey the range-free localization algorithms presented in the literature that use a mobile anchor, such as a rover or a patrol. At the best of our knowledge, except a preliminary version of this work presented in [7], there are no range-free localization algorithms that use a drone as a mobile anchor.

The range-based Omni algorithm discussed in [8, 9, 10] equips both the sensors and the drone with an Ultra Wide Band (UWB) omnidirectional antenna able to take measurements. A detailed analysis of the measurement errors (instrumental, rolling, altitude) that can affect the slant measurements taken by a drone has been proposed in [11]. Moreover, in [12, 9], the range-based Dir algorithm proposes an improved version of Omni that uses directional antennas. In [13], three path planning algorithms that allow a drone to, respectively, localize, verify, and verify with a guaranteed precision a set of positions in a secure manner have been presented. The security has a higher cost in terms of processing time: for example, several instances of TSP must be solved.

Xiao et al. propose in [14] a range-free localization algorithm, that we call XiaoSp (Xiao Static Path), which uses a rover. The mobile ground vehicle follows a predefined static path and broadcasts its current GPS position at regular intervals of time. They aim to build a constraint area that bounds the actual sensor's position, assuming that all the nodes know the communication range r of the mobile anchor and the distance between two consecutive beacons. Each sensor traces four circumferences of radius r centered at the first and the last beacon that the sensor hears, and at two beacons that precede and follow the two heard beacons. The resulting intersections create a constraint area where the sensor could reside. Depending on the shape of the constraint area, the sensor performs different strategies for determining its current position's estimation.

In [15], a range-free algorithm has been presented by Lee et al., that we denote LeeRp (Lee Random Path), which uses a rover. The rover follows a random path, and regularly broadcasts a beacon that consists of its GPS position, the communication radius r, and the beacon interval d. Like XiaoSp, this algorithm focuses on creating a constraint area where the sensor could reside. After the rover crosses the sensor's receiving disk, the sensor traces one circumference of radius r and another one of radius $r - d$ centered at the two GPS positions broadcast with the first and the last heard beacons. These circumferences, which create two annuli, intersect in two distinct and symmetrical intersection areas. The sensor uses the third beacon to find out in which intersection area it resides. Finally, the sensor estimates its position at the center of such an area, using easy geometrical rules.

Moreover, Ou et al. propose in [16] a range-free 2D localization algorithm, that we call OuRp (Ou Random Path). Although the behavior of this algorithm is very similar to that in [15], the sensors exploit a completely different argument to localize themselves. Namely, OuRp exploits the fact that if a sensor detects a chord in its receiving disk, delimited by the first and the last

consecutive heard beacons, the sensor must reside on the perpendicular bisector of such chord. Thus, if a sensor detects two non-parallel chords, it knows that its position is at the intersection of the two bisectors. Since the mobile anchor follows a random path, there is no guarantee that all the nodes will be localized. To improve on this aspect, in [17], another range-free localization algorithm has been presented by Ou et al., that we call OuSp (Ou Static Path), in which the trajectory is defined by a series of vertical and horizontal scans. OuSp improves over OuRp because it guarantees that all the sensors are localized at the end of the algorithm.

Recently, another range-free algorithm has been presented in [18]. Here, the mobile anchor designs a random path, and each sensor collects all the beacons that it can hear. Among all the collected beacons, the sensor selects the three beacon points which form the triangle with the largest perimeter to obtain the first constraint area where the sensor could reside. After that, the constraint area is iteratively narrowed down by successive approximations and optimizations exploiting new generated random points. This scheme cannot be considered a light localization algorithm at the sensor's side due to the high computation of multiple intersections. Therefore, in Sec. 5, we will compare Drf and DrfE algorithms only with the light range-free algorithms presented in OuSp, XiaoSp, and LeeRp.

1.2. Our Contributions

In this work, we present a new *range-free* localization algorithm, called Drone Range-Free (Drf), that uses a drone as the mobile anchor. We show that Drf can guarantee any user-defined localization precision by varying the interval between two consecutive beacons. We also present a new enhanced version, called Drone Range-Free Enhanced (DrfE), which alleviates some of the most frequent problems that may occur applying the localization algorithm in an anisotropic scenario. To estimate the performances of our new algorithms, we conduct extensive simulations. We compare the average and maximum localization error as well as the path length of Drf and DrfE to those of OuSp, XiaoSp, and LeeRp. In the isotropic model, Drf outperforms the precision of all the other algorithms but LeeRp which, however, traverses a very long path. In the anisotropic model, DrfE localizes all the sensors and exceeds in precision over all the algorithms.

The rest of this paper is so organized: Sec. 2 explains the communication model and the localization method when a drone is used. Sec. 3 presents the new localization algorithm, called Drf, under the isotropic model. In Sec. 4, we move towards a more realistic scenario by considering radio signals affect by the *Degree Of Irregularity* (*DOI*), and we present an enhanced version of Drf, called DrfE, to cope with the radio irregularity. An extensive set of simulations for both the isotropic and anisotropic models is proposed in Sec. 5. Sec. 6 offers conclusions.

2. The Framework

In this section, we introduce the communication model with a flying anchor, the adopted range-free localization method, and the problems raised in the localization process by a mobile anchor.

We assume to have sensors uniformly and randomly deployed in a limited area. Each IoT device, or sensor p, is equipped with an isotropic antenna and can establish communications in a circular sphere around it of radius r_p. W.l.o.g., we use the term *isotropic* when we refer to the antenna radiation pattern on the 3D space, and the term *omnidirectional* when we consider a spherical cap projected in the 2D space. We want to localize the sensors using a drone D that acts as a mobile anchor node. We aim to avoid the cost of setting a fixed infrastructure for the localization process. The drone flies at a given altitude h above the ground and it is in turn equipped with an isotropic

antenna which transmits in a circular sphere of radius r_d, centered at the current position of the drone. From now on, $r = \min\{r_p, r_d\}$ will be denoted as the *communication radius*.

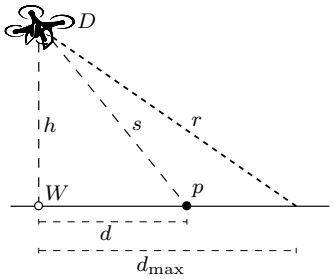
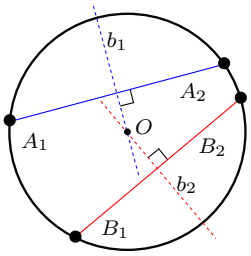


Figure 1: The drone and the sensor p.

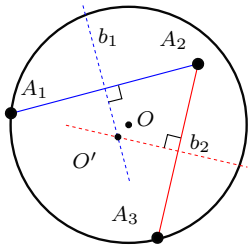
Fig. 1 shows the drone that is flying at an altitude h and its beacon can be heard by any sensor p on the ground such that the slant distance from the drone and p is $s \leq r$. The radius of the omnidirectional antenna on the ground when the drone is at an altitude h is $d_{\max} = \sqrt{r^2 - h^2}$. So, a sensor can hear a message sent by the drone if and only if it is positioned at a ground distance $d \leq d_{\max} = \sqrt{r^2 - h^2}$ from the projection of the drone on the ground. Therefore, we consider that each sensor has an effective omnidirectional *receiving disk* on the ground of radius $d_{\max}$, inner to its transmission circle of radius r. The radius of the receiving disk decreases when h increases.

The drone is equipped with a GPS module. The drone is able to convert its GPS coordinates into the (x, y, z) coordinates of a predefined 3D-Cartesian system. Regularly, the drone broadcasts a message called *beacon*. The beacon consists of the drone's 2D position (x, y), i.e., the projection on the ground of its current 3D position (x, y, z). From now on, we call *waypoint* the ground position (x, y) of the drone. Each sensor on the ground that can hear a beacon stores the waypoint in its register. A localized sensor knows its position with respect to the reference Cartesian coordinate system with origin at HOME in $(0, 0)$. The drone sends the beacon at regular instants of time. Let the *inter-waypoint* distance I_w be the distance between the waypoints into two consecutive beacons. We assume that I_w is an input parameter of our algorithm and it is fixed for each mission.

In the following, we explain the range-free localization mechanism, known in the literature (see [16, 17]), that we adopt in DRF and DRFE. A sensor to be localized has to detect two chords in its receiving disk. Then, the sensor learns to be at the intersection of the two perpendicular bisectors associated with such two chords. This mechanism is explained in Fig. 2(a).



(a) Two chords and the intersection.



(b) The estimation O' using two quasi-chords.

Figure 2: The range-free localization method.

The first chord is identified by the points A_1 and A_2 of the circle centered in O where the sensor resides. The perpendicular bisector of the chord A_1A_2 is the line b_1. The second chord is

identified by the circle's points B_1 and B_2 and its perpendicular bisector is the line b_2. Then, the intersection point of the two different perpendicular bisectors b_1 and b_2 coincides with the center O. This localization method fails when the two chords are parallel: in such a case, b_1 and b_2 coincide and there are infinite intersections, and therefore infinite positions where the sensor may reside.

2.1. The Quasi-Chord Problem

The localization is exact when the two detected waypoints exactly fall on the circumference of the receiving disk. Unfortunately, since the mobile anchor node sends the message beacons at discrete time instants, the last waypoint of a chord that the sensor records may indeed fall inside the disk. This situation is illustrated in Fig. 2(b) for the endpoint A_2. From now on, we call *quasi-chord* the segment detected by the sensor listening to the drone to highlight the fact that the endpoints of the chord may be internal to the disk. In Fig. 2(b), both A_1A_2 and A_2A_3 are quasi-chords. Because the point A_2 does not belong to the circumference, the bisectors b_1 and b_2 of the quasi-chords do not pass through the center O, and their intersection point O' does not match the sensor's position O. The distance $\overline{OO'}$ represents the *localization error* and it is due to the fact that, since the beacons are sent at discrete intervals, the length of the chord may not be exact. As we will see in Sec. 4, this problem is even more serious in the anisotropic model.

3. The Isotropic Model

In this section we describe in detail the new algorithm Drone Range-Free, or briefly Drf.

3.1. Drf*: The Drone's Algorithm*

We consider a network of n sensors deployed in a rectangular and flat area Q of size $Q_x \times Q_y$.

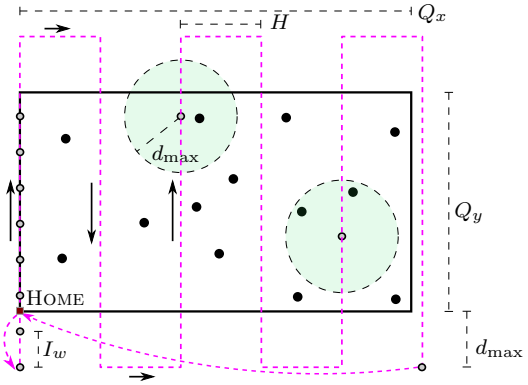


Figure 3: The drone's trajectory Π of algorithm Drf.

The drone's algorithm is shown in Algorithm 1. We assume that the origin $(0, 0)$ of the Cartesian coordinate system is at the bottom-left corner of the deployment area, called Home. The trajectory Π starts and finishes at Home and it consists of vertical scans connected by horizontal scans of length H, as depicted in Fig. 3. Let the *inter-scan distance*, i.e., the distance between two consecutive vertical scans, be H. Each vertical scan continues beyond the top and bottom borders for a segment of length $d_{\max}$. The last vertical scan falls at distance at most H beyond the right border of the deployment area. Completed the last vertical scan, the drone returns to Home.

Algorithm 1 DRF: Drone behavior

Input: $Q = Q_x \times Q_y$; H, I_w, HOME $=(0,0)$
Output: Trajectory Π

$\nu \leftarrow \left\lceil \frac{Q_x}{H} \right\rceil$ ▷ number of vertical scans
$i \leftarrow 0$
$x_F \leftarrow 0, y_F \leftarrow -d_{\max}$ ▷ starting point
moveFromTo$(Home, F)$
$x_P \leftarrow x_F, y_P \leftarrow y_F$
$V \leftarrow Q_y + 2d_{\max}$ ▷ length of a vertical scan
while $i < \nu$ **do**
  if $i \bmod 2 = 0$ **then**
    $x_{P'} \leftarrow x_P, y_{P'} \leftarrow y_P + V$
  else
    $x_{P'} \leftarrow x_P, y_{P'} \leftarrow y_P - V$
  end if
  sendBeaconFromTo(P, P', I_w)
  if $i < \nu - 1$ **then**
    $x_{P''} \leftarrow x_{P'} + H, y_{P''} \leftarrow y_{P'}$
    moveFromTo(P', P'')
  end if
  $P \leftarrow P''$
  $i \leftarrow i + 1$
end while
moveFromTo$(P, Home)$

3.2. DRF*: The Sensor's Algorithm*

The sensor's algorithm during the localization is shown in Algorithm 2. Each sensor listens to its radio until it has received three waypoints, not all three on the same vertical scan. DRF selects for each sensor the segment of the first vertical scan that crosses its receiving disk that is longer than I_w as the first quasi-chord. So the first quasi-chord is a vertical quasi-chord of length $\geq I_w$. Then, the sensor accepts any subsequent waypoint that it hears and builds the second quasi-chord using one of the two endpoints belonging to the first quasi-chord and the third heard waypoint.

3.3. Analysis

In this section, we show how to select a proper value of the inter-scan distance H and we study the localization precision of our solution.

3.3.1. Choice of H

The choice of H is crucial for the correctness as we prove in Lemmas 1 and 2.

Lemma 1. *The inter-scan distance $H = \sqrt{{d_{max}}^2 - I_w^2}$ guarantees that each sensor detects at least two quasi-chords, and at least one of them has length I_w.*

Proof. First observe that a sensor may reside at the border of Q: in such a case only half of its receiving disk falls in the area. Since our algorithm needs three waypoints taken from two different scans, we must guarantee in the worst case that there are two vertical scans in half of the receiving disk of the sensor. Thus, $H \leq d_{\max}$. Moreover, we must select H in such a way that there is at least one scan that intersects the receiving disk with a segment of length $\geq 2I_w$ to satisfy the constraint that any vertical quasi-chord selected by the algorithm is at least of length I_w. Recalling that the length of a chord at distance t from the center of the disk is $2\sqrt{{d_{\max}}^2 - t^2}$, to have a chord of length at least $2I_w$, at least one scan must crosses the sensor disk at most at distance $\sqrt{{d_{\max}}^2 - I_w^2}$. So, the value of H that minimizes the length of the drone's trajectory is $\sqrt{{d_{\max}}^2 - I_w^2}$. □

Algorithm 2 DRF: Sensor behavior

```
1: heard ← 0, i ← 0
2: C_p[1 : 3] ← ∅, B_p[1 : 2] ← ∅
3: while i < 3 do
4:     M ← WaitAndParseBeacon()
5:     if M ≠ ∅ then                                              ▷ timeout
6:         if heard = 0 then
7:             heard ← 1, B_p[1] ← M, i ← i + 1
8:             C_p[i] ← M
9:         else
10:            B_p[2] ← M
11:        end if
12:    else                                                       ▷ timeout expired
13:        if (heard = 1 & B_p[2] = ∅) then                       ▷ discard
14:            C_p[1] ← ∅, i ← 0
15:        end if
16:        if (heard = 1 & B_p[2] ≠ ∅) then                       ▷ 1st chord
17:            i ← i + 1
18:            C_p[i] ← B_p[2]
19:        end if
20:        heard ← 0
21:    end if
22: end while
23: return computeLocation(C_p)
```

Lemma 2. *Using an inter-scan distance $H = \sqrt{{d_{max}}^2 - I_w^2}$, if a sensor discards a single waypoint, it can still hear two quasi-chords.*

Proof. We must consider three cases.

a) at distance greater than H from the vertical borders of Q. A quasi-chord is discarded if it has length $< I_w$. Hence, the vertical scan crosses the left side of the receiving disk of the sensor at a distance $\sqrt{{d_{\max}}^2 - I_w^2} = H < t \leq d_{\max}$ from the center. So the next scan will be at distance $t - H$ from the sensor. The chord on such vertical scan has length $2\sqrt{{d_{\max}}^2 - t^2} \geq 2I_w$. Both the first two scans, the one discarded and the subsequent, are at the left of the sensor. So, since the radius of the disk is always greater than H, we are sure that a third quasi-chord will fall in the disk.

b) the sensor is at distance at most H from the right border of Q. If the scan is exactly on the right border, the sensor never discards the first quasi-chord. Otherwise, the first quasi-chord on the left may be discarded. If this happens, the sensor selects the first quasi-chord longer than I_w on the last scan inside Q, and the third waypoint on the last scan, which is outside Q.

c) the sensor is at distance at most H from the left border of Q. It never discards a chord. It always selects the first two waypoints on the first scan which passes on the left border of Q and the third waypoint is on the subsequent scan. □

3.3.2. Worst Case Localization Error

Now we prove the localization precision of our solution. Let p be the sensor that we want to localize. To better describe the behavior of the localization error, in our discussion, we use a Cartesian coordinate system with origin in the actual position O of p. The circular area from which the sensor can hear the drone is delimited by the circumference Γ_O. We select the first two waypoints C and B on the leftmost scan σ (see Fig. 4). Since we use a vertical quasi-chord and since the distance between any two beacon-points is I_w, we are guaranteed that whichever vertical

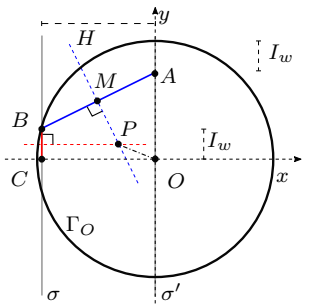


Figure 4: The representation of the worst case during localization.

quasi-chord we pick, the perpendicular bisector of that quasi-chord is not more than of $I_w/2$ from the parallel line passing through O. In addition to the two waypoints C and B on the leftmost scan σ, we select a third waypoint A on the next scan σ', as illustrated in Fig. 4. The sensor p estimates its position at the intersection P of the bisectors of the two quasi-chords CB and BA, which are clearly not parallel. The distance $\overline{PO}$ is the localization error (see Fig. 4). Note that, with respect to O, either B and A belong to I $\cup$ II quadrants, or B and A belong to III $\cup$ IV quadrants (see Fig. 4). The first case occurs only if the drone traverses bottom-up the σ scan, and top-down the σ' scan. Viceversa, the second case occurs only if the drone traverses top-down the σ scan, and bottom-up the σ' scan. Since the two cases are symmetric, we limit our analysis to the first case in which $y_A \geq 0$ and $y_B > 0$.

The maximum error on the position of the bisector of the vertical quasi-chord happens if the quasi-chord has one extreme on Γ_O and the other on the scan σ, at distance I_w from Γ_O. So to maximize the error due to the vertical quasi-chord, we fix $B \in \Gamma_O$ and C on σ and at distance I_w from the intersection of σ and Γ_O. The x-coordinate of A is uniquely determined once that of B has been fixed, while y_A is the only variable. Specifically, we study the error $\overline{PO}$ when: i) B lies at the intersection between σ and Γ_O and the coordinate x_B varies between $-H$ to 0, ii) C lies on σ and it is at distance I_w from Γ_O, iii) A resides on the scan σ' with $x_{\sigma'} = x_B + H$ and $y_A > 0$, and A is at distance $0 \leq t_A \leq I_w$ from the point given by the intersection of σ' and Γ_O. We have evaluated $\overline{PO}$ using Maple when t_A varies, and we have found that, when $I_w \ll d_{\max}$, the maximum error occurs for $t_A = I_w$. Precisely,

Lemma 3. *When $I_w \ll d_{max}$, the maximum error occurs for: i) $B = (-H, \sqrt{{d_{max}}^2 - H^2}) = (-H, I_w)$, ii) $C = (-H, \sqrt{{d_{max}}^2 - H^2} - I_w) = (-H, 0)$, and iii) $A = (0, d_{max} - I_w)$.*

Proof. The Maple computation has been reported in the Appendix. □

Assuming that the endpoints of the quasi-chords are the points A, B, and C derived in Lemma 3 that leads to the maximum error, the following result can be stated:

Theorem 1. *When $I_w \ll d_{max}$, the maximum localization error for* DRF *is:*

$$\overline{PO} = \frac{\sqrt{2}}{2} I_w \cdot \sqrt{\frac{5d_{max} - 4I_w}{d_{max} + I_w}} \tag{1}$$

Proof. Recalling that P is at the intersection of the line that represents the maximum vertical error $y = \frac{I_w}{2}$ and $\perp_{BA}$ that passes through the midpoint $M = (x_M, y_M)$ of the points A and B given

in Lemma 3, we know that $\overline{PO} = \sqrt{\left(-m\left(\frac{I_w}{2} - y_M\right) + x_M\right)^2 + \left(\frac{I_w}{2}\right)^2}$ where the slope $m = \frac{y_A - y_B}{H}$. Given that the maximum error occurs when $B = (-H, I_w)$, $C = (-H, 0)$, and $A = (0, d_{\max} - I_w)$, we obtain the claimed result as follows:

$$\overline{PO} = \sqrt{\left(-\frac{d_{\max} - 2I_w}{H} \cdot \frac{I_w - d_{\max}}{2} - \frac{H}{2}\right)^2 + \left(\frac{I_w}{2}\right)^2} = \sqrt{\left(\frac{{d_{\max}}^2 + 2I_w^2 - 3I_w d_{\max} - H^2}{2H} - \frac{H}{2}\right)^2 + \frac{I_w^2}{4}}$$

$$= \sqrt{\left(\frac{3I_w^2 - 3I_w d_{\max}}{2H}\right)^2 + \frac{I_w^2}{4}} = \sqrt{\frac{9I_w^2(I_w^2 - 2I_w d_{\max} + {d_{\max}}^2)}{4({d_{\max}}^2 - I_w^2)} + \frac{I_w^2}{4}} = \frac{\sqrt{2}}{2} I_w \cdot \sqrt{\frac{5d_{\max} - 4I_w}{d_{\max} + I_w}} \quad \square$$

A good bound of Eq. (1) when $I_w \ll d_{\max}$ is:

$$\overline{PO} = \frac{\sqrt{2}}{2} I_w \cdot \sqrt{\frac{5d_{\max} - 4I_w}{d_{\max} + I_w}} \approx \frac{\sqrt{10}}{2} I_w \tag{2}$$

Using the simple relationship between I_w and the error in Eq. (2), if a precision ϵ_L (i.e., the maximum obtainable localization error) is required by the final user of the localization process, it is possible to derive the particular value of I_w that guarantees such a precision. Therefore, when $I_w \ll d_{\max}$, DRF guarantees the required precision ϵ_L, if:

$$I_w = \frac{2}{\sqrt{10}} \epsilon_L \tag{3}$$

4. The Anisotropic Model

The ideal isotropic model, in which the drone's antenna radiates the radio signal in a sphere of radius r is impossible in the real world. In a real scenario, the antenna pattern is anisotropic. We model the antenna pattern using the *Degree of Irregularity* (DOI) parameter defined in [19] as *the maximum path loss percentage variation per unit degree change in the direction of radio propagation.* A zero-value of DOI indicates a perfect sphere, while positive values indicate some irregularity. DOI is able to reflect the two main properties of radio irregularity, i.e., anisotropy and continuous variation. Fixed the signal wavelength λ, the Received Signal Strength (RSS) is:

$$RSS = P_{T_X} - FSPL(d) = P_{T_X} - 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) \tag{4}$$

In the isotropic model, fixed a sensitivity value $RSS = \Sigma$, the signal is heard at distance d if $\Sigma \geq P_{T_X} - FSPL(d)$. So the communication radius r is the largest value of d that satisfies $\Sigma = P_{T_X} - FSPL(r) = P_{T_X} - 20\log_{10}\left(\frac{4\pi r}{\lambda}\right)$.

The DOI model varies the distance up to which the signal is heard in function of the direction i between the transmitting and the receiving antennas. In presence of DOI, Eq. (4) is rewritten, according to [19], as $RSS = P_{T_X} - 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) \cdot K_i$ where λ depends on the radio frequency and:

$$K_i = K_{i-1} \pm Rand \cdot DOI \quad i \in \mathbb{N} \tag{5}$$

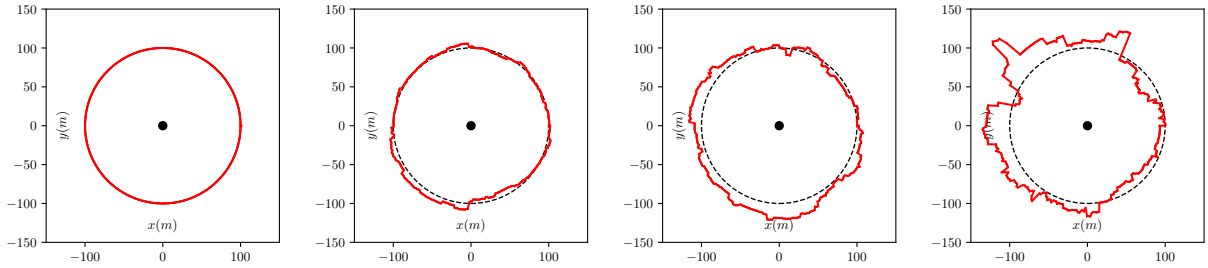
for $0 < i < 360$, where $|K_0 - K_{359}| \leq DOI$ and $K_0 = 1$. Thus, in presence of DOI, fixed the same sensitivity Σ, the signal is heard up to the largest distance r'_i such that $\Sigma \geq P_{TX} - FSPL(r'_i) \cdot K_i$.

Finally, to bring back the 3D distance on the ground, we compute:

$$d'_i = \sqrt{r'^2_i - h^2} \tag{6}$$

for $0 \leq i < 360$. Note that when the relative direction is $i = 0$, it holds $r'_0 = r$ and $d'_0 = d_{\max}$.

An example of the antenna pattern projected on the ground using different values for *DOI* is depicted in Fig. 5. Here $d_{\max} = \sqrt{r^2 - h^2} = 100\,\text{m}$ is drawn by the black dashed line, while the red line represents the maximum 3D radius, projected on the ground and different in each direction, for various values of the *DOI* parameter.



(a) $DOI = 0$ (b) $DOI = 0.003$ (c) $DOI = 0.005$ (d) $DOI = 0.010$

Figure 5: Examples of irregular antenna pattern on the ground.

4.1. Impact of DOI and Problems with Chords

In Sec. 2.1, the chords were built assuming a perfect circle as the receiving disk of the sensor. The quasi-chords were just affected by the inter-waypoint distance I_w. In the realistic model, as shown in Fig. 5, the resulting radio pattern could be fairly different from the original ideal circle centered at the origin of the sensor. In this general case, it is quite easy to see that the chords are afflicted by two indicators: the inter-waypoint distance I_w and the irregularity of the radiation pattern. An example is shown in Fig. 6. Here the expected communication radius is depicted by a thin dashed line, while the effective irregular radiation pattern is the solid filled one. According to the DRF algorithm, the sensor detects the first chord using the endpoints A_1 and A_2: A_1 is placed outside the dashed line because the sensor can hear the beacon at a distance greater than $d_{\max}$, while A_2 falls inside the disk due to the inter-waypoint distance I_w. The localization error in Fig. 6 is the distance between O and P.

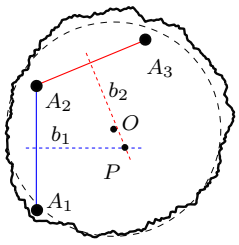
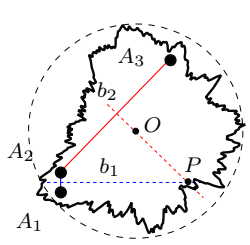


Figure 6: $DOI = 0.003$.

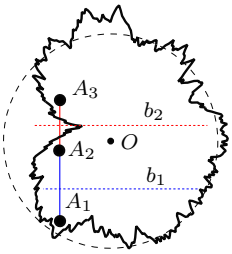
In the following, we introduce an enhanced version of DRF, called DRFE, that restrains the effect of the irregularity on the localization precision.

4.2. The Enhanced Algorithm

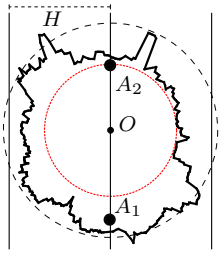
In this section, we propose a new enhanced version of the DRF algorithm, called DRFE, explicitly designed to deal with the aberrations of the anisotropic model. We have taken into account the following enhancements: 1) set a threshold on the chord length, 2) merge broken (collinear) chords on the same scan, and 3) redefine the inter-scan value.



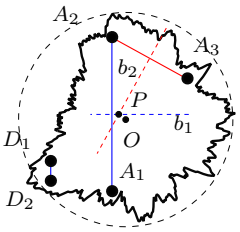
(a) No bound on the chord length.



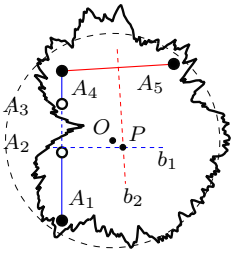
(b) Collinear chords.



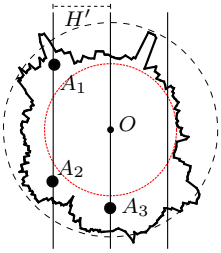
(c) Isotropic inter-scan H.



(d) Minimum chord length $\rho_{\min}$.



(e) Collinear chords merged.



(f) Smaller inter-scan $H' < H$.

Figure 7: The three enhancements of DRFE.

1) Threshold Chord Length. Figs. 7(a) and 7(d) give a useful hint of the anomalies due to the strong impact of DOI. In this particular case, the scan intersects the radiation pattern in a narrow edge located in the bottom-left side: the sensor hears the first endpoint A_1 and, starting from A_2, it does not hear anymore. The detected vertical quasi-chord leads to a very large error: the y-coordinates of both P and O are quite apart (see Fig. 7(a)). We propose to limit the effect of such pattern shrinkage by accepting only quasi-chords longer than a certain threshold $\rho_{\min}$. As shown in Fig. 7(d), discarding the current quasi-chord and taking a longer quasi-chord on the subsequent scan, the error may decrease. Clearly, to increase the chances to find a longer vertical chord, the inter-scan distance must decrease so as the receiving disk is intercepted by more than three scans.

Similarly, whenever the quasi-chord is longer than $\rho_{\max} = 2d_{\max}$, we are sure that the scan intersects an irregular disk. Since we have no evidence in which direction the signal overflows the regular disk, we can just discard the chord. Hence, increasing the number of scans that intersect the receiving disk of a sensor gives more chances to find enough chords to localize each sensor.

2) Merging Collinear Chords. Sometimes the pattern shape could generate lobes (see Figs. 7(b) and 7(e)). Consider for example Fig. 7(b). According DRF, the sensor detects the first quasi-chord A_1A_2. The sensor waits until it hears the successive beacon sent by the drone. So, when the sensors registers A_3, it can start the localization procedure. It is easy to see that in this case we have collinearity with respect to the three selected endpoints, and hence the bisectors of the two chords are parallel and they do not intersect. This anomaly can be easily resolved by adding, to

the beacon message, the identifier of the current vertical scan σ. When the sensor stops to hear the beacons (in A_2) for the current scan σ, it remains pending for any possible beacon which contains the same σ value. Therefore, as depicted in Fig. 7(e), the sensor merges the two quasi-chords A_1A_2 and A_3A_4 into a new longer chord (A_1A_4), and discards the internal endpoints A_2 and A_3.

3) Redefining Inter-scan Value. In the isotropic model, H has been chosen with the goal to guarantee that each sensor detects at least two quasi-chords, out of which one of length at least I_w. For this reason, the inter-scan value was set to $H = \sqrt{{d_{\max}}^2 - I_w^2}$. Now, we do not have a perfect receiving disk but rather an irregular shape whose radius d'_i in direction i, according to Eq. (6), with $0 \leq i < 360$. We consider the largest inscribed circle, centered at the sensor position:

$$\underline{d_{\max}} = \min_i \{d'_i\} \leq d_{\max} \tag{7}$$

Repeating the same reasoning we did in the isotropic model, in order to cross the shape at least two times and to have at least one chord of length $\rho_{\min}$, the new value of H' is:

$$H' = \sqrt{{\underline{d_{\max}}}^2 - {\rho_{\min}}^2} \tag{8}$$

Figs. 7(c) and 7(f) explain the aforementioned enhancement. The black dashed line represents the isotropic radius $d_{\max}$, and the red line depicts the largest inscribed circle of radius $\underline{d_{\max}}$. Using $H = \sqrt{{d_{\max}}^2 - I_w^2}$, we may not be able to guarantee at least two quasi-chords, as shown in Fig. 7(c). Instead, using a smaller value for the inter-scan (Fig. 7(f)), the sensor can hear a sufficient number of scans. This enhancement, however, leads to longer localization paths for the drone.

4.2.1. The DrfE *Algorithm*

In this section we present the pseudo-code of DrfE, the enhanced version of Drf.

The drone's algorithm in DrfE is almost the same as that one seen in Drf. In Algorithm 3 Line 3, the drone computes the static path with respect to the smaller inter-scan value H' applying Eq. (8). Moreover, now the drone sends the current vertical scan σ (Line 5) along with the waypoint (x, y) to allow the merging of the collinear chords.

On the sensor's side, two enhancements are implemented. We set thresholds on the chord's length, and we merge the chords that belong to the same scan. In Algorithm 4 Line 5, if the sensor detects a quasi-chord of length less than the threshold $\rho_{\min}$ or greater than $2d_{\max}$, it discards the current quasi-chord. Moreover, in Line 3, when the sensor hears the first beacon of the vertical scan σ, it waits until it hears the last beacon on σ before setting the final end-point of the chord.

Algorithm 3 DrfE: Drone behavior

Input: Q: area; DOI; I_w, $\rho_{\min}$: min chord, $Home$
Output: Trajectory Π
1: setParameters()
2: $H' \leftarrow$ computeInter-ScanDistance($DOI, \rho_{\min}$)
3: computeStaticPath(H')
4: **while** missionIsNotFinished() **do**
5: sendBeaconFromTo($P, P', \sigma, \rho_{\min}$)
6: **end while**
7: moveFromTo($Home$)
8: land()

Algorithm 4 DrfE: Sensor behavior

Input: $\rho_{\min}$: min chord; $\rho_{\max} = 2d_{\max}$: max chord
1: setParameters()
2: **while** twoChordsAreNoteDetected() **do**
3: hearsAllBeaconsAlongScan(σ)
4: $c \leftarrow$ chordDetected()
5: **if** len(c) $< \rho_{\min}$ **or** len(c) $> \rho_{\max}$ **then**
6: discardChord(c)
7: **end if**
8: **end while**
9: **return** computeLocation(C_p)

5. Simulation evaluation

We have implemented both Drf and DrfE localization algorithms in MATLAB and we have tested their performance, also in comparison with other algorithms presented in the literature.

5.1. Settings

We simulate a localization mission by deploying at random n nodes, with $n = \{50, 100, 200, 300\}$ on a 500×500 m^2 map. We generate 35 different random sensor deployments for each value of n. In all the experiments, we set the 3D communication radius $r = \{50, 100, 150\}$ m and the altitude $h = \{15, 30\}$ m varying the inter-waypoint distance $I_w = \{1, 5, 10\}$ m. In the anisotropic model, r is then affected by the $DOI = \{0.003, 0.005, 0.01\}$ parameter.

We define the *experimental error* as the distance between the actual sensor's position and the estimated position calculated. For each localization mission, we register the *maximum* and *average* experimental errors, which are, respectively, the maximum and the average experimental error measured over all the n sensors. We plot the results by averaging the maximum and average experimental error on all the 35 deployments. The performance of all the algorithms is evaluated by using two metrics: 1) localization error, and 2) path length. We do not explicitly and experimentally evaluate the energy spent by the sensors because in all the studied algorithms the computational burden is very light (few multiplication and additions). Moreover, we consider the path length as an indicator of the energy spent by the drone. For example, in our setting, considering that a typical mission of Drf has about 5 km length, the drone can accomplish two distinct missions, while the sensors can participate to at least 30 missions without recharging their battery.

For the isotropic model, in Sec. 5.2, we experimentally confirm that it is possible to keep the maximum experimental error below the user-required *localization precision* ϵ_L by varying the inter-waypoint distance as claimed in Eq. (3). Moreover, we compare the performances of Drf with that of OuSp, LeeRp, and XiaoSp. For the anisotropic model, in Sec. 5.3, we first compare the performances of Drf with that of DrfE for different levels of DOI. In particular, we trace the number of sensors that remain unlocalized after both algorithms. A sensor is unlocalized if it cannot estimate its position. For example, a sensor remains unlocalized if it hears two parallel quasi-chords, one on the extension of the other, or if it does not hear a sufficient number of beacons. Finally, we compare the performances of the new algorithms with those of OuSp, LeeRp, and XiaoSp under different levels of irregularity.

5.2. Isotropic model

To adapt OuSp, LeeRp, and XiaoSp to drones, we select the report the communication radius on the ground at $d_{\max} = \sqrt{r^2 - h^2}$, instead of r.

5.2.1. Localization error

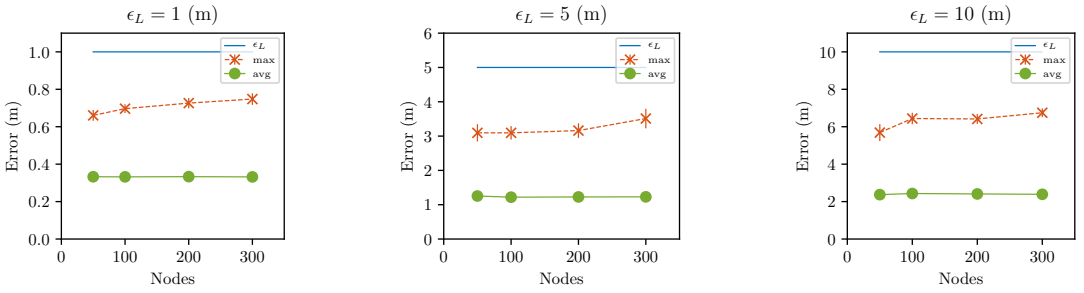


Figure 8: The localization error in DRF when $r = 100, h = 15\,\text{m}$.

When $r = 100$, $h = 15\,\text{m}$, Fig. 8 illustrates the behavior of DRF using different values of the user-defined localization precision ϵ_L. Precisely, fixed $\epsilon_L = \{1, 5, 10\}\,\text{m}$, according to Eq. (3), DRF sets, respectively, $I_w = \{0.60, 3.20, 6.30\}\,\text{m}$. The experiments confirm that the maximum experimental error is always below the user-defined precision ϵ_L. In general, the average error is one-fifth of the user-defined precision ϵ_L.

Fig. 9 compares the localization errors for DRF, LEERP, XIAOSP and OUSP.

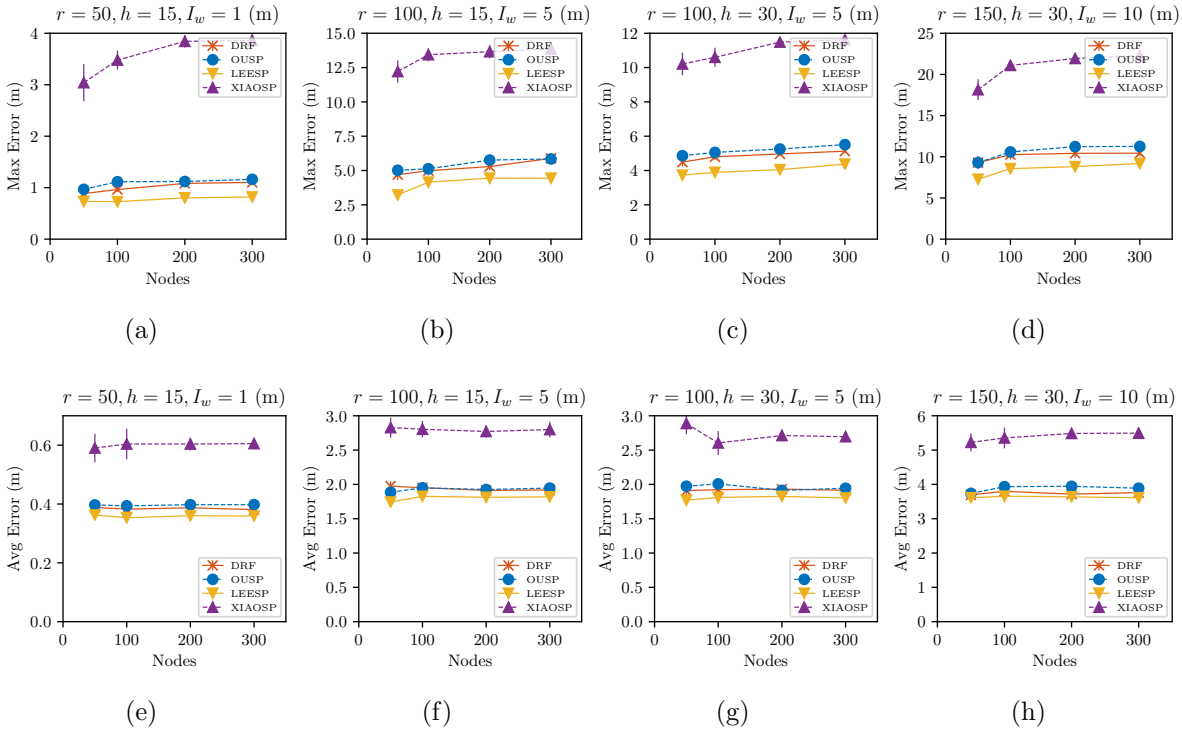


Figure 9: The average and maximum error in DRF, OUSP, LEERP, and XIAOSP.

The upper row of Fig. 9 plots the maximum error, whereas the lower row plots the average error of the compared localization algorithms under different settings. The maximum and the average

error tend to be close to, respectively, I_w and $\frac{I_w}{2}$ in all the algorithms, but XiaoSp. The maximum error of XiaoSp is about three times larger than that of the other algorithms when r and I_w are small. The gap between the maximum errors of XiaoSp and the other algorithms decreases when r and I_w increase. Then, we can conclude that XiaoSp is the least accurate algorithm, and as such it is less sensitive to the increase of the implicit uncertainty, i.e., the inter-waypoint distance. In general, for all the algorithms, when h increases, the maximum localization error increases, while the average localization error remains stable.

Drf improves over OuSp, especially for large values of r because Drf forces a vertical quasi-chord. The uncertainty due to the inter-waypoint cannot be larger than I_w for vertical chords, while the uncertainty can be larger for arbitrary oblique chords. So, OuSp loses to Drf because it has, in general, two oblique chords.

Finally, in all our experiments, as shown Fig. 9, the most accurate algorithm is LeeRp. Both in maximum and in average, LeeRp allows to have smaller localization errors because its error depends only on the error on the vertical chord delimited by the first and the last heard beacon. Nevertheless, LeeRp has the longest path as shown in Fig. 10 and it can withstand the irregularity of the radio signal worse, as we will prove in Section 5.3.

In conclusion, LeeRp is slightly more accurate than Drf, which is slightly better than OuSp. The error performance of XiaoSp follows that of all the other three algorithms with a large gap.

5.2.2. Path Length

Fig. 10 compares the static path performed by the drone in XiaoSp, LeeRp, OuSp, and Drf.

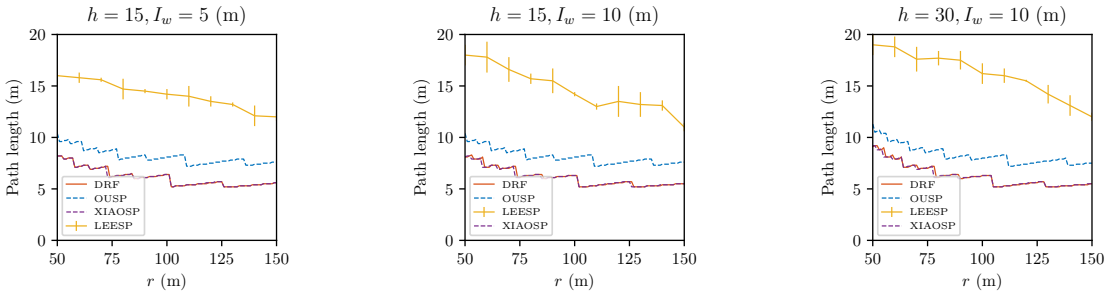


Figure 10: The path length in Drf, OuSp, LeeRp, and XiaoSp.

As expected the length of the path, decreases when r increases. The static paths performed by the drones in Drf, OuSp, and XiaoSp scan the deployment area in a very similar way. It is worthy to note that LeeRp, which is the most accurate localization algorithm, requires a very long and undetermined random path. This algorithm could hypothetically never terminate with an overall 100% localization. Since we know the number of sensors, we stop LeeRp when all the nodes have been localized. In our experiments, on average, the length of the path of LeeRp is twice that of Drf. Considering that with a single recharge a drone can fly about 10 km, only the three paths of Drf, OuSp and XiaoSp can be accomplished by a customary drone without recharging the battery.

5.3. Anisotropic model

To test the performance of the DRF and DRFE algorithms in the anisotropic model, we assume the same experimental setting described in Sec. 5.1.

On the sensor's side, the main difference is that the receiving disk of each sensor is affected by a degree of irregularity. We assume that all the sensors are afflicted by the same *DOI* value, although each sensor has its own antenna pattern. To derive the antenna pattern, we randomly generate the variation of the path loss by using Eq. (5), and we obtain by Eq. (6) the maximum current 2D distance d'_i up to which the sensor hears the beacon sent by the drone. In the simulation, to decide if a sensor hears a beacon, we check if the projection W of the drone's position on the ground falls inside the irregular antenna pattern of the sensor itself. The act of hearing is simulated by: i) determining the angle i between the waypoint W and the sensor P, ii) computing the 2D distance $\overline{WP}$, and finally iii) comparing $\overline{WP}$ with d'_i.

On the drone's side, the main difference is the inter-scan H'. To compute H' (see Eq. (8)), one must know the minimum radius $\underline{d_{\max}}$ given by Eqs. 6 and (7). To estimate $\underline{d_{\max}}$, which depends on the *DOI* parameter, we generated thousands of antenna patterns at different radii and altitudes, using the Weibull distributions with *scale* and *shape* parameters in the ranges suggested by [19]. For each generated antenna pattern, we recorded the minimum radius $\underline{d_{\max}}$ and the overall average value is reported in Tab. 1. These values are computed at a 95% confidence level.

Table 1: The experimental average difference $d_{\max} - \underline{d_{\max}}$ in percentage, for some *DOI* values.

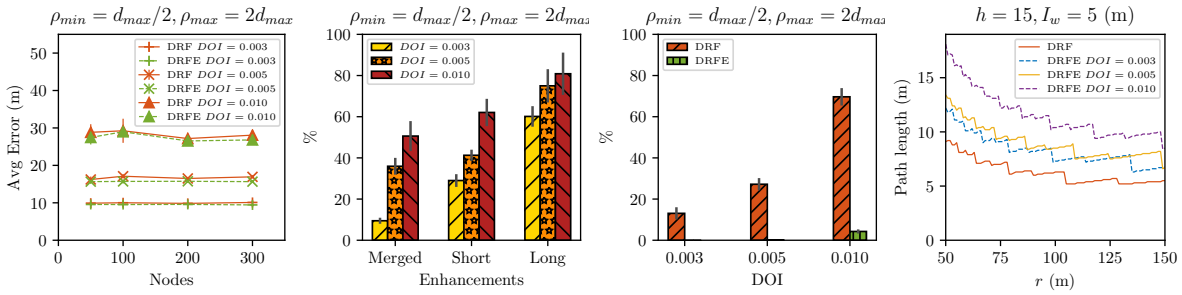
DOI	$d_{\max} - \underline{d_{\max}}$ (%)
0.003	−9.80
0.005	−15.57
0.010	−28.32

5.3.1. Localization error

Our first goal is to compare the performances of the two new range-free localization algorithms DRFE and DRF in presence of irregularities in the propagation of the signal. We aim to verify how the *DOI* value impacts the DRF performance, and how much better DRFE is than DRF.

Impact of DOI. In Fig. 11, we compare DRF and DRFE in presence of radio irregularities. In the simulations, we set $DOI = \{0.003, 0.005, 0.010\}$, and according to Tab. 1 and depending on the current setting, we derive the value $\underline{d_{\max}}$ to be used to design the static path of DRFE. Precisely, two vertical scans of the static path of DRFE are at distance H', given by Eq. (8).

In Fig. 11(a), we compare the average localization error of DRF and DRFE when the minimum chord length is set to $\rho_{\min} = d_{\max}/2$ and the maximum set to $\rho_{\max} = 2d_{\max}$. This value of $\rho_{\min}$ is used to select the chords close to the center of the receiving disk of each sensor where the signal is stronger and less affected by the DOI parameter. Results show that the main difference between the two algorithms is the number of unlocalized sensor, which is reported in Fig 11(c). For $DOI = 0.003$, in DRF about 10% of the sensors remain unlocalized, while DRFE is capable to localize all of them. Where $DOI = 0.005$, the number of unlocalized sensors increases for DRF. Finally, a large number or sensors remain unlocalized when $DOI = 0.010$ in DRF and even few sensors in DRFE do not estimate their position (less than 5%). The growth of the number of the unlocalized sensors is much more than linear in DRF. This shows that it is crucial to select



(a) (b) (c) (d)

Figure 11: The impact of DOI when $r = 100, h = 15, I_w = 5\,\text{m}$.

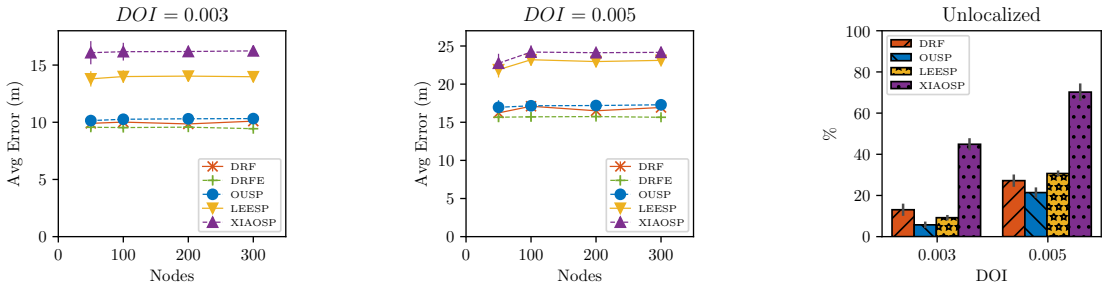
the inter-scan distance H' for being able to collect at least three non-collinear beacons for each sensor. Regarding the error of localized nodes only, both DRF and DRFE work well, as reported in Fig. 11(a). Doubling the value of the DOI parameter, the average error almost doubles.

Fig. 11(b) reports the usage of each single enhancement used in DRFE when $n = 300$. For $DOI = 0.003$, almost 10% of the chords are merged, about 30% are discarded because too short, and about 60% are discarded because too long. For larger DOI values, the percentage of merged chords increases. Same behavior is also valid for the discarded ones.

Fig. 11(d) depicts the path length comparison. As expected DRF has the shortest path length because it imposes a larger inter-scan value than that of DRFE. When the DOI value increases, the overall path length obviously increases because $\underline{d_{\max}}$ decreases. This increase of the path length is the price to pay in order to localize all the sensors in the more realistic anisotropic model. It is likely that to accomplish the DRFE mission, especially when $DOI = 0.010$, a customary drone needs to stop once for recharging the batteries, or two drones can be used.

Summarizing, the comparison between DRF and DRFE at different levels of irregularity shows that, excluding extreme cases, DRF can handle weak anisotropic signals still achieving a reasonable precision on the average case. DRFE gains more in precision over DRF if the irregularity is strong. Notable, DRFE localizes almost all the sensors.

DRFE *vs. all.* Finally, Fig. 12 compares all the algorithms in presence of DOI.



(a) (b) (c)

Figure 12: Comparison when $r = 100, h = 15, I_w = 5\,\text{m}$.

To start, Fig. 12(c) reports the percentage of unlocalized in the anisotropic model for each algorithm. When $DOI = 0.003$ and 0.005, DRFE localizes all the sensors (as already seen in

Fig. 11(c)). Thanks to the random path, the number of unlocalized nodes is not very critical for LeeRp. Instead, XiaoSp has a number of unlocalized sensors larger than Drf although it uses the same inter-scan distance as Drf.

As regard to the average localization error of the localized nodes only, Figs. 12(a) and 12(b) show that the radio irregularities heavily afflict both LeeRp and XiaoSp. Interestingly, OuSp, Drf, and DrfE that are based on the perpendicular bisector method work better than LeeRp and XiaoSp that are based on constraint area method in presence of radio irregularities. This is due to the fact that in LeeRp and XiaoSp the range $d_{\max}$ is fundamental to derive the node position. Namely, both methods assume that when the sensor receives, it is at distance $d_{\max}$ or $d_{\max} \pm I_w$ from the drone's position and they draw a constraint area based on such assumption. Whereas, due to the radio irregularity, the communication range can be quite different. On the other hand, OuSp, Drf, and DrfE find the sensor position only as a function of the heard positions, without making any assumption on the relative distance between the sensor and the drone.

In conclusion, the enhanced version DrfE is quite effective because it clears the number of unlocalized sensors and it preserves accuracy. This comes with an increase of the path length, which is larger for small values of r and for large values of DOI.

6. Conclusion

In this paper, we presented two new range-free localization algorithms, called Drf and DrfE, that replace multiple fixed anchors nodes with a single flying drone. Drf assumes an ideal isotropic model of communication and it guarantees the localization precision required by the final-user. When the radio signal propagates anisotropically, Drf handles localization with a good precision on average in presence of weak irregularities, but up to 20% of sensor remain unlocalized. Then, we proposed DrfE to cope with high levels of irregularity and to clear the number of unlocalized sensors. The efficiency of the DrfE algorithm is paid in terms of length of the drone's path.

Finally, a wide set of simulations is offered to compare the performance of Drf and DrfE with that of other range-free algorithms, originally proposed in the literature for being implemented using a rover. Both Drf and DrfE outperform all the other algorithms in almost all scenarios.

More investigation of the practical challenges of our new localization algorithms is left for future work. We also plan to consider a localization mission performed by a small fleet of drones, to overcome the path length issue.

Acknowledgment

The work has been partially supported by GEO-SAFE (H2020-691161), and Project *NALP-SAPR: Navigazione Autonoma e Localizzazione Precisa per Sistemi Aeromobili a Pilotaggio Remoto*, granted by FSE-Regione Umbria and by Fondo Ricerca di Base, 2018, University of Perugia.

Appendix: Proof of Lemma 3

Proof. The configuration of A, B, and C that leads to the maximum error is shown in Fig. 4. Given the initial configuration, and recalling that P is at the intersection of the two quasi-chords bisector, points A, B, and P satisfy the following equations: i) $x_B^2 + y_B^2 = {d_{\max}}^2, y_B > 0$, ii) $(x_B + H)^2 + (y_A + I_w)^2 = {d_{\max}}^2$, iii) $x_P = -\frac{(y_A - y_B)}{H}\left(\frac{I_w}{2} - \frac{y_A + y_B}{2}\right) + \frac{x_A + x_B}{2}, y_P = \frac{I_w}{2}$. We find that x_P

can be expressed in terms of x_B, $d_{\max}$, H as follows: $x_P = f(x_B)$ with $f(x) = x + \frac{1}{2}H - \frac{Q_1(x)Q_2(x)}{H}$ where $Q_1(x) = I_w - \frac{1}{2}\sqrt{{d_{\max}}^2 - (x+H)^2} - \frac{\sqrt{{d_{\max}}^2 - x^2}}{2}$ and $Q_2(x) = -I_w + \sqrt{{d_{\max}}^2 - (x+H)^2} - \sqrt{{d_{\max}}^2 - x^2}$. Differentiating f, we find that $f'(x) = 0$ for x such that: $8x^4 + 16\sqrt{{d_{\max}}^2 - I_w^2}x^3 - 8I_w^2x^2 - 18\sqrt{{d_{\max}}^2 - I_w^2}{d_{\max}}^2x + 9I_w^2{d_{\max}}^2 - 9{d_{\max}}^4 = 0$. As $I_w \ll d$, the equation $f'(x) = 0$ is well-approximated by $\text{Poly}(x) = 0$ with $\text{Poly}(x) = 8x^4 + 16d_{\max}x^3 - 8I_w^2x^2 - 18{d_{\max}}^3x - 9{d_{\max}}^4$. Using routine of quartic equation solving, the existence of real solutions of $ax^4 + bx^3 + cx^2 + dx + e = 0$ depends on the sign of $P_5(a,b,c,d,e) - P_6(a,b,c,d,e)$ with $P_5 = \frac{b^2}{2a^2} - \frac{4c}{3a} - P_3$, $P_6 = \frac{-\frac{b^3}{a^3} + 4\frac{bc}{a^2} - 8\frac{d}{a}}{P_4}$, $P_4 = \sqrt{\frac{b^2}{4a^2} - \frac{2}{3}\frac{c}{a} + P_3}$, $P_3 = \frac{2^{1/3}}{3a}\frac{c^2 - 3bd + 12ae}{P_2^{1/3}} + \frac{2^{2/3}P_2^{1/3}}{6a}$, $P_2 = P_1 + \sqrt{-4(c^2 - 3bd + 12ae)^3 + P_1^2}$, and $P_1 = 2c^3 - 9bcd + 27ad^2 + 27b^2e - 72ace$. In our case, using $a = 8$, $b = 16d_{\max}$, $c = -8I_w^2$, $d = -18{d_{\max}}^3$, and $e = -9{d_{\max}}^4$ we find: $P_5 - P_6 = \left(2 - \frac{1}{48}2^{2/3}15\,552^{1/3} - \frac{20}{\sqrt{2^{2/3}3^{2/3}+4}}\right){d_{\max}}^2 + O(1)$. A quartic equation has real solutions if $P_5 - P_6 \geq 0$. For our purpose, we have $P_5 - P_6 \sim -6.23 \cdot {d_{\max}}^2 < 0$ and $f(x)$ is monotonically increasing from $f(-H)$ to $f(0)$. Being $f(x) < 0$, $|f(x)|$ is maximized for $x_B = -H$. □

References

[1] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, G. K. K., A survey on mobile anchor node assisted localization in wireless sensor networks, IEEE Communications Surveys & Tutorials 18 (3) (2016) 2220–2243.

[2] G. Mao, B. Fidan, B. D. Anderson, Wireless sensor network localization techniques, Computer networks 51 (10) (2007) 2529–2553.

[3] G. Han, H. Xu, T. Q. Duong, J. Jiang, T. Hara, Localization algorithms of wireless sensor networks: a survey, Telecommunication Systems (2013) 1–18.

[4] L. Gupta, R. Jain, G. Vaszkun, Survey of important issues in uav communication networks, IEEE Communications Surveys & Tutorials 18 (2) (2016) 1123–1152.

[5] A. Idries, N. Mohamed, I. Jawhar, F. Mohamed, J. Al-Jaroodi, Challenges of developing uav applications: A project management view, in: IEOM, 2015 International Conference on, IEEE, 2015, pp. 1–10.

[6] E. Q. Shahra, T. R. Sheltami, E. M. Shakshuki, A comparative study of range-free and range-based localization protocols for wireless sensor network: Using cooja simulator, International Journal (IJDST) 8 (1) (2017) 1–16.

[7] F. Betti Sorbelli, C. M. Pinotti, V. Ravelomanana, Range-Free Localization Algorithm Using a Customary Drone, in: 2018 IEEE SMARTCOMP 2018, Taormina, Sicily, Italy, 18-20 June 2018, 2018, pp. 9–16.

[8] C. M. Pinotti, F. Betti Sorbelli, P. Perazzo, G. Dini, Localization with Guaranteed Bound on the Position Error using a Drone, in: MobiWac 2016, Malta, December 13-17, 2016, ACM, 2016, pp. 147–154.

[9] F. Betti Sorbelli, S. K. Das, C. M. Pinotti, S. Silvestri, Range based Algorithms for Precise Localization of Terrestrial Objects using a Drone, Elsevier Pervasive and Mobile Computing (PMC) 48 (2018) 20–42.

[10] F. Betti Sorbelli, C. M. Pinotti, On the Localization of Sensors using a Drone with UWB Antennas, in: CEUR Workshop Proceedings, July 19-20, 2018, L'Aquila, Italy, Vol. 2146, 2018, pp. 18–29.

[11] F. Betti Sorbelli, S. K. Das, C. M. Pinotti, S. Silvestri, On the Accuracy of Localizing Terrestrial Objects Using Drones, in: ICC 2018, Kansas City, MO, USA, 20-24 May 2018, 2018, pp. 1–7.

[12] F. Betti Sorbelli, S. K. Das, C. M. Pinotti, S. Silvestri, Precise Localization in Sparse Sensor Networks using a Drone with Directional Antennas, in: ICDCN 2018, Varanasi, India, January 4-7, 2018, 2018, pp. 34:1–34:10.

[13] P. Perazzo, F. Betti Sorbelli, M. Conti, G. Dini, C. M. Pinotti, Drone Path Planning for Secure Positioning and Secure Position Verification, IEEE Transactions on Mobile Computing 16 (9) (2017) 2478–2493.

[14] 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.

[15] 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).

[16] K.-F. Ssu, C.-H. Ou, H. C. Jiau, Localization with mobile anchor points in wireless sensor networks, IEEE transactions on Vehicular Technology 54 (3) (2005) 1187–1197.

[17] C.-H. Ou, W.-L. He, Path planning algorithm for mobile anchor-based localization in wireless sensor networks, IEEE Sensors Journal 13 (2) (2013) 466–475.

[18] M. Singh, S. K. Bhoi, P. M. Khilar, Geometric constraint-based range-free localization scheme for wireless sensor networks, IEEE Sensors Journal 17 (16) (2017) 5350–5366.
[19] G. Zhou, T. He, S. Krishnamurthy, J. A. Stankovic, Models and solutions for radio irregularity in wireless sensor networks, ACM Transactions on Sensor Networks (TOSN) 2 (2) (2006) 221–262.