UNIVERSITÀ
DEGLI STUDI
FIRENZE

UNIVERSITÀ
DEGLI STUDI
PERUGIA

Istituto Nazionale
di Alta Matematica

Università di Firenze, Università di Perugia, INdAM consorziate nel CIAFM

DOTTORATO DI RICERCA

IN MATEMATICA, INFORMATICA, STATISTICA

CURRICULUM IN INFORMATICA

CICLO XXX

Sede amministrativa Università degli Studi di Firenze

Coordinatore Prof. Graziano Gentili

Localization of Terrestrial Objects Using a Drone with UWB Antennas

Settore Scientifico Disciplinare INF/01

Dottorando:
Francesco Betti Sorbelli

Tutore:
Prof. Maria Cristina Pinotti

Coordinatore:
Prof. Graziano Gentili

Anni 2014/2017

Contents

List of Figures

List of Tables

Abstract

Unmanned Aerial Vehicles (UAVs) have enormous potentials for several important applications, such as search and rescue and structural health monitoring. An important requirement for these applications is the ability to accurately localize objects, such as sensors or "smart-things", equipped with wireless communication capability. However, most previous works in this area neglect the unavoidable errors that are involved in the localization process, thus resulting in poor performance in practice. In this thesis, we express the measurement error on the ground as a function of the rolling, altitude, and instrumental precision provided by the hardware on the drone, in our case, the Impulse Radio Ultra Wide Band (IR-UWB) technology. Then, we study many sensor localization techniques that replace the expensive and fixed anchors with a single drone equipped with omnidirectional and directional antennas. Finally, the performance of our solutions, evaluated in terms of the static path length and of the achieved precision, are validated by simulation study.

2

Chapter 1

Introduction

This thesis has been inspired by two forthcoming popular technologies: drones and directional antennas.

Drones have been widely used in military contexts: for first reconnaissance in the enemy field or for monitoring the borders of a country as a countermeasure against illegal immigration. Drones are now available off-the-shelf at a very affordable price, and they are becoming popular in civilian applications as well. Amateurs already use them for entertainment. But, drones are becoming common in emergency plans to monitor rescue operations, to fly-over large-scale disaster areas, such as earthquakes or nuclear disasters. Not only amateurs or first help forces, but also giants of the digital innovation, like Amazon and Google, think to use drones to speed up shipping services. Although the roadmap to reach such prime-air services is still long and paved by great challenges, in the near future, it is expected that drones will be in charge of homeland security (private surveillance and reconnaissance), forest fire monitoring, and farming, to name a few applications.

Directional antennas focus transmission power in a particular direction, limiting the nodes (or antennas) that can hear the communication. This improves the communication range, reduces interference with nearby antennas, and more in general reduces the antennas that can reply to a communication. Directional antennas provide increased performance over omnidirectional antennas, and increase the communication locality.

In this thesis, we envision drones that act as a lightweight and re-usable infrastructure to connect and collect data from the plethora of objects of our everyday life endowed with computation, communication, and storage capabilities (Internet of Things). The first challenge to face is the ability to localize such terrestrial nodes (Things) in order to make meaningful the collected data. Our drone is not equipped with cameras, but with Impulse Radio Ultra Wide Band (IR-UWB) antennas. IR-UWB uses a very low energy level for short-range communication (up to 300 m

line of sight, and 40 m non-line of sight), has a high data rate and is immune to multipath interference. We expect the drones will substitute the anchor nodes equipped with GPS, drastically cutting the deployment of the infrastructure for cyber-physical systems. Chapter 2 presents the basic elements of our work.

In Chapter 3, we consider the measurement precisions. When the mobile anchor flies, it takes 3D measurements that have to be brought back to 2D (or ground-) measurements. Projecting the distances on the ground, we commit errors that increase with instrumental error, the altitude, and the inaccuracy of the drone position. We learn two important lessons: the error is bounded if the terrestrial node and the drone are sufficiently apart on the ground, and if the measurements of a terrestrial node are taken by positions that surround it.

In Chapter 4, we propose a localization algorithm with a mobile anchor that follows a static path. In this chapter, the drone is equipped with omnidirectional antennas. The peculiarity of this algorithm is the ability to guarantee a user-defined precision, just monitoring a single parameter: the distance on the ground. However, two localizations are required to find the position of each node.

Although we expect UAVs will be a significant part of our daily life, we must be aware that they will raise many privacy and security threats. In Chapter 5, we propose a secure localization algorithm. As one can see the computational complexity of this algorithm is higher than that of the previous algorithm, but this is expected as the price to be paid for a secure localization algorithm.

In Chapter 6, we introduce directional antennas. Swarms of drones can communicate by using directional antennas. We present a realistic fading model and an ideal model to describe the behavior of the directional antennas used by the drones. We explore the connectivity property of directional networks, showing that directional networks provide connectivity at very low power levels. The ideal model introduced here is then relevant for representing the directional antennas in Chapter 7.

In Chapter 7, we propose a new localization algorithm that uses a drone equipped with six directional antennas for localizing terrestrial nodes. Due to the ability of the directional antennas to focus communications, we find the beamwidth of the directional antennas able to guarantee the localization of all the nodes deployed. We show that it is possible to achieve precise localization by using a single trilateration for each node. In general, this algorithm is computationally lighter than that proposed in Chapter 4; we trade computational complexity with a slightly more complex hardware requirement.

In conclusion, we have shown that the altitude, which is the peculiar characteristic of a flying

anchor node, affects the localization error, but it can be well controlled with simple algorithms that find precise and secure localizations for any number of terrestrial nodes deployed in a predefined area. The use of smart hardware equipment, that is a set of directional antennas instead of an omni directional antenna, simplifies the localization algorithm, still preserving the precision. A secure localization, however, requires a longer and more complex path than a precise localization. Namely, in the case of a precise localization, the drone follows a precomputed static path which depends only on the deployment area, while in the case of a secure localization, computing the path is computationally expensive because the path depends on the positions of each node.

Chapter 2

Technologies and Terminology

2.1 The Wireless Sensor Networks

In the recent past, technological breakthroughs in ultra-high integration and low-power electronics have enabled the development of miniaturized battery-operated sensor nodes (*sensors*, for short) that integrate signal processing and wireless communications capabilities [1, 2].

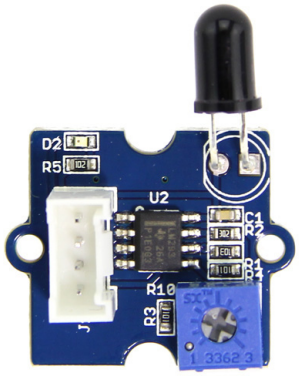


Figure 2.1: A simple sensor.

Together with innovative and focused network design techniques that will make possible massive deployment [3] and sustained low power operation, the small size and cost of individual sensors are a key enabling factor for a large number of applications. Indeed, aggregating sensors into sophisticated computational and communication infrastructures, called wireless sensor networks, has a significant impact on a wide array of applications ranging from smart kindergarten [4, 5], to smart learning environments [6–8], to habitat monitoring [9, 10], to environment monitoring [11, 12], to greenhouse and vineyard experiments [13], to forest fire detection [14], to helping the elderly and the disabled [11, 15], among others. The peculiar characteristics of sensor networks pose unique challenges to the design of protocols. First of all, the limited energy budget

requires the design of ultra-lightweight communication protocols. However, how data collected by sensors are queried and accessed and how concurrent sensing can be performed internally are of significance as well. An important guideline in this direction is to perform as much local data processing at the sensor level as possible, avoiding the transmission of raw data through the sensor network. This implies that the sensor network must be multi-hop and, for reasons of scalability, that no sensor knows the topology of the network.

Several possible techniques can be used for interfacing sensor networks to the outside world and for harvesting the data they produce. The simplest technique involves using one or several sinks (or actors), i.e., special long-range radios deployed alongside with the sensors [16–20]. Each sink has a full range of computational capabilities, can send long-range directional broadcasts to the sensors at distance at most ρ, can receive messages from nearby sensors, and has a steady power supply. In this scenario, the raw data collected by individual sensors are fused, in stages, and forwarded to the nearest sink that provides the interface to the outside world.

2.2 The Sensors Localization

The random deployment results in sensors initially unaware of their location. Localization is one the most important task in a wireless sensor network, since it provides fundamental support for many location-aware protocols and applications. Due to limitations in form factor, cost per unit and energy budget, individual sensors are not expected to be GPS-enabled. Moreover, many probable application environments limit satellite access. However, for the sensors localization, another approach is needed based on measurements and further positions' estimations. From these observations, the localization process can be split in two different steps: (i) distance measurements between neighboring nodes, and (ii) geometric calculation based on measured distances. Considering the distance measurement technique utilized, a localization algorithm could be categorized into two different classes of techniques, i.e., *range-free* and *range-based* localization algorithms. In the former, the sensor's position estimation is done by only using connectivity information between the sensors without using any type of ranging measurements. In the latter, the estimations are done exploiting several properties of the communication signals, i.e., Received Signal Strength Indicator (RSSI), Time of Arrival (ToA), Time Difference of Arrival (TDoA) and Angle of Arrival (AoA) [21, 22].

To exploit the distance measurement among neighbors, since it is very expensive to equip each node inside the network with a GPS module, a fixed set of *anchor nodes*, i.e., nodes whose positions are known a-priori, is used. Since the measurement range is not very long, an anchor

node can only be of support for a limited number of sensors. Therefore, the number of the fixed anchor nodes grows very fast especially if the network is dense. To resolve this problem, it is more convenient to replace all the fixed anchor nodes with *mobile anchor nodes* (see Fig. 2.2).

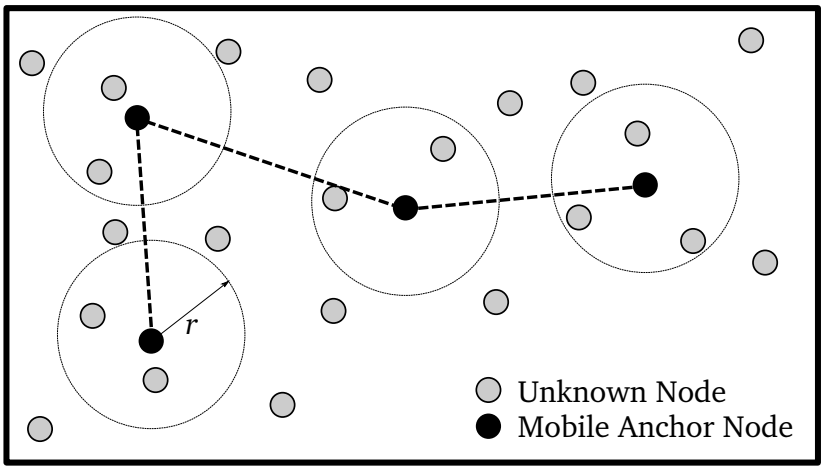


Figure 2.2: Mobile anchor node assisted localization.

Each mobile anchor node is equipped with a GPS unit and periodically stops and broadcasts its position to help nearby unknown nodes with localization (i.e., sensor in the circle in Fig. 2.2) [23, 24].

2.2.1 Classification of Localization Algorithms

In order to better understand the different typologies of localization algorithms, we present here a short survey. A taxonomy of the localization algorithms using mobile anchor nodes is proposed in Fig. 2.3. Since our study concentrates on localization algorithms based on mobile anchor nodes that assist all the sensors inside the network for localizing, we just consider the class of the *Localization Algorithms based on Mobile Anchors*. In general, it is worthy to note that there are many other ways to localize (for example, as mentioned before using fixed anchor nodes), but in our research we concentrate only on localization algorithms based on mobile anchor nodes.

Mobility Model

The localization algorithms are classified based on the *Mobility Model* in the network. If the movement pattern of each anchor is independent of the mobility of the other mobile anchor nodes, the mobility is *Individual*. In [25] the simplest case of Mobility Model is considered in which the mobile nodes move randomly and independently. In turn, the Individual Mobility Model class

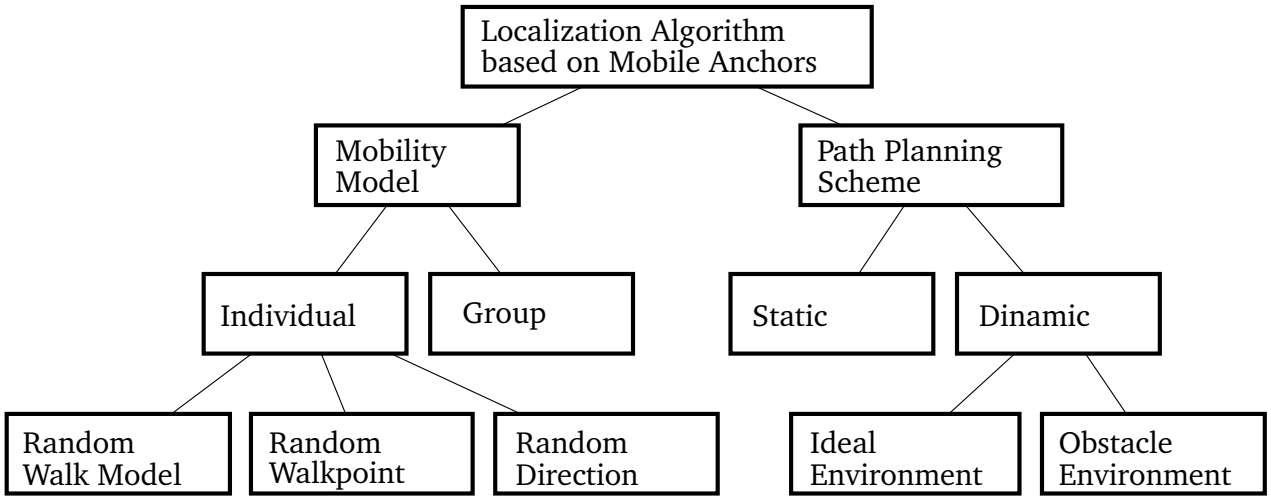


Figure 2.3: Classification of mobile anchor node assisted localization algorithms.

can be classified into three sub-categories, i.e., the Random Walk Model, the Random Walkpoint Model and the Random Direction Model.

The *Random Walk* Mobility Model is an Individual Mobility Model in which the mobile node moves at random following an irregular path in different directions and at different speeds. According to this pattern, a mobile node changes its current location to a new location by randomly choosing the speed and the direction distributed in a certain range [26].

The *Random Waypoint* Mobility Model [27] is an extension of the Random Walk Mobility Model because it considers pause time between changes in direction and speed. The algorithm proceeds in steps. At each step, the mobile node chooses the next location, it moves until arriving at the destination but waits in that position for a period of time which is uniformly distributed (i.e., a pause time). At this point, the mobile node can decide either to stop or to choose another location and repeat the process [28]. In [29] an algorithm that follows the random waypoint mobility model is proposed. The mobile anchor moving from one random point to another can draw chords in the transmission area of the sensors. To accomplish the localization the algorithm exploits the fact that if a sensor detects a chord in its transmission area, the sensor must reside on the line orthogonal at the middle point of that chord. Thus, in general, if a sensor detects two chords, it knows that its position is at the intersection of the two lines associated with such chords.

The *Random Direction* Mobility Model is a model that forces a mobile node to travel up to the edge of the sensing area by selecting at random only the speed and the direction [30]. This model was created to overcome the clustering of trajectories of the mobile nodes in the central part of the network, which is a drawback of the Random Waypoint Mobility Model. In [31] is presented

a fine-grained localization method using a mobile beacon node, which is based on geometric constraints as an extension of [29].

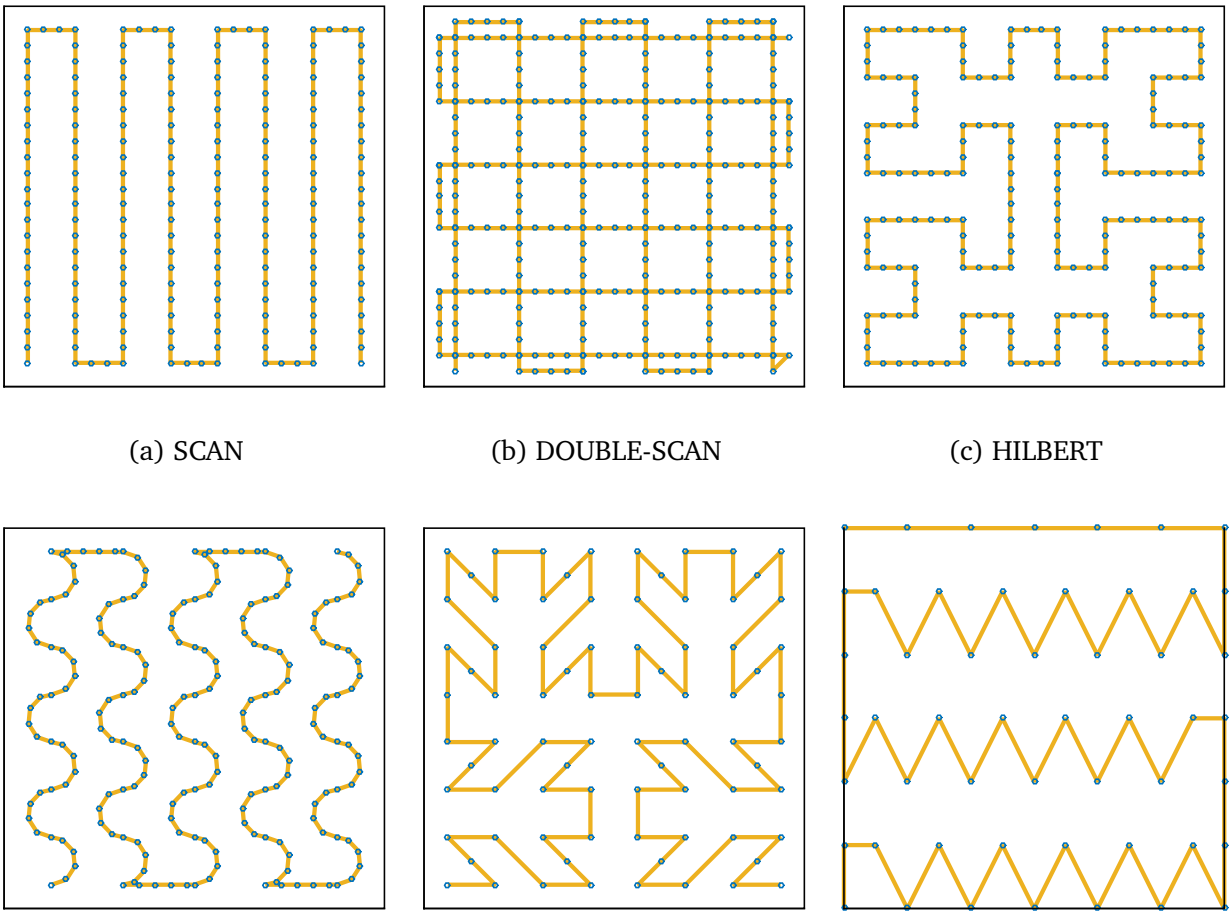
In the *Group* Mobility Model, in [32] a technique in which there are overall five anchor mobility nodes is proposed. Here, without using any distance or angle information, it is possible to estimate the position of the unknown nodes by only using elementary geometric considerations from the comparison of the Received Signal Strength (RSS) transmitted by the five anchors. In [33] nine anchors are exploited in order to localize. The nine mobile anchor nodes form a circle. Unknown nodes inside the circle will record the RSS from mobile anchor nodes and estimate their locations by comparing the RSS.

Path Planning

In recent years, with the advent of autonomous vehicles, the interest of researchers in rovers, drones or other vehicles is increased [34]. In other words, autonomous vehicles can be used for localizing objects after the study of *Path Planning*. The main goal in a Path Planning Scheme is to find a feasible movement trajectory for the mobile anchor node in order to improve the performance. The general requirements of Path Planning are essentially three, i.e.: (i) it should pass closely to as many potential node positions as possible, aiming at localizing as many unknown nodes as possible, (ii) it should provide at least three non-collinear anchor points in the 2D space (or at least four non-coplanar in the 3D space) in order to avoid wrong estimations, and (iii) is should be as short as possible in order to reduce the energy consumption of the anchor nodes and the unknown nodes.

As depicted in Fig. 2.3, a Path Planning Scheme can be classified into two sub-categories, i.e., Static or Dynamic. If the Path Planning Scheme is *Static*, i.e., the trajectory is designed before the localization process, the mobile anchor node follows a sequence of predefined points inside the sensing area. In the literature there exist many 2D Static Path Planning schemes.

In [35] 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 (see Fig. 2.4a), in which the mobile anchor node follows a path formed by vertical straight lines interconnected by horizontal lines. Essentially, the mobile anchor node sweeps the area along the y-axis. The main drawback is that it provides a large amount of collinear anchor points. In order to resolve the collinearity problem, DOUBLE-SCAN has been proposed (see Fig. 2.4b), in which the mobile anchor node 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 (see Fig. 2.4c) curve divides the area into 4^n square cells



(a) SCAN (b) DOUBLE-SCAN (c) HILBERT

(d) S-CURVES (e) Z-CURVE (f) LMAT

Figure 2.4: Different movement trajectories.

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 anchor points but the path length can be very long if the resolution increases. Therefore, the above path planning schemes are based on straight lines. In order to heavily reduce the collinearity during localization, in [36] was introduced the S-CURVES (see Fig. 2.4d) static path. S-CURVES 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 cover properly the four corners of the square sensing area. In [37] Z-CURVE (see Fig. 2.4e) is presented; it is very similar to HILBERT algorithm. The authors in [38] developed LMAT (see Fig. 2.4f), one of the best techniques in the state of art. The main idea is to plan a series of mobile anchor nodes in such a way as to form regular equilateral triangles, avoiding collinearity problems. Each node inside a triangle is localized by a trilateration procedure using the three vertices. The distance between the node and the mobile anchor node is measured based on Received signal strength indication (RSSI). Finally, the authors of [29] improved the localization procedure considering a Static Path Plan Scheme [39].

In the Path Planning Scheme *Dynamic*, i.e., the trajectory is not designed before the localization process, the mobile anchor node moves trajectories dynamically or partially according to the observable environments or deployment situations, etc. It is useful to use such trajectories when the unknown nodes are assumed to be not uniformly deployed. Namely, if this assumption is true, a Static Path Planning Scheme is not the best solution because it results in a waste of path length, long localization time and low utilization of beacon messages.

Under the *Ideal Environment*, i.e., obstacle free, a localization technique is presented in [40] which consists of three sub-processes, that is, a reference movement phase, a sensor localization phase and a movement path decision phase. The mobile anchor node initially sends different beacon messages. Depending on the message beacon packets requested by the unknown nodes, the mobile anchor node determines the movement trajectory in order to minimize the path length. At each step, the mobile anchor node chooses the nearest target among candidate points and receives additional request messages from new request nodes.

Finally, under the *Obstacles Environment*, we have a mobile anchor node which has the ability to face and detour obstacles, moving in a snake-like algorithm. When the mobile anchor node faces the obstacle, it changes its direction and stores all the information for future movements.

2.3 Introduction to Drones

Recently, Unmanned Aerial Vehicles (UAVs), also known as drones, are receiving increasing attention from the research and industry community [41].

Figure 2.5: The 3DR Solo Drone model.

Depending on their utilization, drones can be used for military and civilian applications. As an example in civilian applications, a drone is employed in rescue operations to recover from large-scale disaster events, such as earthquakes or nuclear disasters. Instead, in a military application, a drone can be used for monitoring the borders of a country as a countermeasure against illegal immigration.

However, it is expected that in the near future, drones will be used for other purposes and services, like surveillance and reconnaissance [42], homeland security [43, 44], forest fire monitoring [45], environmental monitoring [46], security and border surveillance [43], farming [47], or even Internet delivery [48], architecture surveillance [49] and goods transportations [50]. Therefore, UAVs will be a significant part of our daily life. However, this affects our privacy, hence an efficient drone path planning algorithm is required in order to obtain authentication, security, and trust.

Of particular interest are the applications enabled by combining the UAV technology with the paradigm of the Internet of Things (IoT) [41]. According to this paradigm, objects of our everyday life will be endowed with computation, communication, and storage capabilities, thus enabling them to share context and be connected anytime, anywhere [51, 52].

Usually drones are designed for specific tasks but it is possible to perform other user-defined tasks by equipping the drone itself with particular sensors, actuators and cameras. In this direc-

tion, one of the main challenges is the battery drainage, because the energy consumption during the flight is very high especially if the drone carries other specific hardware.

2.3.1 Classification of UAVs

We present now (see Fig. 2.6) a little survey about UAVs in order to better understand the different capabilities and services that UAVs could provide.

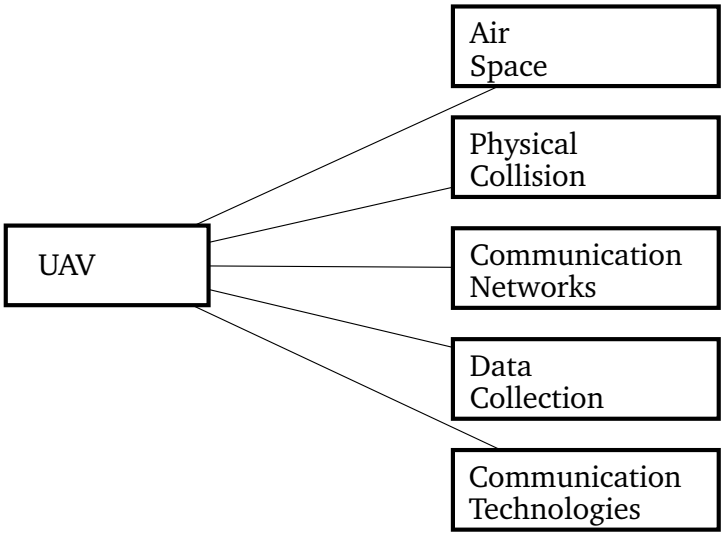


Figure 2.6: Criteria for Classifying UAVs.

Air Space

According to the adopted technology for the specific task about *Air Space* communications, we can distinguish between two different metrics, i.e., altitude and range [53]. We can classify the drones in three classes, i.e., *small*, *medium* and *large*. About altitude, a drone is classified as small if its maximum altitude is 300 m, while if the drone can fly above 5500 m it is considered as large. Otherwise, it belongs to the medium category. Concerning the other metric, if the maximum range is less than 3 km, a drone is considered as small, if it varies between 150–250 km it is medium, otherwise large. Regarding the first two classes, both support line-of-sight (LoS) communications. However, the large drones are beyond LoS (BLoS). Therefore, the majority of IoT services would be provided, principally, by small and medium drones. Our research is in the area of small drones.

Physical Collisions

One of the fundamental requisites during the flight, is to guarantee and preserve security for the people. The drone's flight operations need to be as secure as possible avoiding *Physical Collisions*.

In [54] a method that calculates the expected level of safety depending on different situations has been studied. The results show that the risk of impact to the ground varies significantly based on population density. Moreover, the risk increases when the speed and the mass of a drone increases. Another important aspect concerns the Collision Avoidance control. In fact, a real application needs to be carefully designed in order to avoid impact with ground objects or with other flying vehicles into the sky. The authors in [55] studied the collision avoidance system based on to the traffic alerts used for manned aircrafts. They show that the safety evaluations must be comprehensive and specific to the system and certain elements of the flight operation. Finally, path planning is another key point about maximization of safety [56].

Communication UAV Network

Once a drone has collected data or just finished its task, it needs to communicate the information to the base station in the network. An infrastructure in which a drone operates executing some task consists of terrestrial, satellite and wireless links to connect to the ground base station for storing the information shared by the drone [57–59]. Hence, it is very important to properly study a *Communication Network* between the drone and the base station. Under this aspect, we can have several Communication Network typologies, as depicted in Fig. 2.7.

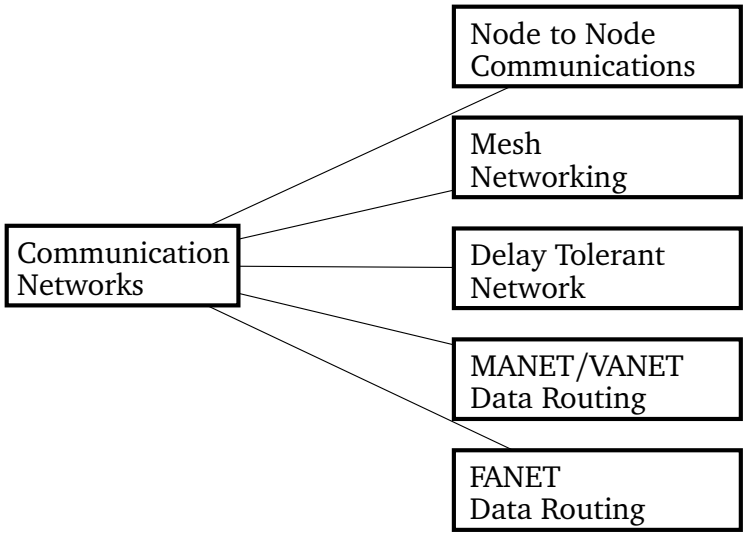


Figure 2.7: Classification of Communication Networks for UAVs.

The simplest architecture, i.e., *Node-to-Node Communication*, requires only direct links between the drone and the ground base station. In [60] a case study in which both UAV-to-UAV (U2U) and UAV-to-Infrastructure (U2I) communications cooperates at different layers inside the networking model is studied. The authors state that Mobile ad-hoc Network (MANET) protocols

are preferred for U2U communications due to the high mobility of the flying UAV. Moreover, for exchanging messages between UAVs and the infrastructure, it is sufficient to rely on U2I communications.

Another network type is *Mesh networking*, an architecture in which every node, i.e., UAV or a ground station, can act as a data relay. It is worthy to note that communications between multiple UAVs and the ground station are done exploiting intermediate nodes inside the network [61]. Moreover, there are many advantages to the usage of mesh networking. First of all, a short range of communication helps in simplifying the link requirements, so the bandwidth is reused more frequently and efficiently. Another aspect implies the U2U communication that becomes direct, in addition to maintaining the communication links thanks to mesh routing protocols.

An architecture that is able to guarantee interoperable communications among several network typologies is the *Delay Tolerant Network*. Such architectures, are suitable when the network is very heterogeneous and when the network presents link intermittent connectivity, lack of end-to-end connectivity between end-users as well as high latency [62, 63].

As introduced previously, *MANET* stands for Mobile ad-hoc Network. A MANET is an infrastructureless network [64] that exploits the connection links between the independent mobile nodes connected among them. This creates a very dynamic network in which the topology can change rapidly over time. In general, the MANET concept is referred to mobile nodes. Since a vehicular application includes a mobile node, there exists another concept, i.e., Vehicular ad-hoc Network (*VANET*). Under this aspect, a VANET extends the range of MANET. The authors in [65] have studied the characteristics of MANETs, VANETs and UAV networks. It has been seen that, evaluating some metrics like link establishment and topology, classical Mesh networking is still the most appropriate for UAVs, but the architecture of multi-UAV networks is still an open study area. In [66, 67] the Unmanned Aeronautical ad-hoc Network (UAANET) is explored, which is a kind of MANET that utilizes geographical routing.

Finally, we introduce the Flying ad-hoc Network *FANET* that represents a new class of ad-hoc network [68–74]. A FANET is based on the concept of a MANET but it is also considered as a subtype of VANET. Implementing the FANET approach is very useful since that solves the range restriction among the UAVs and the ground station and permits reliable and fast communications especially in critical situations where a classical ad-hoc network is not able to do this.

Data Collection

When a UAV flies in the sky it collects data using onboard IoT devices and other tools [75]. Once a drone has finished its task, it needs to share the information to a ground station. Hence, proper

Data Collection is needed.

Data collection and reporting methods can be classified into the following:

- *Time Driven*: the sensors will be activated at a certain time;
- *Space Driven*: the sensors will be activated regardless of timing;
- *Space-Time Driven*: a drone will execute specified tasks at a certain time and location;
- *Event Driven*: when an event is triggered, the drone will activate the sensors;
- *Query Driven*: a specified subject asks for certain information;
- *Space-Query Driven*: a specified subject asks for certain information and location;

One of the most common IoT component of UAVs is the camera. Cameras allow UAVs to execute several application, varying from autonomous vision system for safe landing [76–79] to path planning trajectory, environment monitoring, tracking and other purposes [76, 80].

In addition to the camera, other sensors are utilized for sensing and collecting data in the IoT infrastructure. There exist different types of physical sensors [81], each of them suitable for measuring various properties [82]:

- *Mechanical Sensors*: Pressure, movement, position, speed, acceleration;
- *Heat Sensors*: Temperature, heat, and heat flow;
- *Radiation Sensors*: Visible light, infrared, ultraviolet, nuclear;
- *Chemical Sensors*: Solution concentration, substance composition sensors/detectors;
- *Other Types of Sensors*: Touch, linear (analog), digital (numeric);

These sensors increase the ability to measure, analyze, and aggregate data at a much localized level.

Communication Ground-UAV Technology

In this section, we study the communication technologies that can be used to control the drone from the ground and to collect data from the ground with a drone. Depending on the *Communication Technologies* adopted, we can classify UAVs communications into two categories: short and wide range communications. The wide range communications class is able to cover a large area using technologies such WiMAX, SATCOMs or cellular. The short range communications class

instead, is only able to reach short distances, such Zigbee or Bluetooth. Special attention will be given for the IR-UWB technology because we adopted that technology in our research. The above considerations are shown in Fig. 2.8.

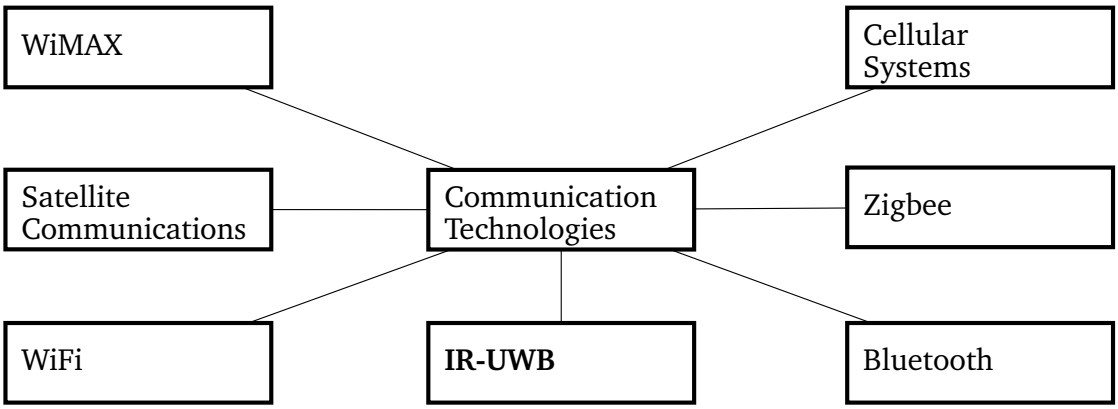


Figure 2.8: Classification of Ground-UAV Communication Technologies.

Nowadays, the recent improvements in wireless systems communications, in particular *Cellular Systems*, allows us to remotely control a UAV beyond the visual line of sight. In [83] the problem about the frequency bands utilized by drones is studied. In fact, the Industrial, Scientific and Medical (ISM) frequency band is used in most of UAV civil applications. The main problem is that those bands are very limited and it implies a limited flight coverage. In order to avoid this, cellular networks for the communications are considered. For example, the authors in [84] exploit the 4G-LTE system for UAV-based real-time video surveillance and streaming.

WiMAX, is a family of wireless communication standards based on the IEEE 802.16 set of standards. This wireless technology has many benefits because it permits, at a low cost, to cover a larger area than the classical Wi-Fi [85]. The typical data rates vary from 3–5 Mb/s for mobile applications up to 20–30 Mb/s for standard applications. Different bands are available for WiMAX applications. The frequencies commonly used are 2.5 GHz and 5.8 GHz. WiMAX is developed to handle high-quality voice, and video stream while providing high QoS. In [86] a rescue system based on a UAV infrastructure that uses the WiMAX technology has been studied. They show that WiMAX is an appropriate technology for a hostile environment due to its flexibility, wide coverage, and mobility.

In oder to establish reliable communications, especially in those applications in which the BLoS paradigm is applied, it is convenient to use *Satellite Communications*. With SATCOMs it is possible to overcome the problems like large distances or environmental obstructions, and they provide and assure mission critical communications [87]. A study in [88] evaluates the image transmission system based on SATCOMs communications as relay. It is shown that satellite relay

obtains larger overlay range and provides long distance image transmission with good image quality.

Wi-Fi is a technology for wireless local area networking with devices based on the IEEE 802.11 standards. Wi-Fi includes a set of standards for implementing WLAN communication in 2.4, 3.6, 5, and 60 GHz frequency bands. This technology is used in flight control and real-time data such as photo and video transmission between UAVs and devices on the ground. The research in [89] has developed a method to carry the Wi-Fi signal to the emergency areas by using a UAV system. Exploiting the capabilities of directional antennas, it is possible to extend the traditional Wi-Fi signal from 100 m up to 25 km, because they focus the power in one direction only.

Bluetooth is a wireless technology standard for exchanging data over short distances. The IEEE standardized Bluetooth as IEEE 802.15.1, but no longer maintains the standard. Since it is designed for short distances, it ranges between 10 m to 100 m and operates in 2.4 GHz frequency band. In [90] a system in which a UAV needs to operate in a short range wireless for executing some tasks it is developed. For this purpose, they proposed a network based on Bluetooth. The goal is to achieve an autonomous flight that is capable of carrying on different flight missions. The network is composed by a master station, which contains the flight controller, and several slave stations deployed along the structure. They proved that it is possible to have reliable communications using the round robin scheduling algorithm together with the proposed platform.

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power and low-bandwidth connections. This technology is suitable for low data rate applications that require battery life and secure networking. Moreover, due to the low power consumption, it allows to establish a long life network and, combined with a mesh network, even to have a long range as well as high reliability. A study in [91] presents a UAV system for indoor localization in a known environment. They used Zigbee together with the inertial navigation system. The results shows that the localization system is viable, effective, flexible, and easily adaptable to various situations.

2.4 Introduction to UWB

Ultra-wideband (UWB) or Impulse Radio UWB (IR-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. Ultra-wideband is based upon the IEEE 802.15.4a standard, also known as 802.15.4-2011. The technology operates on two sets of frequencies, 4.5 GHz and 6 GHz. Features of 802.15.4 include extended range, up to 300 m line of sight, and 40 m non-line of sight. The

ultra-wide band functionality offers immunity to multipath interference while delivering high data rates and low power consumption.

A typical UWB system uses very short pulses on the order of picoseconds or nanoseconds with a very low duty cycle for transmission and reception of the information. We can define the duty cycle as the ratio, in percentage of time, that a pulse is present to the total transmission time. Interestingly, to have a low duty cycle implies a very low transmission power in communication and this leads to a longer battery life for UWB devices.

Another important aspect related to the UWB technologies is the intrinsic security due to their low average transmission power. In fact, an eavesdropper has to be very close to the transmitter (about 1 m) to be able to detect the transmitted information. Moreover, the pulses are modulated uniquely to each transmitter/receiver pair. Since we are talking about picoseconds, it is almost impossible to predict when further pulses will arrive.

Finally, other advantages are: resistance to jamming, high performance in multipath channels, superior penetration properties and simple transceiver architecture.

	IEEE Standard					
	WLAN			Bluetooth	UWB	ZigBee
	802.11a	802.11b	802.11g	802.15.1	802.15.4a	802.15.4
Operational Frequency (GHz)	5	2.4	2.4	2.4	4.5–6	2.4
Maximum Data Rate (Mbps)	54	11	54	1	> 100	0.25
Maximum Range (m)	100	100	100	10	300 [92]	50

Table 2.1: UWB capabilities compared to other IEEE standards.

There are a lot of applications in which the UWB technology can be used. Some of these applications involve: military communications on board helicopters and aircrafts, anticollision vehicular radars [93], through wall imaging used for rescue, security, and medical applications [94], accurately locating a person or object within one inch of its location through any structure [95], detecting land mines [96] and assessing enemy locations and tracking troops [97].

Finally, UWB capabilities compared to the other IEEE standard are shown in Tab. 2.1.

2.5 Terminology

Let us first introduce some terminology. In Chapter 3 we give a little introduction on measurements, in Chapters 4, 5, and 7, we present many localization algorithms in wireless sensor net-

works, and in Chapter 6 we give an introduction to directional antennas. Even though in each Chapter we have reported the adopted terminology, here we itemize for the sake of completeness a brief summary concerning our used notations.

- **static path**: is a precomputed path that the drone follows in the air. The static path comprises many points at which the drone transmits beacons;
- **waypoint**: is a predetermined point along the static path at which the drone sends messages. A waypoint $W = (x, y, z)$ is formally a 3D point where x, y represent the coordinates on the ground and z is the altitude; We refer to $w = (x, y)$ when the waypoint is projected on the ground;
- **angles** $\beta_1, \beta_2, \beta_3$: when three projections of waypoints range a point, they divide the turn angle (2π) into three pairs of angles $\beta_1, \beta_2, \beta_3$ in which $\beta_1 + \beta_2 + \beta_3 = \pi$ (see Fig. 2.9).

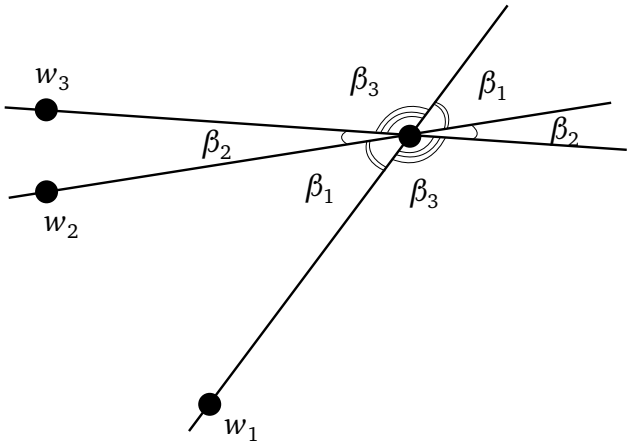


Figure 2.9: $\beta_1, \beta_2, \beta_3$ angles.

- **constraint** $\beta_{\min}$: is the constraint which controls the collinearity of the drone. In the best case the three β_i are equal to $\pi/3$. This constraint allow us to reach a satisfiable level of precision by forcing the minimal angular aperture to a value less than $\pi/3$ but tending to maximum possible.
- **slant distance** (or line-of-sight distance): is the measured 3D distance between the drone in the air and the sensor on the ground;
- **ground distance**: is the 2D projection to the ground of the 3D measured slant distance;
- **constraint** $d_{\min}$: is the constraint which controls the ground precision for a node;

- **constraint** $d_{\mathbf{max}}$: is the constraint which controls the reachability for each point;
- **collinearity**: the estimation of a point is done by performing trilateration that requires three mobile anchor nodes (waypoints). If these three waypoints are placed along the same straight line, the trilateration procedure returns two different points (one real, one "alias");
- **precision**: is the maximum error that can occur;
- **altitude**: is the elevation of the drone above the ground;
- $\overline{PQ}$: is the Euclidean distance between the points P and Q.

Chapter 3

Measurement Precision using IR-UWB technology[1]

3.1 Introduction

In this chapter we will consider the precision in the measurements of the distances from the drone to the terrestrial nodes. From now on in this thesis, we will use as synonyms sensor and Terrestrial Node (TN). By them, we denote a device on the ground that has to be localized. We also consider the impact of the measurement error when the trilateration procedure is accomplished to find the TN position.

Figure 3.1: How to bound the measurement errors?

During the localization procedure, measurement errors are unavoidable, and can seriously impact the localization accuracy. Measurement errors can occur due to instrumental inaccuracies in calculating the distance between the drone and the ground sensors. The instrumental

[1]Part of this work has been accepted for publication to IEEE International Conference on Communications, ICC 2018

inaccuracies depend on the technology. For example, using Wi-Fi the instrumental error may range from 7–10 m, and using Bluetooth it can rise up to 15 m [92]. In Chapter 1 we have seen that Ultra-wideband is a promising radio technology that can use a very low energy level. Since Ultra-wideband provides 7 cm precision with 99.99% reliability [92], we adopt this technology.

Measurement errors are also influenced by the error in the position of the drone itself. For example, the 3DR Solo drone that we used in our experiments has a GPS precision of 1 m [98]. Note that the GPS of 3DR Solo drone has a high accuracy position if we think that, according the United States government, the GPS accuracy precision for smartphones is within 4.9 m [99].

Another source of inaccuracy for the drone position is the altitude. Although in general an altimeter is more accurate than GPS, the main problem with an altimeter is that it relies on a barometer, hence due to weather changes, it needs to be recalibrated.

3.2 Notations and Terminology

Let us first introduce some terminology. Let the *slant distance* denote the 3D distance between the drone and the sensor. The *precision* is the maximum error that can occur. Let ϵ_s denote the *instrumental precision*; in this thesis we consider $\epsilon_s = 0.10$ m because we adopt IR-UWB technology. Although we focus on the IR-UWB technology, the solutions and bounds can be applied to other technology as well, once a suitable value of the precision ϵ_s is provided.

The inaccuracy in the drone's position depends on its drift with respect to its trajectory, and on the changes of its altitude. We say that the drone *rolls* when it drifts to some directions on a fixed plane, and that the drone *uplifts* or *downfalls* when it elevates and decreases its altitude, respectively. Figure 3.2 depicts the cylinder where the drones may reside due to the rolling and altitude errors. The instrumental error ϵ_s is depicted as well.

We denote with γ_d and γ_h, respectively, the *rolling* and the *altitude* precision. Since the GPS precision guaranteed for 3DR Solo is 1 m, we assume $\gamma_d = \gamma_h = 1$ m in our experiments.

Let the *slant* error, E_s, be the 3D measurement error. It is the difference between the *exact* distance $\overline{WP}$ and the *measured* distance $\overline{\widetilde{WP}}$ (see Fig. 3.2). As explained, the measured distance can be different from the exact one due to the instrumental error and the inaccuracy of the drone's position.

To localize the object, we must convert the 3D slant distances into the 2D ground distances. The exact 2D *ground* distance is the distance between the object P and the projection W' on the ground of the exact drone's position W (see Fig. 3.2). Let the *ground* error, E_d, be the 2D measurement error $\overline{PP'}$, where P' is the estimated position of P.

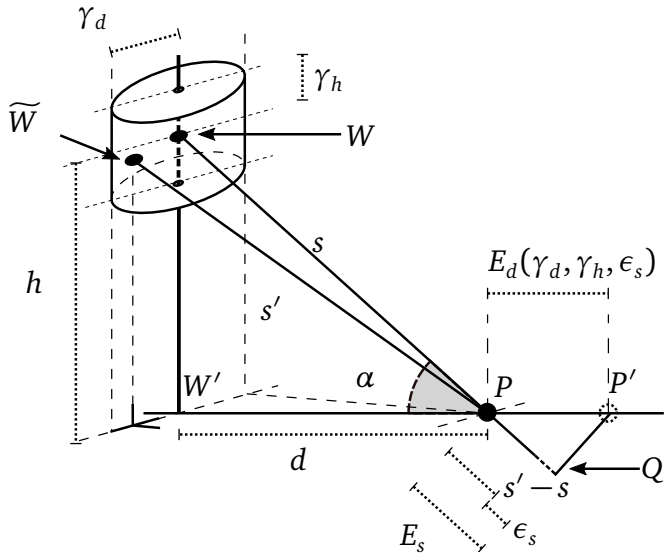


Figure 3.2: The ground error.

Finally, let the *localization* error, E_L, be the localization error that occurs during the trilateration process.

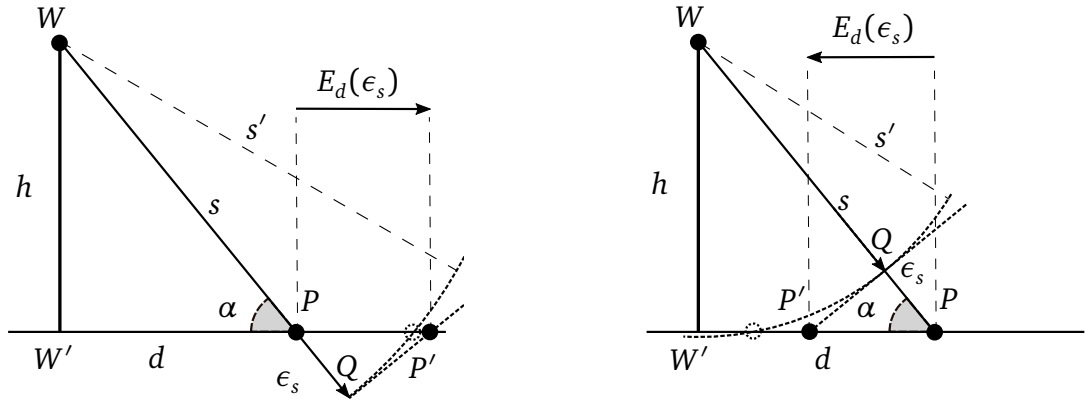
3.2.1 The Ground Error

In this section, we study the ground error E_d. We conduct our analysis by breaking up the error into three components: (1) the *instrumental* precision ϵ_s, (2) the *rolling* precision γ_d, and (3) the *altitude* precision γ_h. Note that each component depends on an independent hardware component, namely IR-UWB, GPS, and barometer, and thus it makes sense to study them separately.

From now on, let α be the angle of incidence between the exact slant distance and the ground level (see Fig. 3.2).

Instrumental component

Here, we investigate the impact of the instrumental error e_s on E_d, with $|e_s| \leq \epsilon_s$. Let s be the exact 3D distance between the drone and the object P (with little abuse of notation, P denotes the position of the object P). Then, let $s' = s + e_s$ be the measure of the segment $\overline{WP}$, where $-\epsilon_s \leq e_s \leq \epsilon_s$. In the following, we geometrically show how the measured slant distance s' is converted into the ground distance. Fig. 3.3 illustrates the reasoning behind the choice of $e_s = \pm\epsilon_s$. We draw a circumference of radius s' centered in the waypoint W. Such circumference will intersect the line that passes between W and P in Q (see Fig. 3.3). Since $s' \neq s$, Q does not coincide with P, and Q is not at the ground level. Specifically, the segment $\overline{PQ}$ of length $e_s = s' - s$



(a) Overestimation (b) Underestimation

Figure 3.3: The instrumental precision ϵ_s and $E_d(\epsilon_s)$.

is on the extension of $\overline{WP}$ if $e_s > 0$; whereas $\overline{PQ}$ is on the radius $\overline{WP}$ if $e_s < 0$.

Since in general $\epsilon_s \ll s$, we can approximate the circumference of radius s' with its tangent at Q. The point P', where the tangent intersects the ground[2], is the estimated position for P according to the measurement s'. Thus, recalling that W' is the projection of W on the ground, $\overline{PP'}$ is the error on the ground derived from the slant error e_s. By elementary geometric rules applied to the right-angled triangle PQP', we obtain: $\overline{PP'} = e_s \cdot \frac{1}{\cos(\alpha)} = e_s \cdot \sqrt{1 + \frac{h^2}{d^2}}$. because $\angle QPP'$ is equal to the incidence angle α. The precision $E_d(\epsilon_s)$ when the instrumental error is maximum and the object is at ground distance d from the drone is given by:

$$E_d(\epsilon_s) = \epsilon_s \cdot \sqrt{1 + \frac{h^2}{d^2}} \tag{3.1}$$

where h is the drone's altitude.

As one can see, the ground precision $E_d(\epsilon_s)$ varies with the distance d on the ground. When $h \neq 0$, the error increases when d decreases (whereas, when $h = 0$, as for a terrestrial anchor, the error does not depend on d). When $h \neq 0$, the worst case occurs when the drone is perpendicular to the point to be measured (i.e., $W' = P$, $d = 0$, $E_d \to \infty$). From this observation, we can assert that, when the measurements are taken by a UAV, rather than a terrestrial anchor, in order to bound $E_d(\epsilon_s)$ from above, it is convenient to add the constraint that all the measurements have to respect a given minimum ground distance.

[2]As one can see from Fig. 3.3, the intersection P' between the tangent and the ground approximates the intersection (white dot) between the circumference and the ground. However, the two intersections become closer and closer when s increases.

Rolling component

In this section, we consider the rolling error. When the drone hovers in position $W = (x, y, z)$, it may not be in W, but rather in position $\widetilde{W}$ due to the GPS precision or the weather conditions (see Fig. 3.4).

To better define the rolling error, we set a 3D-Cartesian coordinate system whose origin is the projection $W' = (0,0,0)$ on the ground of the exact drone's position W, whose x-axis passes through the object to measure P, and z-axis passes through W. Thus, $W = (0,0,h)$ and $P = (d,0,0)$. Then, let the actual drone's position be $\widetilde{W} = (e_x, e_y, h)$, with $-\gamma_d \leq e_x, e_y \leq \gamma_d$, where γ_d is the rolling precision. Obviously, $\widetilde{W}' = (e_x, e_y, 0)$ is the projection of $\widetilde{W}$ on the ground, which is inside a circle of radius γ_d centered at the origin W'. For each point of the circle, it holds $e_x = \gamma_d \cos(\psi)$ and $e_y = \gamma_d \sin(\psi)$, where $\psi = \angle \widetilde{W}' W' P$ and $0 \leq \psi \leq 2\pi$. The measured slant distance s' between $\widetilde{W}$ and P given by:

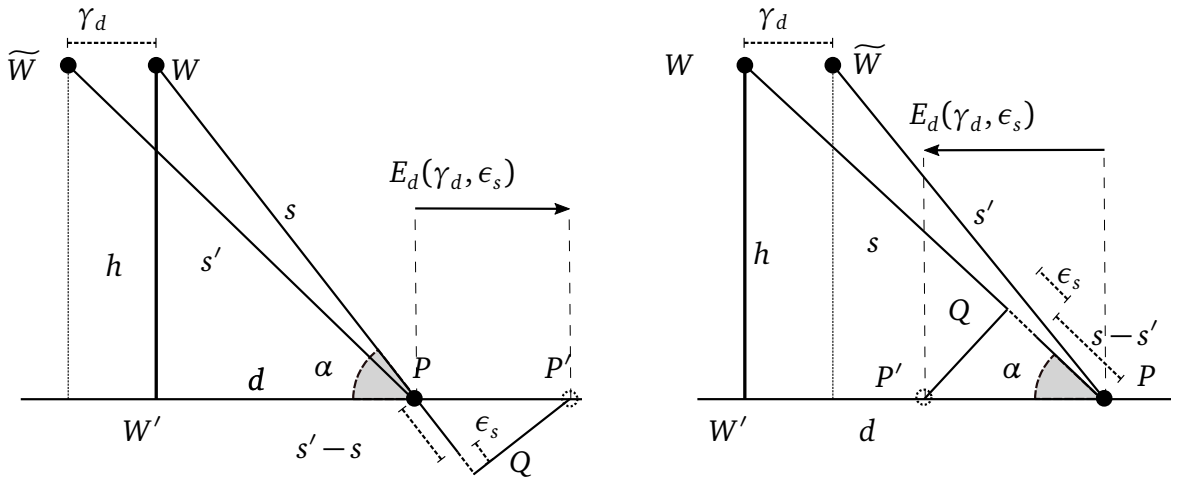
$$
\begin{aligned}
s' &= \sqrt{h^2 + (d - e_x)^2 + e_y^2} \\
&= \sqrt{h^2 + (d - \gamma_d \cos(\psi))^2 + (\gamma_d \sin(\psi))^2} \\
&= \sqrt{h^2 + d^2 - 2d\gamma_d \cos(\psi) + \gamma_d^2 \cos^2(\psi) + \gamma_d^2 \sin^2(\psi)} \\
&= \sqrt{h^2 + d^2 - 2d\gamma_d \cos(\psi) + \gamma_d^2}
\end{aligned}
$$

Recalling that $h > 0$, $d > 0$, $\gamma_d \geq 0$ and $0 \leq \psi \leq 2\pi$, we note that s' is maximum when $f = -2d\gamma_d \cos(\psi)$ is minimum. Including the instrumental error, the slant error is:

$$
E_s = s' - s = \sqrt{h^2 + d^2 - 2d\gamma_d \cos(\psi) + \gamma_d^2} - \sqrt{h^2 + d^2} + \epsilon_s
$$

which is maximum when $\cos(\psi) = -1$ and $\epsilon_s > 0$.

To bring back E_s on the ground, we repeat the same construction as in Subsection 3.2.1. We draw a circumference of radius s' centered in the waypoint W, which intersects the line that passes for W and P in Q. The tangent in Q intersects the ground in the estimated position P'. Again, by elementary trigonometry applied to the right-angled triangle PQP' whose $\angle QPP'$ is equal to the



(a) $e_x = -\gamma_d \cos(\pi)$, $e_y = 0$ (b) $e_x = -\gamma_d \cos(0)$, $e_y = 0$

Figure 3.4: The rolling precision γ_d and ground error $E_d(\gamma_d, \epsilon_s)$.

incidence angle α, we obtain:

$$
\begin{aligned}
E_d(\gamma_d, \epsilon_s) &= \frac{E_s(\gamma_d, \epsilon_s)}{\cos(\alpha)} \\
&= \frac{\left|\sqrt{h^2+d^2-2d\gamma_d\cos(\psi)+\gamma_d^2} - \sqrt{h^2+d^2}\right| + |\epsilon_s|}{\cos(\alpha)} \\
&= \frac{\left|\sqrt{h^2+d^2-2d\gamma_d\cos(\psi)+\gamma_d^2} - \sqrt{h^2+d^2}\right|}{\cos(\alpha)} + \frac{|\epsilon_s|}{\cos(\alpha)} \\
&= \frac{|h^2+d^2-2d\gamma_d\cos(\psi)+\gamma_d^2-(h^2+d^2)|}{\cos(\alpha)\left(\sqrt{h^2+d^2-2d\gamma_d\cos(\psi)+\gamma_d^2}+\sqrt{h^2+d^2}\right)} + E_d(\epsilon_s) \\
&= \frac{|\gamma_d^2-2d\gamma_d\cos(\psi)|\sqrt{h^2+d^2}}{\left(\sqrt{h^2+d^2-2d\gamma_d\cos(\psi)+\gamma_d^2}+\sqrt{h^2+d^2}\right)d} + E_d(\epsilon_s)
\end{aligned}
$$

When $s' > s$ (i.e., $\frac{\pi}{2} < \psi < \frac{3\pi}{2}$), that is, when the drone rolls away from the object, it holds:

$$
E_d(\gamma_d, \epsilon_s) \leq \frac{|\gamma_d^2 - 2d\gamma_d|}{2d} + E_d(\epsilon_s)
$$

and assuming $\gamma_d \ll d$

$$
\leq |\gamma_d| + E_d(\epsilon_s) \tag{3.2}
$$

When $s' < s$ (i.e., $0 \leq \psi \leq \frac{\pi}{2}$ or $\frac{3\pi}{2} \leq \psi \leq 2\pi$), that is, when the drone rolls close to the object,

since $s + s' > s$, we obtain weaker bound:

$$E_d(\gamma_d, \epsilon_s) < \frac{|\gamma_d^2 - 2d\gamma_d|}{d}\frac{s}{s} + E_d(\epsilon_s)$$
$$< 2|\gamma_d| + E_d(\epsilon_s) \tag{3.3}$$

Now, if $\gamma_d \ll d$ holds, $\frac{s}{s'} \to 1$. Since $s + s' \geq 2s'$, we have:

$$E_d(\gamma_d, \epsilon_s) < \frac{|\gamma_d^2 - 2d\gamma_d|}{2d}\frac{s}{s'} + E_d(\epsilon_s)$$
$$< |\gamma_d|\frac{s}{s'} + E_d(\epsilon_s) \to |\gamma_d| + E_d(\epsilon_s) \tag{3.4}$$

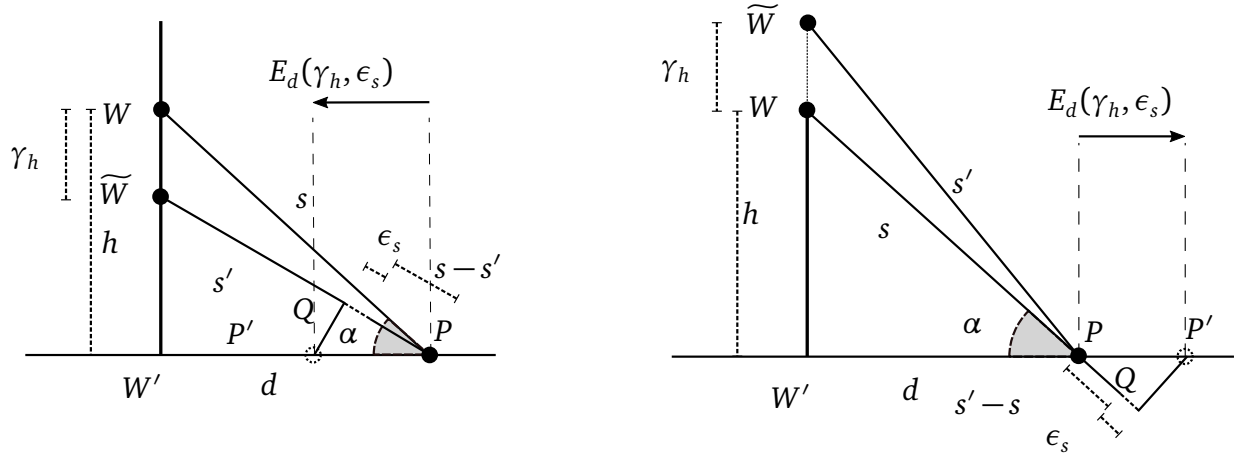
We will see in our experiments that indeed the stricter bound in Eq. (3.4) well approximates the rolling error even when the drone rolls close to the terrestrial node.

Altitude component

When the drone is subject to an uplift (resp., downfall), the measured slant distance s' is overestimated (resp., underestimated) (see Fig. 3.5).

Repeating calculations similar to those above, and assuming that the altitude precision γ_h is very small with respect to d, we find that the ground error can be approximated as:

$$E_d(\gamma_h, \epsilon_s) \approx |\gamma_h|\frac{h}{d} + E_d(\epsilon_s) \tag{3.5}$$



(a) $\gamma_h < 0$ (b) $\gamma_h > 0$

Figure 3.5: The altitude precision γ_h and ground error $E_d(\gamma_h, \epsilon_s)$.

We can summarize our discussions so far as follows.

Theorem 1. *Let ϵ_s, γ_d and γ_h be respectively the instrumental precision, rolling precision, and altitude precision that may affect the slant measurements. Bringing back the distance of the ground, the largest error E_d becomes:*

$$E_d(\gamma_d, \gamma_h, \epsilon_s) \approx |\gamma_d| + \frac{h}{d}|\gamma_h| + |\epsilon_s|\sqrt{1 + \frac{h^2}{d^2}} \tag{3.6}$$

□

Analyzing Eq. (3.6), it is clear that increasing d, the impact of both instrumental and altitude precisions decreases. Note that, E_d cannot be smaller than the rolling precision γ_d. Nonetheless, since the absolute value of γ_d depends only on the GPS and on the flight conditions, i.e., γ_d is independent of d, E_d becomes more and more negligible when d is large.

In conclusion, the ground error can be further bounded by adding the constraint that only ground distances above a certain threshold $d_{\min}$ are considered. If using the drone it is ensured that $d \geq d_{\min}$ always holds, then the *ground precision*, i.e., the maximum error on the ground distance, becomes:

$$E_{d_{\min}}(\gamma_d, \gamma_h, \epsilon_s) \approx |\gamma_d| + \frac{h}{d_{\min}}|\gamma_h| + |\epsilon_s|\sqrt{1 + \frac{h^2}{{d_{\min}}^2}} \tag{3.7}$$

Although the altitude can also be used to control E_d, we do not rely on it because the altitude is a parameter constrained by the environment. As an example, the drone must avoid trees by not flying below 20 m; similarly, it cannot fly above 30 m in an industrial depot or a hangar.

3.2.2 Trilateration Error

This section, discusses the localization error that may affect the estimated position of the terrestrial node when the trilateration procedure is applied.

Let us briefly recall that the trilateration procedure of the object P takes as input the ground distances d_1, d_2, d_3 of P from three waypoints W_1, W_2, W_3 respectively.

Given three ground measures, we compute the estimated position of P as the point (x, y) that minimizes the least squares:

$$\begin{aligned} &\min \quad \delta_1^2 + \delta_2^2 + \delta_3^2 \\ \text{s.t.} \quad &\sqrt{(x_{w_i} - x)^2 + (y_{w_i} - y)^2} + \delta_i = d(w_i, P) \\ &\text{for} \quad i = 1, 2, 3. \end{aligned} \tag{3.8}$$

The procedure returns as the estimated position of the terrestrial node P, the intersection of the three circumferences, corresponding to the radii d_1, d_2, d_3 centered at the projections w_1, w_2,

and w_3 of the waypoints. Due to the ground errors, however, the three circumferences do not intersect at a single point, but they delimit a small *star* area, as depicted in Fig. 3.6. In fact, a pair of extreme circumferences, one obtained by considering the radius affected by the maximum positive E_d error ($d_i + E_d$, measurement overestimation) and one whose radius is affected by the maximum negative E_d error ($d_i - E_d$, measurement underestimation) is drawn in place of each circumference of radius d_i.

Assuming that all the ground distances are sufficiently large compared to the ground error, these extreme circumferences can be linearized (i.e., replaced by the tangent to the radius) without changing significantly the area. Each different non-parallel pair of linearized circumferences intersects at a single point the 12 vertices of the star shape. Note that P is at the center of the star. The trilateration procedure returns as the estimated position, instead of the exact intersection P, a point P' in the star. The point P' is selected by means of the least-squares-error method [100]. The maximum absolute value of the position error, i.e., $\overline{PP'}$, called *localization precision* E_L, occurs when the estimated position P' is at the furthest vertex of the star shape. In other words, the positioning error is bounded by the distance between the center of the star P (i.e., the actual position of the sensor) and its farthest vertex.

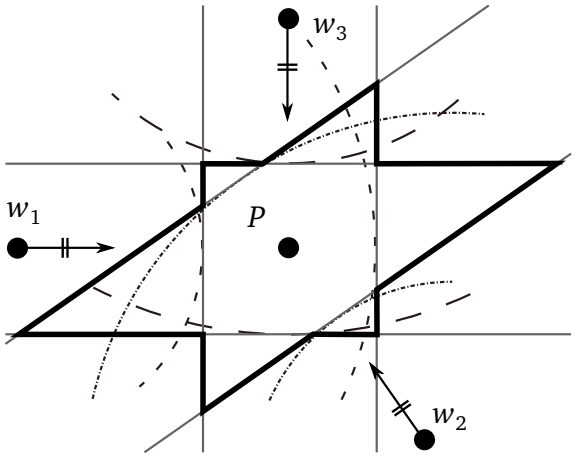
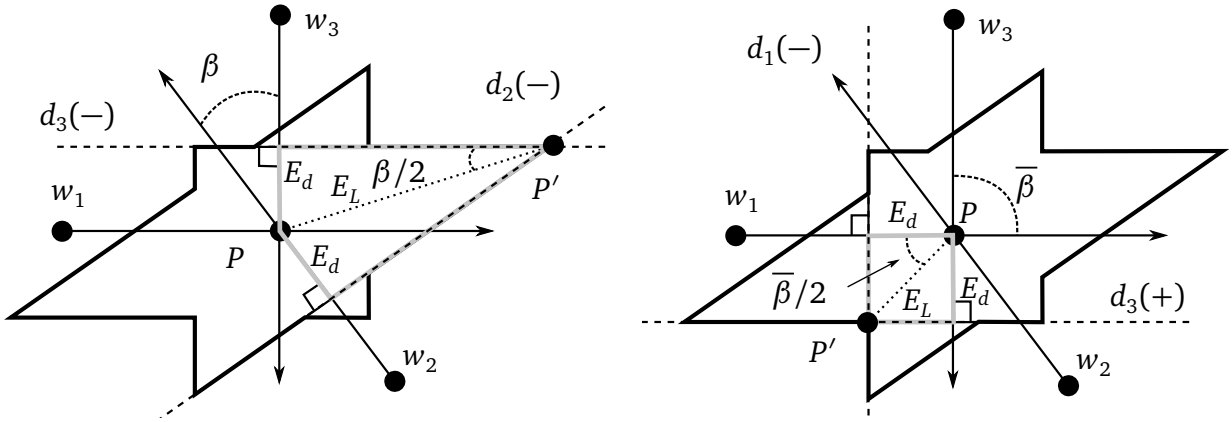


Figure 3.6: Linearization of each measurement.

As an example, in Fig. 3.7a, the distance between the actual point P and the estimated point P' at the intersection of two measurement underestimations $d_2(-)$ and $d_3(-)$ is $\frac{E_d}{\cos(\beta/2)}$, where β is one of the three different angles in which the turn angle (2π) in P is divided by the lines $\overline{w_1P}$, $\overline{w_2P}$, and $\overline{w_3P}$.

In Fig. 3.7b, the distance between P and the vertex P' that results from the measurement underestimation, i.e., $d_1(+)$, and the measurement overestimation, i.e., $d_3(-)$ is depicted. In this case, the distance $\overline{PP'} = \frac{E_d}{\sin(\overline{\beta}/2)}$, where $\overline{\beta}$ is one of the three different angles in which the turn

angle in P is divided by the lines $\overline{w_1P}$, $\overline{w_2P}$, and $\overline{w_3P}$.



(a) Same signs estimation.

(b) Different signs estimation.

Figure 3.7: Localization precision with different conditions.

For each vertex of the star, depending on the signs of the estimations (+ overestimation, − underestimation) of each pair of circumferences, we have: $\frac{E_d}{\sin(\beta_i/2)}$ if the signs are the same; and $\frac{E_d}{\cos(\beta_i/2)}$ if the signs are different, where $\beta_1 \leq \beta_2 \leq \beta_3$ are the three different angles formed in P such that $\sum_i \beta_i = \pi$ (see Fig. 2.9). In the following, we prove that the farthest vertex occurs when the measurement estimations have the same signs and the angle is minimum.

Lemma 1. *Let* $\beta_{min} = \min_i\{\beta_i\}$, $\beta_{max} = \max_i\{\beta_i\}$ *and* $\sum_i \beta_i = \pi$. *Then* $\sin(\frac{\beta_{min}}{2}) \leq \cos(\frac{\beta_{max}}{2})$.

Proof. Let $\beta_{\min} = \min_i\{\beta_i\}$ and $\beta_{\max} = \max_i\{\beta_i\}$. Then, we have: $\beta_{\max} \leq \pi - 2\beta_{\min} \Rightarrow \frac{\beta_{\max}}{2} \leq \frac{\pi}{2} - \beta_{\min}$ from which $\cos(\frac{\pi}{2} - \beta_{\min}) \leq \cos(\frac{\beta_{\max}}{2})$, and thus $\sin(\beta_{\min}) \leq \cos(\frac{\beta_{\max}}{2})$. Since $0 \leq \beta_{\min} \leq \pi/3$, it yields:

$$\sin\left(\frac{\beta_{\min}}{2}\right) \leq \sin(\beta_{\min}) \leq \cos\left(\frac{\beta_{\max}}{2}\right)$$

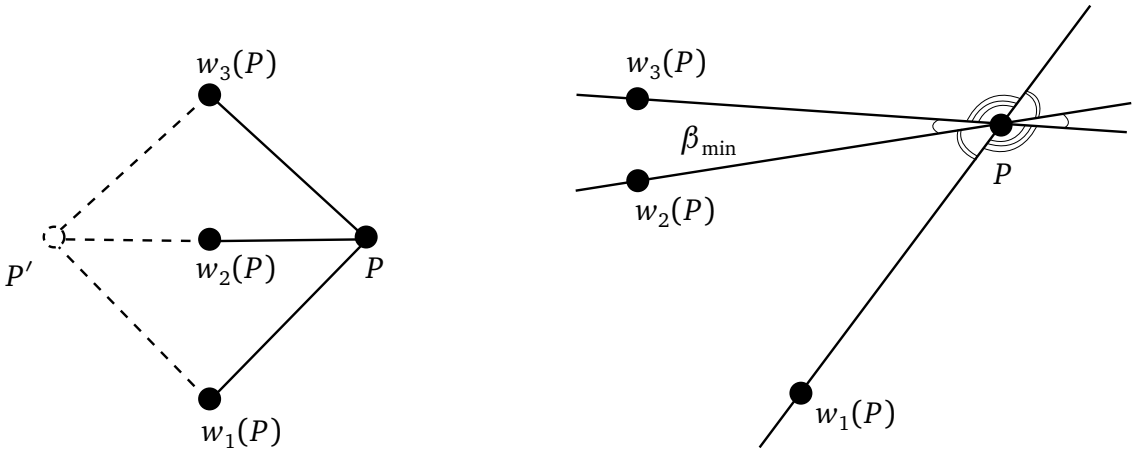
□

Thus, the furthest vertex is at distance $\frac{E_d}{\sin\left(\frac{\beta_{\min}}{2}\right)}$ from P.

Theorem 2. *Given the precisions* ϵ_s, γ_d *and* γ_h, *and recalling that* $E_d(\gamma_d, \gamma_h, \epsilon_s) \approx |\gamma_d| + \frac{h}{d}|\gamma_h| + |\epsilon_s|\sqrt{1 + \frac{h^2}{d^2}}$, *the maximum localization error is obtained as:*

$$E_L(\gamma_d, \gamma_h, \epsilon_s) = \frac{E_d(\gamma_d, \gamma_h, \epsilon_s)}{\sin\left(\frac{\beta_{min}}{2}\right)} \tag{3.9}$$

□



(a) Among the ranging waypoints.

(b) With the node to be localized.

Figure 3.8: Collinearity problems.

Moreover to minimize the localization error due to the trilateration, the three waypoints from which a node is measured cannot be collinear neither among them nor collinear with the sensor [101]. In fact, if the waypoints are collinear among them, the trilateration algorithm cannot distinguish between the real position P of the sensor and the mirror image P' of P (see Figure 3.8a).

With regard to the collinearity with the node (see Figure 3.8b), if $\beta_{\min}$ tends to 0 then the localization error becomes very large. Thus a good localization aims at a large value of $\beta_{\min}$.

Therefore, from Eq. (3.9), we learn that, given a certain ground error, the localization error is minimized when $\beta_{\min} \to \frac{\pi}{3}$. From now on, we refer to this desirable condition with the expression "the waypoints surround the terrestrial node". Finally Fig. 3.9 reports the localization precision E_L computed by varying both the values d, where $0 \leq d \leq 60\,\text{m}$, and $\beta_{\min}$, where $5° \leq \beta_{\min} \leq 60°$, and assuming only the instrumental error, that is, $\gamma_d = \gamma_h = 0\,\text{m}$ and $\epsilon_s = 0.10\,\text{m}$. It is worthy to note that the localization precision E_L is strongly influenced by a bad angle. As expected, when both d and $\beta_{\min}$ tend to 0, E_L grows quickly.

3.3 Experimental Evaluation

In this section, we propose two sets of experiments. The first set tests the goodness of the approximation of the instrumental, rolling and altitude errors derived in Eq. (3.1), Eq. (3.2) and Eq. (3.5). These experiments are performed by emulating the rolling and altitude errors by arranging the IR-UWB antennas in appropriate positions and taking the measurements from such known positions. The second set of experiments, instead, use the drone that hovers in Assisi, Italy.

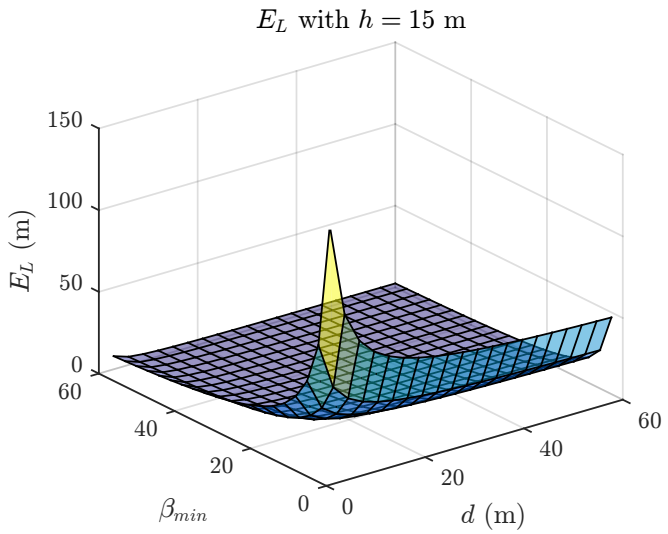


Figure 3.9: The localization precision E_L.

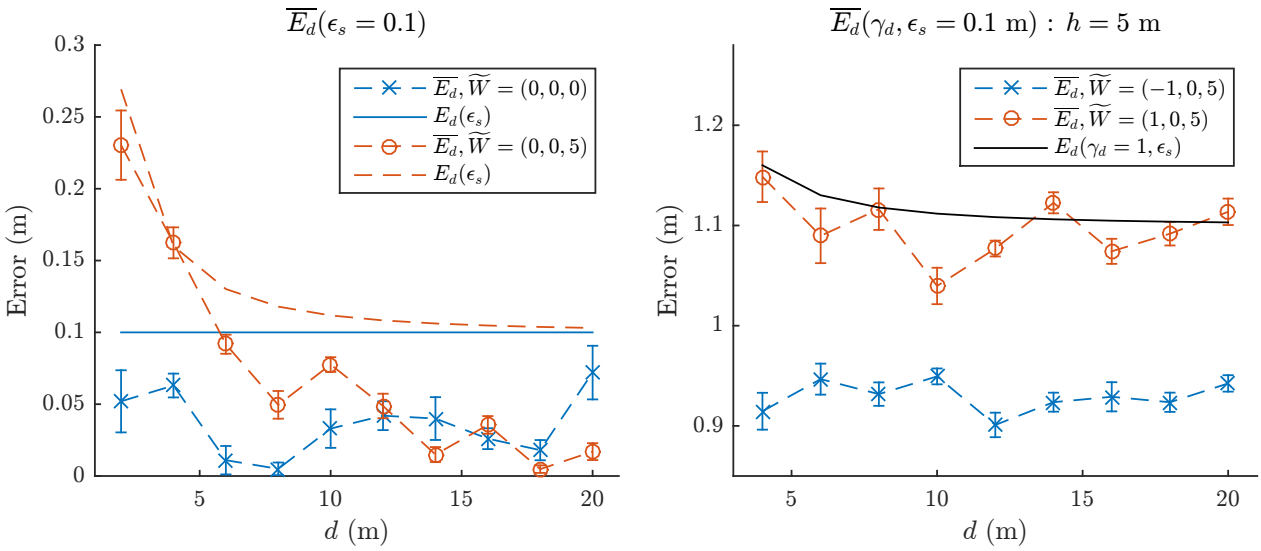
In this case, although we do not know the exact values of γ_d and γ_h, we observe that the error curves behave as expected. Indeed, they follow the two fundamental lessons learned here:

- **Lesson 1**: the error is small when the ground distance between the drone and the node is large, and
- **Lesson 2**: the error is small when the waypoints used to measure a node surround it.

3.3.1 Pre-arranged Antenna Experiments

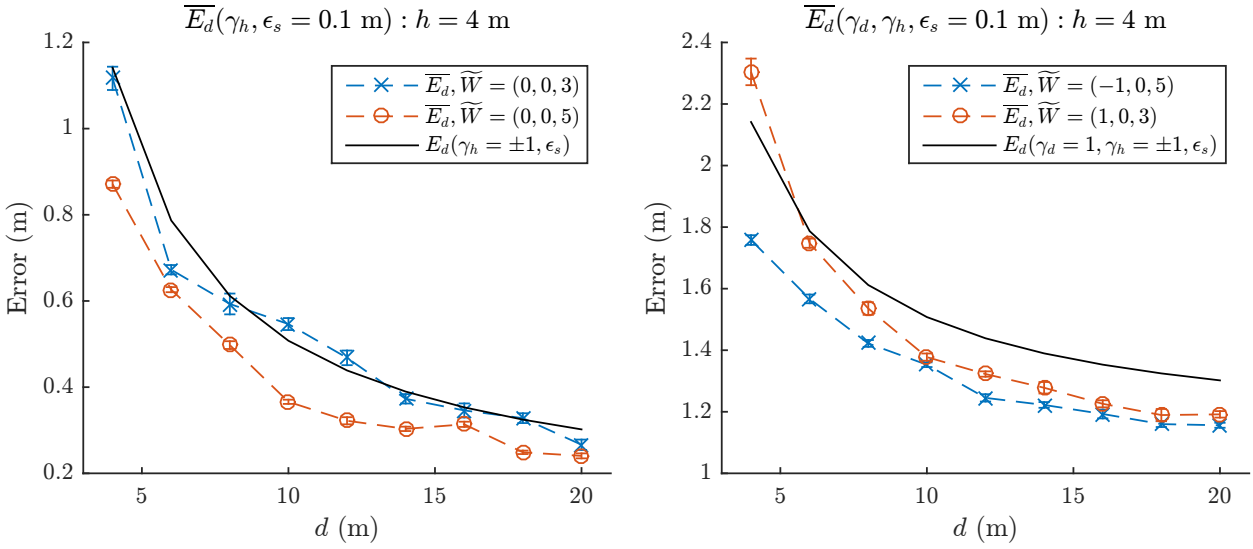
First, we validate the ground error approximation E_d with a set of experiments in a real scenario assuming that we know the position from which the drone takes the measurement and we interpret it as a position subject to a rolling and/or altitude misplacement.

Precisely, we start with the two IR-UWB antennas at the ground level at two fixed positions with a predefined distance d, say $W' = (0,0,0)$ and $P = (d,0,0)$. Next, we move one antenna perpendicularly to its position on the ground to reach a certain altitude, $W = (0,0,h)$. Then we can either simulate the rolling error by displacing the antenna in a circle around W, or simulate the altitude error by changing the altitude. The new position $\widetilde{W}$ is interpreted as the position of the drone subject to an error. As an example, Fig. 3.4a exemplifies $W = (0,0,h)$, $W' = (0,0,0)$, $P = (d,0,0)$, and $\widetilde{W} = (-1,0,h)$. We measure the slant distances s' from $\widetilde{W}$ and report them on the ground by computing $\sqrt{s'^2 - h^2}$. We compare such ground distances with the exact ground distance d. We compute the difference $\overline{E_d}$ between the computed ground distances and d, and compare it with E_d. We repeat the experiments by varying d from 0 to 20 m.



(a) Instrumental error.

(b) Rolling error.



(c) Altitude error.

(d) The experimental combined error.

Figure 3.10: The experimental error and the theoretical error in different cases.

To verify Eq. (3.1), we have measured, and reported in Fig. 3.10a, the error $\overline{E_d}(0.10)$ when the antennas are at $\widetilde{W} = W$. The measurements are affected only by the instrumental error. The experiment is repeated at altitude $h = 0\,\text{m}$ and $h = 5\,\text{m}$. For both altitudes, the experimental error $\overline{E_d}$ is bounded from above by Eq. (3.1). When $h = 0\,\text{m}$, we indeed measure the antenna precision, and our experiments perfectly match the precision 0.07 m declared by DecaWave [92] for outdoor measurements. When $h = 5\,\text{m}$ and d tends to 0 m, the error is twice or more than that at $h = 0\,\text{m}$. As expected, the errors for $h = 0\,\text{m}$ and $h = 5\,\text{m}$ tend to coincide when d increases.

To verify Eq. (3.2), we have measured and reported in Fig. 3.10b the error $\overline{E_d}(\gamma_d, \epsilon_s)$, when $\widetilde{W} = (\pm 1, 0, 5)$ and the strict bound in Eq. (3.2). It is worthy to note that the error tends to $|\gamma_d|$.

To verify Eq. (3.5), we have measured, and reported in Fig. 3.10c the error $\overline{E_d}(\gamma_h, \epsilon_s)$, when $\widetilde{W} = (0, 0, 4 \pm 1)$. As can be seen, the approximation for the ground error given in Eq. (3.5) satisfactorily estimates the experimental error, especially when $h = 5\,\text{m}$. Moreover, the altitude error becomes smaller when d increases as expected from Eq. (3.5).

We finally measured and reported in Fig. 3.10d the error $\overline{E_d}(\gamma_d, \gamma_h, \epsilon_s)$ when $\widetilde{W} = (\pm 1, 0, 4 \pm 1)$ along with the bound in Eq. (3.7). The curves almost coincide.

Figure 3.11: Our settings during experiments with the drone.

3.3.2 Drone Experiments

In these experiments, we consider two points $P = (d, 0, 0)$ and $W' = (0, 0, 0)$ on the ground. We fasten one antenna to the drone, and another antenna at P connected to a laptop (see Fig.

3.11). Then, we set the GPS flight mode for the drone and set the drone's altitude at h. During the experiment, when the drone flies over the GPS position that corresponds to $W = (0, 0, h)$, the drone, by means of the antennas, measures the slant distances between W and P and collects them in the laptop which converts the measured slant distances s' to ground distances by computing $\sqrt{s'^2 - h^2}$. The experimental error $\overline{E_d}$ is then the difference between the computed ground distance and the exact distance d between W' and P. Fig. 3.12 reports $\overline{E_d}$ when $3 \leq d \leq 40\,\text{m}$. The experiment is repeated with the drone that flies at altitudes $h = 15\,\text{m}$ and $h = 30\,\text{m}$. At least 10 measurements are taken for each value of d, and the average value along with its 95% confidence interval are plotted.

Since the drone's position is affected by the wind, by the strength of the propellers and by the GPS error, we know that s' is indeed the slant distance between P and the current $\widetilde{W}$ position of the drone. As we know from Eq. (3.7), the error $\overline{E_d}$ increases when h increases and when d tends to 0.

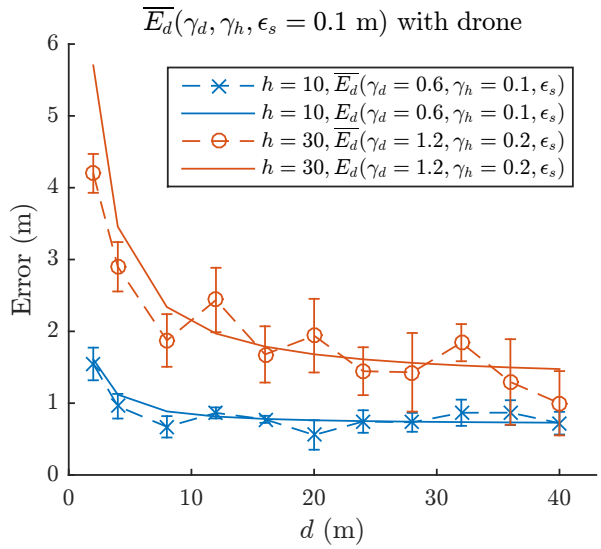
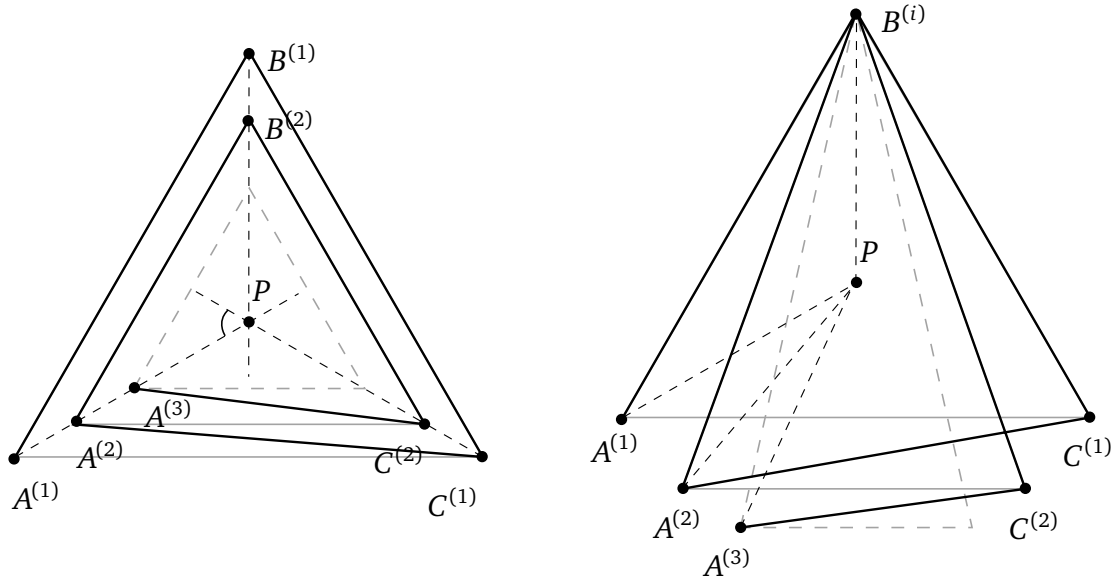


Figure 3.12: The experimental error $\overline{E_d}$ and the approximation E_d.

In Fig. 3.12, we also plot E_d (Eq. (3.7)) using values $\gamma_d = \{0.6, 1.2\}$ and $\gamma_h = \{0.1, 0.2\}$ that empirically fit the experimental curves. We fixed $\epsilon_s = 0.10\,\text{m}$. We do not give any special meaning to these values of the precision, except that they corroborate our belief that the experimental curve and the theoretical curve of the combined error have the same behavior. So, if the drone manufacturers make available the precision values for their products under different flight conditions, we believe that Eq. (3.7) provides a good approximation of the measurement error.

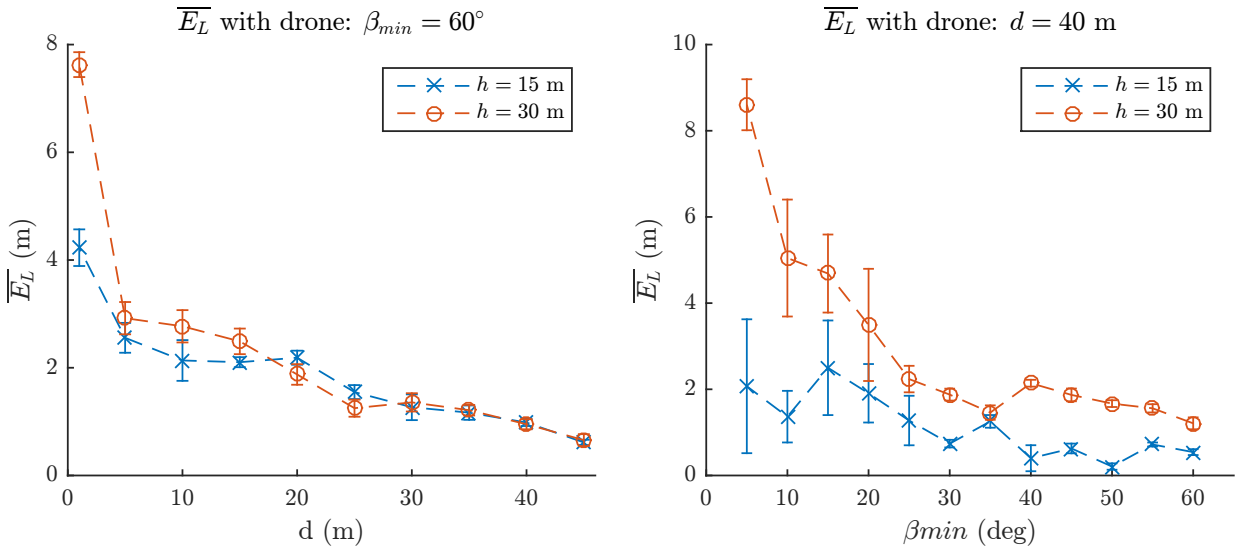
Finally, as regards to the localization error when the trilateration method is applied, we have performed two experiments. In the first experiment, we fix the minimum angle $\beta_{\text{min}} = 60°$ and



(a) Same minimum angle.

(b) Same distance.

Figure 3.13: Localization precision with different conditions.



(a) d varies.

(b) β_{min} varies.

Figure 3.14: Experimental localization.

vary the ground distance $1 \leq d \leq 45\,\text{m}$. To carry out the experiment, the drone follows a path that passes through the vertices of several equilateral triangles that share the same incenter.

The node P to be localized is placed at the incenter. The drone takes the measurements at the triangles' vertices (see Fig. 3.13a). A trilateration of P is performed every time the drone collects the three measurements corresponding to the three vertices of a triangle. Fig. 3.14a plots the experimental difference $\overline{E_L}$ between the estimated position of P and the exact position P. As depicted in Fig. 3.14a, according to Eq. (3.9), $\overline{E_L}$ increases when d decreases, even though the minimum angle is fixed at the best possible value 60°. The values in Fig. 3.14a almost double those in Fig. 3.13a since $\sin(30°) = 1/2$.

In the second experiment, we fix $d = 40\,\text{m}$ and vary the minimum angle such that $5° \leq \beta_{\min} \leq 60°$. To carry out the experiment, the drone follows a path that passes through the vertices of several isosceles triangles sharing one vertex and the circumcenter. While the base of the isosceles triangles becomes shorter and the two waypoints that delimit the base tend to be collinear with P, $\beta_{\min}$ decreases. The node P to be localized is placed at the circumcenter of the triangles. As in the first experiment, a trilateration of P is performed every time the drone collects the three measurements corresponding to the three vertices of each triangle (see Fig. 3.13b). According to Eq. (3.9), the error increases when $\beta_{\min}$ decreases.

Conclusion In this work, we presented an analysis of the errors that can affect the measurements taken by a flying anchor, like a drone. We have learned two important lessons: The measurement errors can be bounded if (i) the drone and the terrestrial node are sufficiently apart on the ground, and (ii) the waypoints that perform the measurement of a terrestrial node surround it.

Chapter 4

Precise Localization with Omnidirectional Antennas[1]

4.1 Introduction

In this work, we propose a localization algorithm that uses a mobile anchor node which is a drone. The drone follows a static path precomputed and as such it localizes all the TNs present in the deployment area. On the static path the drone stops at the waypoints where it takes the measurements.

Figure 4.1: A classical omnidirectional antenna.

The waypoints are selected obeying to the two important lessons we learn in Chapter 3. Our

[1]Part of this work has been published to MobiWac '16-Proceedings of the 14th ACM International Symposium on Mobility Management and Wireless Access

goal is to localize all the TNs with a precision guaranteed. Namely, we think that the precision is related to the application and as such the desired precision is given as input to the problem. We show that our algorithm is able to guarantee the desired precision by taking the measurements sufficiently apart on the ground from the TNs. Our solution considers, for the sake of simplicity, that the measurements are affected only by the instrumental error.

4.2 Model and Assumptions

We consider a network of n sensors, (or simply, n nodes), deployed in a rectangular area Q of size $Q_x \times Q_y$, where $Q_x \geq Q_y$ without loss of generality. For the sake of simplicity, we assume the area is obstacle free and the ground is flat so that the drone is always in the line of sight with the nodes to be measured. Our main goal is to localize, with a user-required precision, each sensor inside the deployment area using a drone equipped with omnidirectional antennas.

The drone acts as a flying anchor. It flies at a given altitude h and it follows a precomputed static path. We assume that the drone can measure its altitude with negligible error, i.e., considering $\gamma_h = 0\,\text{m}$ and also $\gamma_d = 0\,\text{m}$. A technology allowing for such precise altitude measurements is Differential GPS [102]. We also assume that the drone infers the distance measurements from the round-trip time of messages exchanged with the nodes on the ground. To have a good measurement precision of the order of 10 centimeters, we adopt the Impulse-Radio Ultra-Wide-Band (IR-UWB) technology for the antennas [92].

The drone takes measurements at pre-established points of the static path. For each point P the ranging waypoints of P in 3D are denoted as $W_1(P), W_2(P), \ldots, W_m(P)$. In the following we consider these ranging waypoints projected on the ground and we denote them $w_1(P), w_2(P), \ldots, w_m(P)$ (see Section 2.5 for terminology). The ground is assumed to be flat enough to allow the drone to be always in the line of sight with the node to be measured.

To take a measurement, the drone acts as follows. At the waypoint the drone sends beacons with a unique identifier which depends on the coordinates of the waypoint, the current timestamp and the orientation used to send such a beacon. Each sensor on the ground that can hear a beacon replies to the drone with an ack message that contains its ID, the current timestamp, and the identifier of the beacon received. The antenna (for simplicity, we say the drone) computes the distance between itself and the node using the round-trip time of the beacon message and then sends a message with the computed distance to the sensor.

The drone and the sensor have their communication range, r_{drone} and r_{node}, respectively. Since a message can be exchanged between the drone and the sensor only if they can hear each other,

the communication range is $r = \min\{r_{\text{drone}}, r_{\text{node}}\}$. The drone measures the (3D) *slant distance* s, that is, the line-of-sight, between itself and the measured node. By simple geometric argument, since we assume negligible error in altitude, it is easy to see the slant distances when projected on the ground cannot be larger than

$$d_{\max} = \sqrt{r^2 - h^2} \tag{4.1}$$

because the drone flies at altitude at most h, r is the minimum communication range between the devices, and the slant measured distances verify whether $s \leq r$. Therefore, the measured ground distances d cannot be larger than $d_{\max}$.

In this research, whenever we say distances on the ground we mean that the slant distances, taken in 3D because the drone is flying at a certain altitude, are projected on the ground (the plane) where the sensors reside.

Slant distances are affected by errors that depend on the adopted technology, i.e., IR-UWB in our case [92]. To bound the ground precision ϵ_d, we adopt the constraint that all the distances on the ground must be no smaller than $d_{\min}$. In other words, a sensor P can only be measured by waypoints which are at distance at least $d_{\min}$ from P.

4.3 The OMNI algorithm

In this section, we describe the proposed directional localization algorithm OMNI. From now on, node P is short form for the sensor that resides at point P in Q. Since, in principle any point P in Q is candidate to contain a sensor, with a little abuse of notation, we denote P indistinctly as the point P or the node P.

The aim of the OMNI algorithm is to obtain *precise localization* for each point P in Q.

Definition 1. *The node P is precisely localized if the drone chooses three ranging waypoints $w_1(P)$, $w_2(P)$ and $w_3(P)$ for P such that they satisfy the following constraints (see Section 2.5 for a quick check of the terminology):*

1. *the d_{max} constraint, which controls the reachability for each point P in Q: $\overline{w_i(P)P} \leq d_{max}$ for $i = 1, 2, 3$;*

2. *the d_{min} constraint, which controls the ground precision ϵ_d for P in Q: $\overline{w_i(P)P} \geq d_{min}$ for $i = 1, 2, 3$;*

3. *the β_{min} constraint, which controls the collinearity of the drone with P;*

4. *the non-linearity constraint which controls the collinearity among waypoints:* $w_1(P)$, $w_2(P)$ *and* $w_3(P)$ *cannot belong to the same straight line.*

4.3.1 Generating the Static Path

The static path D consists of vertical scans connected by horizontal scan, as depicted in Fig. 4.2. The waypoints belong only to the vertical scans of D. The distance between two consecutive waypoints is fixed to I_w. The waypoints along the vertical scans are numbered bottom-up, starting from 0. We say that a waypoint is *even* if it gets an even number. Each vertical scan continues beyond the top and bottom borders for a segment of length $F_y = (d_{\max} - I_w)\frac{\sqrt{3}}{2}$ and it is rounded to start and finish with an even waypoint. Thus, each vertical scan has length $L = 2I_w \left\lceil \frac{Q_y + 2F_y}{2I_w} \right\rceil$.

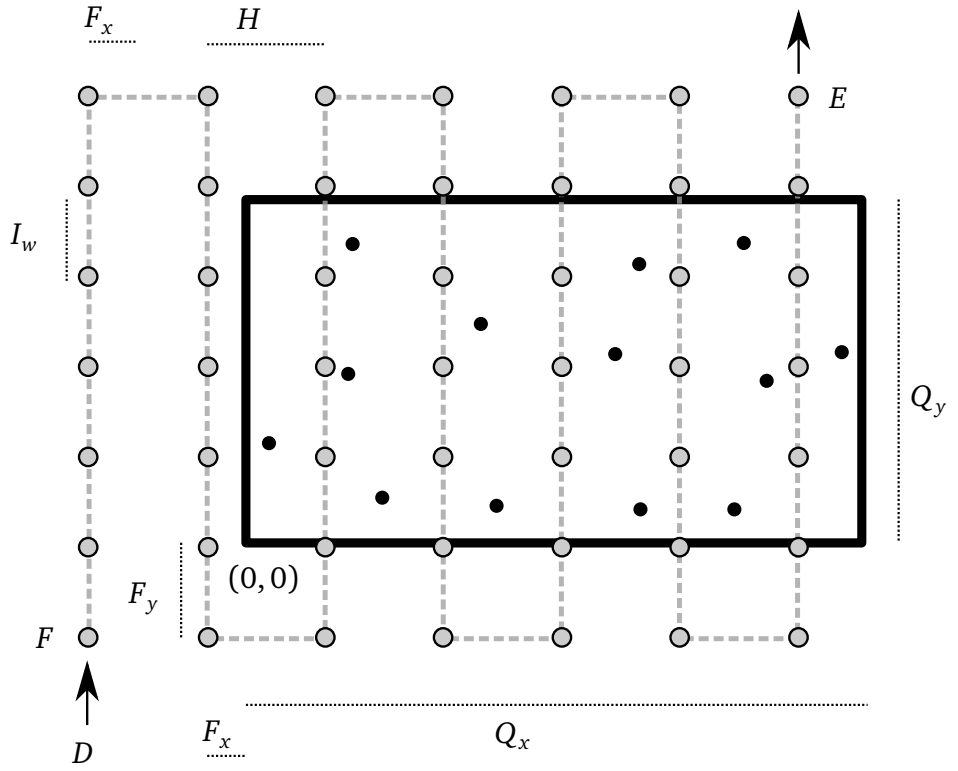


Figure 4.2: The static path (dashed line) D and the deployment area Q.

Two consecutive vertical scans are connected by a horizontal segment of length $H = (d_{\max} - d_{\min} - 2I_w)\frac{1}{2}$. Let H be termed the *inter-scan* distance. The drone traverses one vertical scan of D bottom-up and the next one top-down. The horizontal segments are traversed from left to right (see Fig. 4.2). The starting point of D is $S = (s_x, s_y) = (-F_x - H, -F_y)$, where $F_x = \frac{d_{\min}}{2}$. From the starting point, we can take correct measurements for the bottom left corner of Q. Note that neither F_x or H are rounded to be a multiple of I_w because there are no waypoints on the horizontal segments of D.

The static path that the drone follows into the sky is at a height h above the plane of the deployment area of sensors. The path D consists of two vertical scans outside Q that are used to measure the nodes in the leftmost stripe of Q, followed by several vertical scan inside Q. The last vertical scan is at distance no larger than $\lfloor \frac{d_{\max}-2I_w}{2} \rfloor$ from the rightmost vertical border of the deployment area Q.

For each node, we select the three waypoints in two consecutive vertical scans, to avoid the "alias" problem due to waypoints collinearity. To limit the selection of the three waypoints to two consecutive vertical scans, we assume $H \geq \frac{d_{\min}}{2}$. Therefore, we add the constraint:

$$d_{\min} < \frac{d_{\max}}{2} - I_w. \tag{4.2}$$

4.3.2 Tessellation and Ranging Waypoints

In this section, we explain how to select on D the ranging waypoints that correctly measure each possible node in Q.

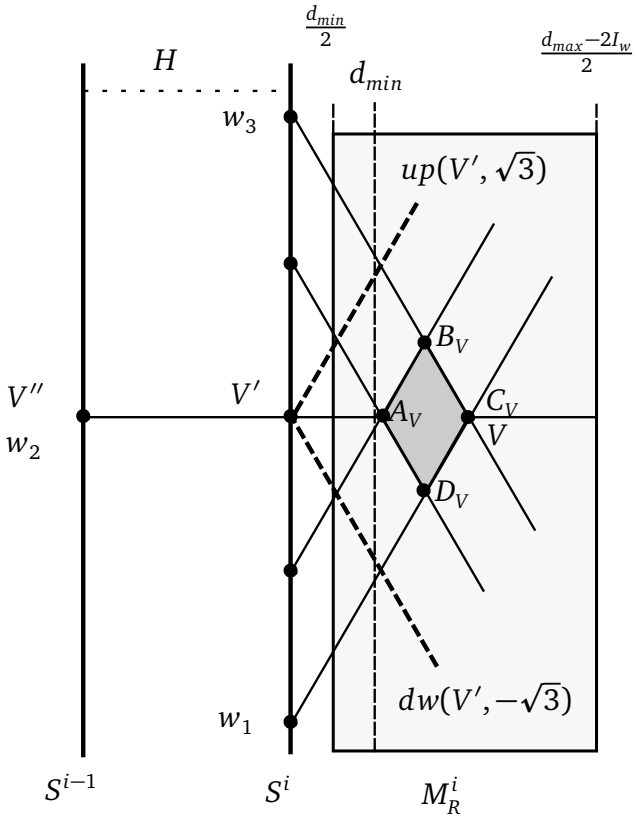


Figure 4.3: The tessellation and, in gray, the stripe M_R^i.

Let the vertical scans in D be numbered from 0 to ν. Consider on D two consecutive vertical scans S^{i-1} and S^i, with $1 \leq i \leq \nu$. Let x_{S^i} be the x-coordinate of the vertical scan S^i of D. In

the following, we prove that the points of Q on the right of S^i whose x-coordinates belong to the interval $\left[x_{S^i} + \frac{d_{\min}}{2}, x_{S^i} + \frac{(d_{\max} - I_w)}{2}\right]$ form the stripe of Q, named M_R^i and illustrated in Fig. 4.3, that can be correctly measured from S^{i-1} and S^i.

To decide how to associate the waypoints on S^i and S^{i-1} with the nodes in Q, we logically tessellate M_R^i as follows. From every even waypoint $w = (x_w, y_w) = (x_w, 2tI_w)$ on S^i, with $0 \leq t \leq \lceil \frac{L}{2I_w} \rceil$, we draw the two lines that pass through w with slope $m_1 = \sqrt{3}$ and $m_2 = -\sqrt{3}$ denoted, respectively, as $up(w, m_1)$ and $dw(w, m_2)$ (see Fig. 4.3).

Such lines design a lattice on M_R^i, made of diamond shapes. Each vertex of the lattice is uniquely associated with the diamond on its left.

Precisely, consider the vertex $V = C_V = (x_V, y_V)$ in Fig. 4.3. It is associated with the diamond $\Delta(V)$ (displayed in dark gray) of vertices (listed in clockwise order starting from V): $C_V = V$, $D_V = \left(x_V - \frac{I_w}{\sqrt{3}}, y_V - I_w\right)$, $A_V = \left(x_V - 2\frac{I_w}{\sqrt{3}}, y_V\right)$, and $B_V = \left(x_V - \frac{I_w}{\sqrt{3}}, y_V + I_w\right)$.

Observe that all the diamond shapes of the tessellation have the same size. For every diamond $\Delta(V)$, the horizontal diagonal $A_V C_V$ has length $|A_V C_V| = \frac{2I_w}{\sqrt{3}}$ while the vertical diagonal $B_V D_V$ has length $|B_V D_V| = 2I_w$. Moreover, the projections V' and V'' of V are waypoints on S^i and S^{i-1} because their y-coordinate is a multiple of I_w.

Let V be at the intersection of the lines $up(V_S, \sqrt{3})$ and $dw(V_N, -\sqrt{3})$. V_S and V_N play a very important role in our localization technique, along with the projection V' of V on S^{i-1}: Indeed, $w_1(P) = V_S$, $w_2(P) = V'$, and $w_3(P) = V_N$ for each P in the diamond $\Delta(V)$, represented by V.

Theorem 3. *Fixing the vertex $V = (x_V, y_V) \in M_R^i$ that occurs at the intersection of the lines $up(V_S, \sqrt{3})$ and $dw(V_N, \sqrt{3})$, the nodes in $\Delta(V)$ are correctly measured by the ranging waypoints V_S, V_N and the projection V'' of V on S^{i-1}.*

Proof. Fix $w_1(P) = V_S$, $w_2(P) = V''$, and $w_3(P) = V_N$. First observe that since V occurs at the intersection of the lines $up(V_S, \sqrt{3})$ and $dw(V_N, -\sqrt{3})$, by elementary geometric arguments it holds that $d(V_S, V) = d(V_N, V) = 2d(V', V) = 2(x_V - x_{S^i})$, where V' is the projection of V on S^i. Thus, recalling that V belongs to the stripe M_R^i on the right of S^i that is, $\frac{d_{\min}}{2} \leq x_V - x_{S^i} \leq \frac{(d_{\max} - 2I_w)}{2}$, it immediately follows that

$$d_{\min} \leq d(w_1(V), V) \leq d_{\max} - 2I_w.$$

Moreover, $d_{\min} < H + \frac{dmin}{2} < d(w_2(V), V) \leq H + \frac{d_{\max} - 2I_w}{2} < d_{\max}$. Thus, the $d_{\min}$ and $d_{\max}$ constraints are satisfied for V.

To prove the $d_{\min}$ and $d_{\max}$ constraints for each internal node $P \in \Delta(V)$, let us first consider each node P in the lower half-diamond $\Delta(V)$, that is, the triangle $A_V C_V D_V$ of $\Delta(V)$. Let P' be the projection of P on S^i and let $\angle P' V_S P$ be the angle at V_S between the ray $V_S P$ and the vertical scan

S^i. Since $d(V_S, P') \leq d(V_S, V')$ because P belongs to the triangle $A_V D_V C_V$ and $\angle P'V_S P \leq \angle V'V_S V$, it holds that:

$$\begin{aligned} d(w_1(V), P) &= \frac{d(w_1(V), P')}{\cos(P'w_1(V)P)} \\ &\leq \frac{d(w_1(V), V')}{\cos(P'w_1(V)P)} \\ &\leq \frac{d(w_1(V), V')}{\cos(V'w_1(V)V)} \\ &= d(w_1(V), V) \\ &\leq d_{\max} - 2I_w \end{aligned}$$

Moreover, for each P in the triangle $A_V B_V C_V$ of $\Delta(V)$, consider the point P'' symmetric to P with respect to the line $V'V$ which belongs to the triangle $A_V D_V C_V$.

By the triangle inequality, given that V belongs to M_R^i and $d(P'', P) \leq d(B, D)$, we have:

$$\begin{aligned} d(w_1(V), P) &\leq d(w_1(V), P'') + d(P'', P) \\ &\leq d(w_1(V), P'') + 2I_w \\ &\leq d(w_1(V), V) + 2I_w \\ &\leq d_{\max} \end{aligned}$$

Repeating the same reasoning, using $w_3(P) = V_N$ in place of $w_1(P) = V_S$, it can be proved that for each $P \in \Delta(V)$, $d_{\min} \leq d(V_N, P) \leq d_{\max}$.

With regard to the $d_{\min}$ and $d_{\max}$ constraints for $w_2(P)$, let Z be the projection of $P \in \Delta(V)$ on the diamond diagonal AC. It easily follows that $d(w_2(V), P) \leq d(w_2(V), Z) + d(Z, P) \leq d(w_2(V), V) + I_w < H + \frac{d_{\max}}{2} - I_w < d_{\max}$, and $d(w_2(V), P) \geq d(w_2(V), Q) \geq H + \frac{d_{\min}}{2} > d_{\min}$ because $H > \frac{d_{\min}}{2}$.

For the $\beta_{\min}$ constraint in vertex V, observe that the tessellation divides the turn angle in V in equal angles. Therefore, $\beta(V) = \frac{\pi}{3}$. Since $\tan(\frac{\beta(V)}{2}) = \frac{1}{\sqrt{3}}$, the $\beta_{\min}$ constrained is satisfied.

In each diamond $\Delta(V)$,$\min_{P \in \Delta(V)} \beta(P) = \beta(A_V)$.

Then, for each $P \in M_R^i$, $\beta_{\min}$ occurs at the vertex A_V with minimum x-coordinate. Moreover,

$$\frac{\beta_{\min}}{2} = \tan^{-1}\left(\frac{\frac{d_{\min}}{2}}{\sqrt{3}\frac{d_{\min}}{2} + 2I_w}\right) \tag{4.3}$$

Figure 4.4 illustrates the width of the angle $\beta(P)$ in a diamond shape of M_R^i and gives evidence that $\beta(A_V) = \min_{P \in \Delta(V)} \beta(P)$. Figure 4.5 shows the fact that increasing the distance of V from S^i, $\beta(A_V)$ increases.

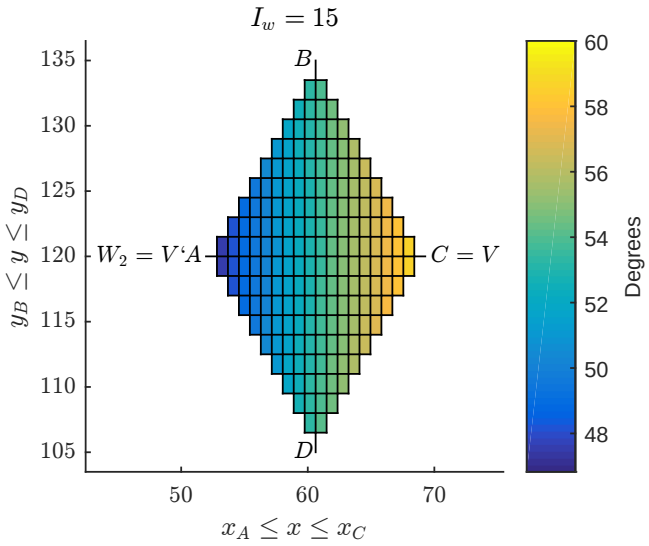


Figure 4.4: The angular aperture $\beta(P)$ for $P \in \Delta(V)$ with V at distance 70 m.

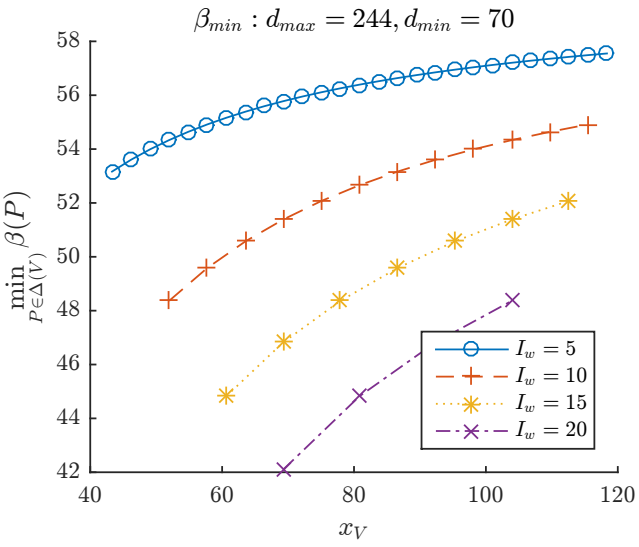


Figure 4.5: The width of $\beta(A_V) = \min_{P \in \Delta(V)} \beta(P)$ versus the x_V.

Since we proved that from S^{i-1} and S^i the nodes in M_R^i are correctly measured, considering all the pairs of consecutive scans (S^0,S^1), (S^1,S^2) ... ($S^{\nu-1}$,S^ν) we cover $\cup_{i=1}^{\nu} M_R^i = Q$. Hence, all the nodes in Q can be correctly measured from the vertical scans $S^0, \ldots, S^\nu$. □

Now, from Eq. (3.9) and considering Eq. (4.3), we are ready to state the following:

Theorem 4. *The localization precision in* OMNI *can be expressed as:*

$$\epsilon_L = \epsilon_s \cdot \frac{\sqrt{\left(1+\frac{h^2}{d_{min}{}^2}\right)\left(1+\tan^2\left(\frac{\beta_{min}}{2}\right)\right)}}{\tan\left(\frac{\beta_{min}}{2}\right)} \tag{4.4}$$

□

4.3.3 Designing the Drone's Mission

In this section, we describe the drone's behaviour. The drone follows the path D, and at each waypoint it sends a message beacon. The non-localized nodes that can hear it reply with an ack message to the drone. From the round-trip time, the drone can infer the slant distance between itself and the node, and from that, it computes the ground distance. The drone locally saves all the ground distance measurements for the nodes not yet localized.

For each node P, the drone first finds a rough estimate (without any guaranteed bound on the position error), called $\hat{P}$, of the position of the node P. Then, the drone locates the logical diamond $\hat{\Delta}$ where $\hat{P}$ resides, and uses the three ranging waypoints associated to $\hat{\Delta}$ to localize P with guaranteed bound precision ϵ_L.

Precisely, to compute $\hat{P}$, the drone performs a trilateration with the first three distance measurements it collects that satisfy the $d_{\min}$ constraint and that belong to two different vertical scans. From $\hat{P}$, it locates the closest vertical scan $S^{\sigma(\hat{P})}$, with index $\sigma(\hat{P}) = \left\lfloor \frac{x_{\hat{P}}+F_x}{H} \right\rfloor + 1$, on the left of $\hat{P}$. Then, based on the distance between $S^{\sigma(\hat{P})}$ and $\hat{P}$ the drone computes $w_1(\hat{P}), w_2(\hat{P}), w_3(\hat{P})$. Let:

$$\bar{y_1} = \left\lfloor \frac{y_{\hat{P}}}{2I_w} - \left\lceil \frac{\sqrt{3}(x_{\hat{P}} - x_{S^{\sigma(\hat{P})}})}{2I_w} \right\rceil \right\rfloor 2I_w$$
$$\bar{y_3} = \left\lfloor \frac{y_{\hat{P}}}{2I_w} + \left\lceil \frac{\sqrt{3}(x_{\hat{P}} - x_{S^{\sigma(\hat{P})}})}{2I_w} \right\rceil \right\rfloor 2I_w$$
$$\bar{y_2} = \frac{(\bar{y_1} + \bar{y_3})}{2}$$

Then, if $x_{S^{\sigma(\hat{P})}} \leq x_{\hat{P}} \leq x_{S^{\sigma(\hat{P})}} + \frac{d_{\min}}{2}$, the drone will again trilaterate P after the distance measurements at the three ranging points:

$$w_1(\hat{P}) = \left(x_{S^{\sigma(\hat{P})-1}}, \bar{y}_1\right) \tag{4.5}$$
$$w_2(\hat{P}) = \left(x_{S^{\sigma(\hat{P})-2}}, \bar{y}_2\right)$$
$$w_3(\hat{P}) = \left(x_{S^{\sigma(\hat{P})-1}}, \bar{y}_3\right)$$

Similarly, if $x_{S^{\sigma(\hat{P})}} + \frac{d_{\min}}{2} < x_{\hat{P}} \leq x_{S^{\sigma(\hat{P})+1}}$, the localization will be completed as soon as the drone has collected the distance measurements from the waypoints:

$$w_1(\hat{P}) = \left(x_{S^{\sigma(\hat{P})}}, \bar{y}_1\right) \tag{4.6}$$
$$w_2(\hat{P}) = \left(x_{S^{\sigma(\hat{P})-1}}, \bar{y}_2\right)$$
$$w_3(\hat{P}) = \left(x_{S^{\sigma(\hat{P})}}, \bar{y}_3\right)$$

In both cases, the localization of P will now satisfy the bound ϵ_L since the three waypoints satisfy the $d_{\min}$, $d_{\max}$, and $\beta_{\min}$ constraints.

4.4 Experimental Evaluation

We have implemented the OMNI localization algorithm in MATLAB. We simulate a localization mission by deploying at random n nodes, with $n = 50, 100, 250$ and 500, on a 500×500 m^2 map (at most one sensor in every square of size 25 m). In all experiments, the drone travels along the path D and takes distance measurements at the waypoints. We assume the slant precision to be $\epsilon_s = 0.1m$, as claimed by DecaWave for their IR-UWB transceivers [92].

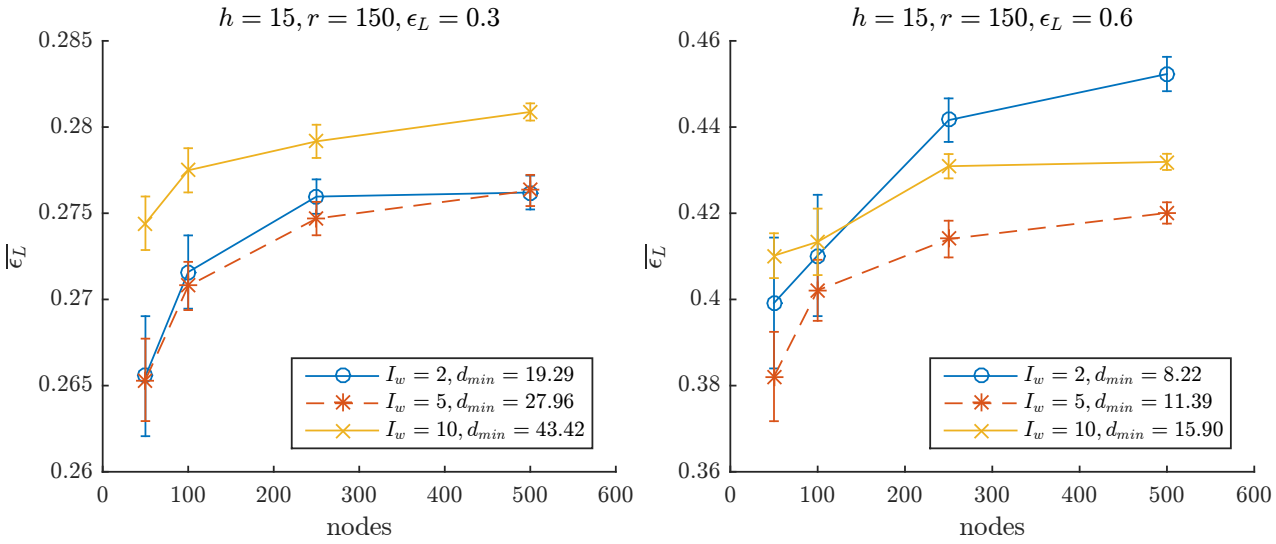
		ϵ_L(m)		
h(m)	I_w(m)	0.3	0.6	0.9
	2	19.29	8.22	5.83
15	5	27.96	11.39	8.16
	10	43.42	15.90	11.22
	2	32.82	13.86	9.63
30	5	41.44	17.61	12.55
	10	55.93	22.78	16.32

Table 4.1: The values $d_{\min}$ for OMNI.

To simulate the slant distance between the drone in position w at altitude h and the node P, we compute the exact slant distance (without error) $s = \sqrt{|wP|^2 + h^2}$, where $|wP|$ is the Euclidean

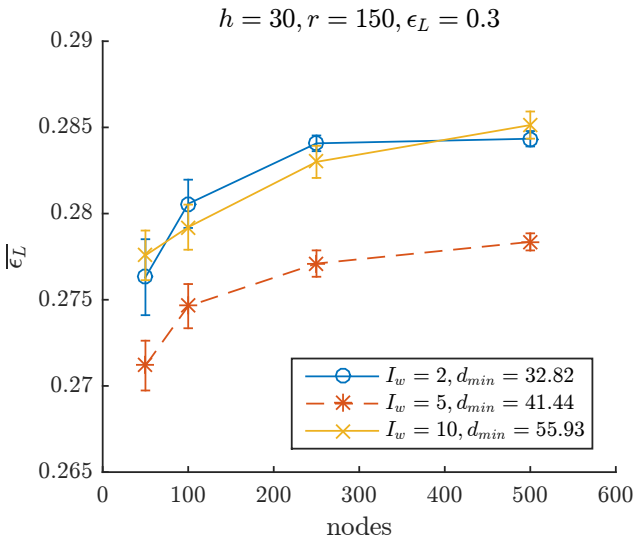
(exact) distance between the projection of w on the ground and the (exact) position of P. Then we generate a slant error $e_s \in [-\epsilon_s, \epsilon_s]$ and we say that the slant distance measured by the drone is $s(w,P) = (s + e_s)$. The ground distance (which includes the ground error) is computed as $d(w,P) = \sqrt{s(w,P)^2 - h^2}$. Given three ground measures, the estimate position of P is computed as the least-squares-error solution of their trilateration.

For $\epsilon_L = \{0.3, 0.6, 0.9\}$, we derive and reported in Table 4.1, the values of $d_{\min}$ for OMNI. If such a value $d_{\min}$ does not satisfy Eq. (4.2), our localization technique cannot achieve ϵ_L, unless we decrease the value of I_w. Fixing n, for each mission, we simulate the localization technique.

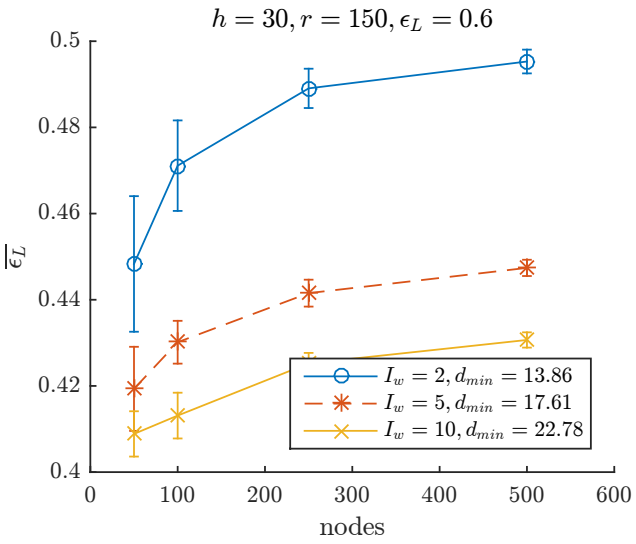


(a) $h = 15$ and $\epsilon_L = 0.3$.

(b) $h = 15$ and $\epsilon_L = 0.6$.



(c) $h = 30$ and $\epsilon_L = 0.3$.



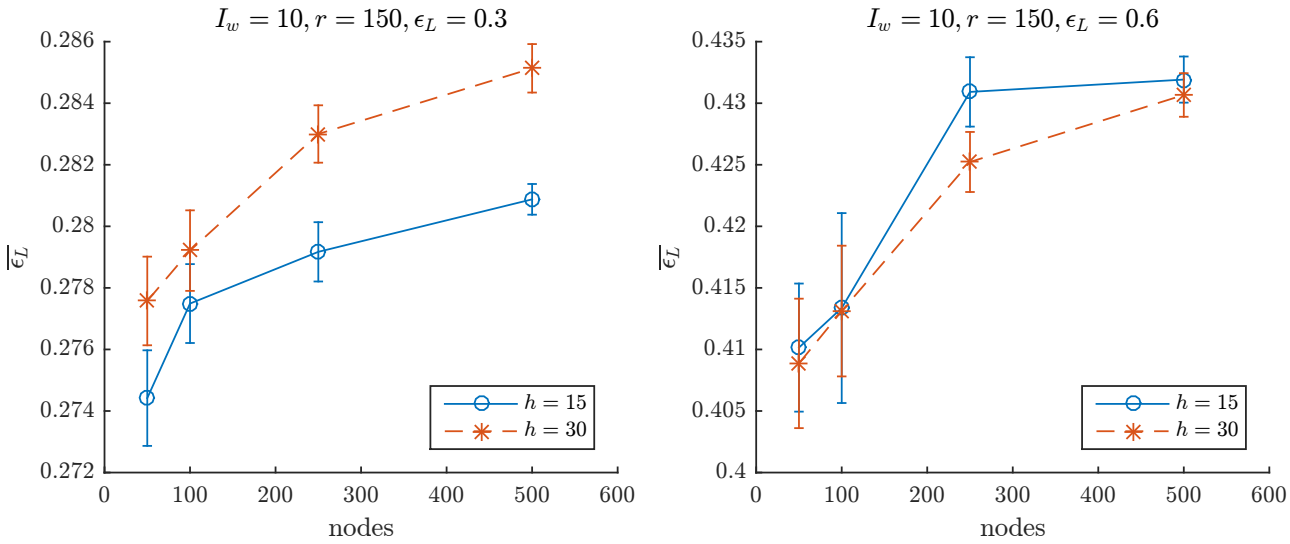
(d) $h = 30$ and $\epsilon_L = 0.6$.

Figure 4.6: The precision $\bar{\epsilon}_L$ when $r = 150\,\text{m}$.

As shown in Fig. 4.6, the desired localization precision is always greater than the average worst experimental localization precision. Not only, $\bar{\epsilon_L} \leq \epsilon_L$, but also $\epsilon_L(M) \leq \epsilon_L$ for each individual mission. The distance between the desired value ϵ_L and the experimental $\bar{\epsilon}_L$ increases when $d_{\min}$ increases and I_w increases. So the experiments guarantee a better (i.e., smaller) localization precision when $d_{\min}$ is larger.

When $h = 30\,\text{m}$ and $\epsilon_L = 0.6\,\text{m}$, $d_{\min}$ is greater than when $h = 15\,\text{m}$ and $\epsilon_L = 0.6\,\text{m}$. In other words, a larger $d_{\min}$ is required to achieve the same ϵ_L at higher altitudes (see Fig. 4.6b and 4.6c). This shows that the experimental localization precision is sensitive to the altitude.

Figs. 4.7a and 4.7b plot the precision $\bar{\epsilon}_L$ for $r = 150\,\text{m}$ at different altitudes when the a-priori localization precision to be guaranteed is $\epsilon_L = 0.6$ and $\epsilon_L = 0.3$, respectively. Although the distance between the theoretical and the experimental precisions decreases when h increases, the experimental localization precision is always smaller than ϵ_L.



(a) $\overline{\epsilon_L} = 0.3$ and h varies.

(b) $\overline{\epsilon_L} = 0.6$ and h varies.

Figure 4.7: The precision $\bar{\epsilon}_L$ when $r = 150\,\text{m}$ for different altitudes.

Conclusion In this work, we presented a localization technique that replaces multiple anchors with a flying drone. Our solution guarantees a provable bound on the position error by tuning a single parameter: the minimum ground distance measurement. During the localization mission, the drone first computes for each sensor a rough estimation of its position, and then it carefully selects the waypoints so as the required localization precision is guaranteed.

Chapter 5

Precise and Secure Localization with Omnidirectional Antennas[1]

5.1 Introduction

The dependability of many distributed systems relies on knowing the position of the component devices. Some examples are sensor networks for environmental monitoring (both for civilian or military purposes), geographic routing, autonomous vehicle coordination, and so on. In all these cases, if the system believes that a device is in a position different from the real one, then it could infer wrong information and possibly take wrong decisions.

Figure 5.1: How to provide security?

Periodically measuring the position of the devices is not enough to guarantee security. Indeed, the majority of the positioning methods are vulnerable to attacks in which an adversary falsifies the position measurement [103].

Providing secure measurement of positions has shown to be a non-trivial problem [104–108]. A promising approach is *verifiable multilateration* [107], which is a provably secure technique to

[1]Part of this work has been published to 2017 IEEE Transactions on Mobile Computing

determine a position by measuring the distances from (at least) three anchors by means of *distance bounding protocols* [109]. A distance bounding protocol is a cryptographic protocol able to measure a secure upper bound to the distance between two devices. Verifiable multilateration is an extremely versatile technique, since it can withstand both external adversaries and compromised devices.

5.2 Related Work

Secure positioning aims at measuring the position of a device in the presence of an adversary (intended as an anchor node or a sensor) that wants to falsify such a measurement. Researchers proposed many methods [103], which offer different levels of security (provable or only statistical), and defend against different kinds of adversary (external or internal). The authors in [107] proposed a provably secure positioning method called *verifiable multilateration*. In this proposal, the system measures the distances from a set of trusted anchors by means of distance bounding protocols. The position is computed by trilateration, and it is considered secure if it lies inside the convex hull of the anchors. A previous work [110] improved verifiable multilateration in such a way as to require considerably less anchors to cover the same area. This is achieved by leveraging the enlargement attack resistance of wireless distance bounding protocols, analyzed by [111] and [112]. For this purpose, we replace the fixed anchors with a single mobile drone, thus completely eliminating the need for an expensive anchor infrastructure.

In [105] it was shown that the security of verifiable multilateration in the presence of non-ideal distance bounding protocols is vulnerable to some extent to PHY-level attacks. The research in [106] proposed a secure location verification mechanism based on mobile stations (not necessarily drones). In their system, the movements of the mobile stations are random, and the security is based on the assumption that the adversary cannot observe nor predict such movements in any way. This assumption could be too strong for some applications. Instead, our approach does not require the unobservability of the drone movements. Moreover, a random path can take a very long time to cover a given area, thus it could be unsuitable for a limited-battery drone. On the other hand, by using verifiable multilateration we can plan shorter, non-redundant and deterministic paths.

A problem related to ours is drone-based (insecure) localization of ground devices [35–37, 113–115]. All these works do not have security in mind, and their position measurements cannot be considered trusted in a hostile environment or in the presence of compromised devices. All the protocols that assume that the devices collect information and infer their position averaging such

information are not secure since an adversary could simply send fake position broadcasts, in such a way to confuse the devices. Authenticating the position broadcasts does not solve the issue, since an adversary could listen to a legitimate broadcast and replay it on different positions. Also all the methods based on the received signal strength are poorly secure, since an adversary has an easy play on falsifying this information. In this research, we assure the trustworthiness of the measured positions by using verifiable multilateration [107], which is a provably secure method.

5.3 Preliminaries

A *distance bounding protocol* [109] is a cryptographic protocol able to measure a distance between two devices, in such a way that an adversary cannot falsify the measurement to be shorter than the real distance (*reduction attack*). In this case we refer to an adversary as a sensor node inside the area. A distance bounding protocol determines a distance by precisely measuring the round-trip time between a challenge and a response message. The messages convey numeric quantities which are unpredictable by an external adversary, so that she cannot reduce the round-trip time by guessing and transmitting in advance the messages. Moreover, the correct response is unpredictable by the responding device itself before having received the challenge. In this way, a compromised device cannot reduce the round-trip time by responding in advance.

Verifiable multilateration [107] is a provably secure technique to determine a position, which leverages distance bounding. In verifiable multilateration, the position of a *node* is determined by measuring the distances between the node and at least three *anchors* (A_1, A_2 and A_3) whose positions are known (Fig. 5.2) where the dashed triangle represents the verifiable triangle.

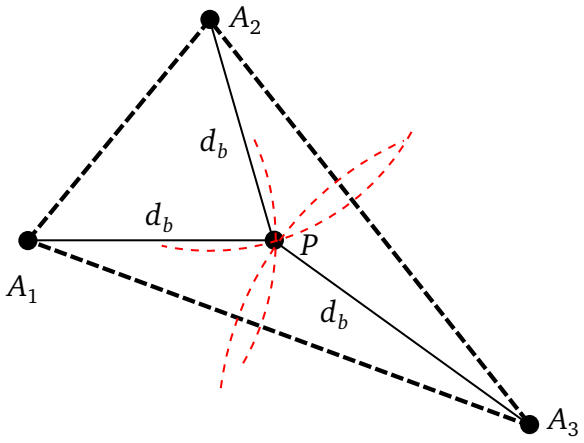


Figure 5.2: Verifiable multilateration.

The distance measurements are performed by means of distance bounding (d_b) protocols.

The node's position is computed by trilateration, and is accepted only if it lies inside the triangle formed by the anchors (*verifiable triangle*). Otherwise, it is discarded as untrusted. Indeed, if an adversary wants to falsify a position measurement inside the verifiable triangle, then she must perform a reduction attack against at least one distance bounding protocol, which is infeasible. Note that the coverage of verifiable multilateration is only the verifiable triangle, because the outside positions are discarded. In the case in which more than three anchors are involved in the localization process, the position is accepted only if it lies in at least one of the verifiable triangles formed by a triplet of anchors. Verifiable multilateration can defend against different threat models: an external adversary or a compromised node.

5.4 Drone-Based Verifiable Multilateration

Verifiable multilateration is able to securely measure the position of a node using a drone. The drone follows a *path*, touching a sequence of *waypoints*.

5.4.1 System Model

We suppose that the drone shares a different secret k with each node, by which the distance bounding protocols are executed. To distribute in a secure manner the secrets to the nodes, a generic key deployment method (e.g., [116]) can be used. In our system, all the nodes are on the ground, while the drone flies at a non-negligible *altitude* (h). Accordingly, we consider $\gamma_h = 0\,\text{m}$ and also $\gamma_d = 0\,\text{m}$. A *path* is a sequence of *waypoints* $W_1, \ldots, W_m$, each of which is a point on the Cartesian plane at height h from the ground. The drone visits the waypoints in the order specified by the sequence. We require that the path is *closed*, in the sense that the drone goes again to the first waypoint at the end.

We distinguish two types of mission: the *localization mission* and the *verification mission*. In a localization mission, we do not have any prior knowledge of the position of the nodes. We only know that the nodes lie somewhere inside a given *deployment area* Q. Moreover, we suppose the deployment area to be rectangular, of sides $Q_x \times Q_y$.

On the other hand, in a verification mission we assume to have prior knowledge of the position of the nodes, from which we compute the drone's path. We call them *prior positions* (P'_i). The prior positions are not trusted, therefore we want to securely verify them by means of verifiable multilateration. If the positions determined by verifiable multilateration are not consistent with the prior ones, an attack is detected. The prior positions could be imprecise, and thus different from the *actual positions* (P_i). We call the distance between the prior and the actual position the

prior position error. We assume that the prior position errors are bounded, and we call such a bound the *prior position precision* (ϵ_P):

5.4.2 General Requirements Of The Path

For both localization and verification missions, the path must respect the following requirements.

- **Communication range requirement.** Each waypoint must measure nodes within the communication range. Farther nodes could be impossible to reach with a distance bounding protocol.
- **Verifiable triangle requirement.** Each node has to be measured from three or more distinct waypoints, and at least one verifiable triangle formed by them must contain the node. This is required for the localization to be secure.
- **Path length requirement.** The path length should be as short as possible. This is preferable for saving time and drone's battery life.

In the following, we will present two path planning algorithms that fulfill such requirements: LOCALIZERBEE and PRECISEVERIFIERBEE. LOCALIZERBEE produces paths for localization missions, while PRECISEVERIFIERBEE for verification missions. PRECISEVERIFIERBEE is an extension of VERIFIERBEE [117] which offers a provable bound on the positioning error, at a cost of a longer path.

5.5 The LOCALIZERBEE algorithm

In a localization mission, we do not have any prior knowledge of the position of the nodes. We only know that the nodes lie somewhere inside the deployment area. To respect the verifiable triangle requirement, given a set of planned waypoints forming a set of verifiable triangles, we have to be sure that each node is contained in at least one of them. We can reach this by imposing that the verifiable triangles are contiguous, and that their union includes the whole deployment area. In this way, we do not leave any part of the deployment area uncovered.

We also have to fulfill the communication range requirement, so we have to impose that the node is reachable by all the waypoints forming the verifiable triangle. To do this, it is sufficient that the sides of the verifiable triangles are smaller than or equal to a *maximal triangle side* (L_{max}), defined as:

$$L_{max} \leq \sqrt{r^2 - h^2}$$

We present LocalizerBee, an algorithm which solves the problem formalized above. LocalizerBee uses a Traveller Salesman Problem (TSP) solver algorithm as a building block.

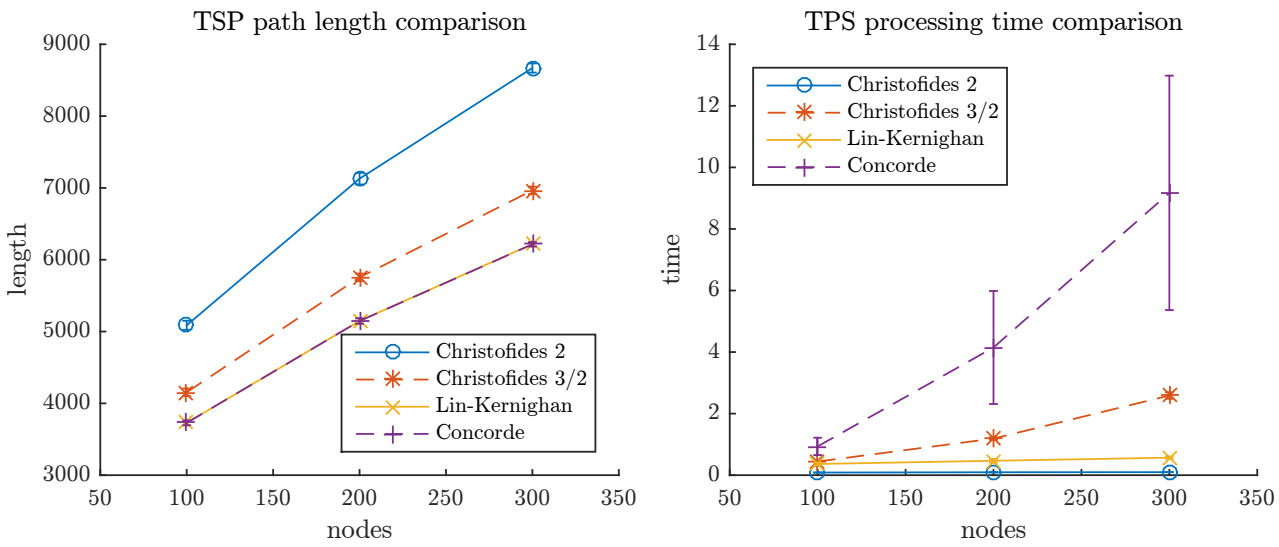
The generic version of TSP is an NP Hard problem in combinatorial optimization. Unless P=NP, there does not exist any polynomial-time approximation scheme for this generic problem [118]. In a special case of TSP in which the triangle inequality holds, it is possible to obtain a polynomial-time approximation scheme. Because the Euclidean distances satisfy the triangle inequality, we can use the Christofides algorithm [119]. There exist two different versions of Christofides algorithm. The first one is an approximation algorithm that guarantees that its solutions will be within a factor of 2 of the optimal solution length, and it has a time complexity of $O(n^2 \log n)$ where n represents the nodes number. The second one, i.e. the improved version, has an approximation factor $\frac{3}{2}$, but requires a time complexity of $O(n^3)$ [120]. The Concorde solver is the best-performing exact TSP solver known. The time complexity is of the form $O(a \cdot b^{\sqrt{n}})$, where $a \approx 0.21$ and $b \approx 1.24$ [121]. The Chained Lin-Kernighan heuristic, implemented by the University of Waterloo, is considered state-of-the-art and provides a trade-off between efficiency and optimality [122].

In Fig. 5.3 we compared the performance of the four TSP algorithms. We generate 35 different random nodes deployments on a 500×500 m^2 map. For each scenario, we simulate the four TSP algorithms. About the path length, as depicted in Fig. 5.3a, the Chained Lin-Kernighan heuristic matches exactly the exact Concorde solver for $100, 200$ and 300 nodes for each scenario. This means that Lin-Kernighan is the best solution considering the lower time complexity. Moreover, as predicted, the bounds of Christofides' algorithms work as well. Concerning the CPU processing time, as shown in Fig. 5.3b, Christofides 2 is the fastest algorithm.

For our purposes, the TSP solver is not required to be optimal, but rather to find an approximate shortest path that visits a set of points, and then returns to the first point (closed path). Of course the performances of the TSP solver will affect those of LocalizerBee both in terms of path length and processing time.

LocalizerBee operates in two phases: (i) waypoint grid construction; (ii) waypoint ordering. In the waypoint grid construction phase, it builds a set of waypoints forming a grid of isosceles triangles which covers the whole deployment area. The triangles have the same base T_x and the same height T_y, and they are placed as shown in Fig. 5.4.

Note that we added two halved triangles on the left and the right sides of each triangle row, in such a way that the union of the verifiable triangles is a rectangle. We use a grid of isosceles triangles and not, for example, regular triangles (see [38]). Indeed, isosceles triangles are more flexible, because they can be "resized" along both dimensions. This allows us to cover the whole



(a) Path length (m).

(b) Processing time (s).

Figure 5.3: TSP comparisons.

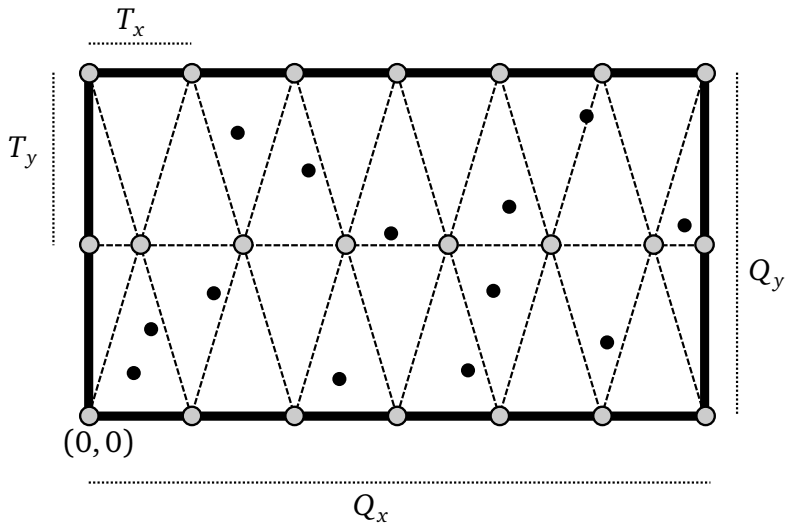


Figure 5.4: Waypoint grid construction phase of LocalizerBee.

deployment area as tightly as possible, without wasting path length for covering external areas uselessly. The base and the height of the triangles must be as long as possible, without violating the maximal triangle side. To this aim, given a base T_x, the longest feasible height $T_{y_{max}}$ is:

$$T_{y_{max}} = \sqrt{L_{max}^2 - (T_x/2)^2}$$

LOCALIZERBEE computes the base and the height of the triangles as follows.

$$T_x = (Q_x) / \left\lceil \frac{Q_x}{L_{max}} \right\rceil$$
$$T_y = (Q_y) / \left\lceil \frac{Q_y}{T_{y_{max}}} \right\rceil$$

In this way, there are $\left\lceil \frac{Q_y}{T_{y_{max}}} \right\rceil$ triangle rows, each containing $2 \cdot \left\lceil \frac{Q_x}{L_{max}} \right\rceil - 1$ unhalved triangles plus 2 halved ones.

In the waypoint ordering phase, LOCALIZERBEE executes the TSP solver to connect all the waypoints. Fig. 5.5 shows an example of LOCALIZERBEE path for a deployment area $Q = 500\,\text{m} \times 500\,\text{m}$, with $r = 100\,\text{m}$ and $h = 30\,\text{m}$. We fixed the home waypoint to be in the south-west corner of the deployment area. As TSP solver, we employed the Chained Lin-Kernighan heuristic.

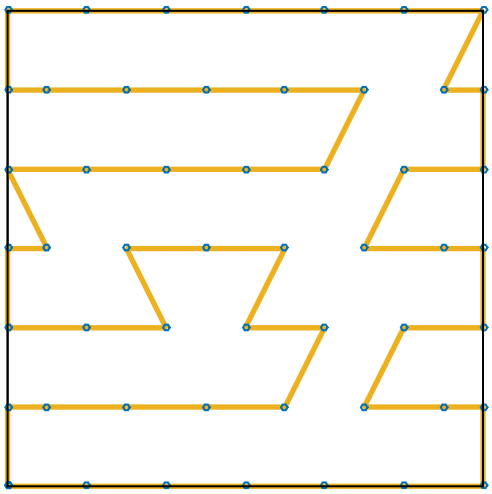


Figure 5.5: Example of LOCALIZERBEE path.

As a future work, it would be interesting to study solutions ad hoc for this special case of TSP.

5.6 The VERIFIERBEE algorithm

In a verification mission, we have prior knowledge of the position of the nodes. By leveraging this information, we can design a path planning algorithm which produces shorter paths than LOCALIZERBEE. The prior positions are affected by an error bounded by ϵ_P. To be sure that a verifiable triangle contains the i-th node, the verifiable triangle must include the whole circle centered in P_i and with radius ϵ_P. We call such a radius the *tolerance radius* (r_{tol}):

$$r_{\text{tol}} = \epsilon_P$$

We also have to fulfill the communication range requirement, so we have to impose that the node is reachable by all the waypoints forming the verifiable triangle that contains it. To do this even in the presence of control errors, it is sufficient that the ground distances between the planned waypoints and the prior position are less than or equal to a *maximal ground distance* (d_{max}), defined as:

$$d_{\text{max}} \leq \sqrt{r^2 - h^2} - r_{\text{tol}} \tag{5.1}$$

VERIFIERBEE uses a TSP solver as a building block to find a first feasible solution. Then, such a solution is iteratively improved, following a local search heuristic.

VERIFIERBEE operates in three phases: (i) initial path construction; (ii) iterative improvement; (iii) waypoint reordering.

In the initial path construction phase, it builds a set of waypoints: the home waypoint plus three waypoints for each node, placed at fixed positions to form a *minimal verifiable triangle*. The minimal verifiable triangle is a regular triangle centered on P_i and with the vertices at $2r_{\text{tol}}$ from the center. By geometry, this is the smallest distance with which the regular verifiable triangle respects the r_{tol} constraint. After having built the set of waypoints, we run the TSP solver on them to find an approximate optimal path that touches them all. The initial path is thus complete, and it is formed by $3n + 1$ waypoints, where n is the number of nodes.

After having built the initial path, VERIFIERBEE changes it iteratively, following a local search heuristic. At each step, VERIFIERBEE analyzes the possible changes and applies the most convenient one, that is the one that decreases the total path length the most. The iterative improvement phase terminates when no change is possible or convenient anymore, meaning that we found a local minimum. The changes are of two kinds: *waypoint moving* and *waypoint pruning*.

The iterative improvement phase may change the position and may decrease the number of the waypoints, but it does not change their order, which remains the same as the initial path. As a consequence, sometimes it is convenient to reorder the waypoints by running the TSP solver again.

This is done in the waypoint reordering phase. The iterative improvement and the waypoint reordering phases are repeated, until the path length stops decreasing.

Algorithm 1 shows a pseudo-code description of VERIFIERBEE.

Algorithm 1 VERIFIERBEE

P_i, W_1, r_{tol}, $d_{\max}$

1: *Path* ← a set of waypoints.
2: *Path* ← SOLVETSP(*Path*) ▷ initial path
3: **while** true **do**
4: *Path* ← ITERATIVELYIMPROVE(*Path*, $r_{\text{tol}}, d_{\max}$)
5: **if** *Path* has not been improved **then**
6: **break**
7: **end if**
8: *Path* ← SOLVETSP(*Path*)
9: **if** *Path* has not been improved **then**
10: **break**
11: **end if**
12: **end while**

5.7 The PRECISEVERIFIERBEE algorithm

VERIFIERBEE builds a path able to securely verify a set of nodes. However, it does not take into consideration the positioning precision at all.

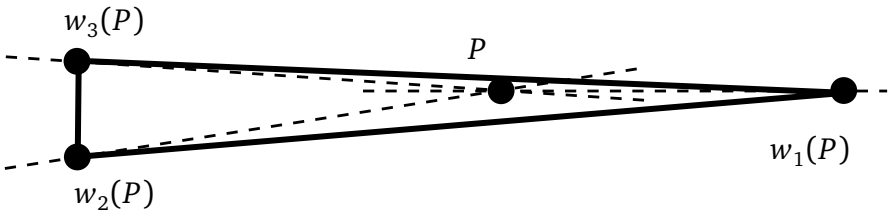


Figure 5.6: Bad layout.

In the process of finding an approximate optimal path, VERIFIERBEE could place the waypoints in such a way to form "bad" geometry layouts, for example three waypoints collinear with the measured node (see Fig. 5.6 and Fig 3.8b). This in turn can cause a big error in the position estimation. As we said in Chapter 3, if a waypoint is in plumb-line above the measured node, then the computed ground distance will be affected by a big error.

In VERIFIERBEE, the presence of the r_{tol} constraint avoids the possibility of waypoint-only collinearity. On the other hand, the possibility of measuring from above or of having node-waypoint collinearity are avoided only if r_{tol} is sufficiently large. The greater r_{tol} is, the more they are mitigated. However, purpose of the the r_{tol} constraint is to fulfill the verifiable triangle requirement, and should not be used for other purposes. To avoid bad geometry layouts, it is preferable to introduce specialized constraints. We thus impose two additional constraints to the path: the *minimal ground distance* d_{min} and the *minimal angular aperture* β_{min} (see Chapter 1, Section 2.5 Terminology).

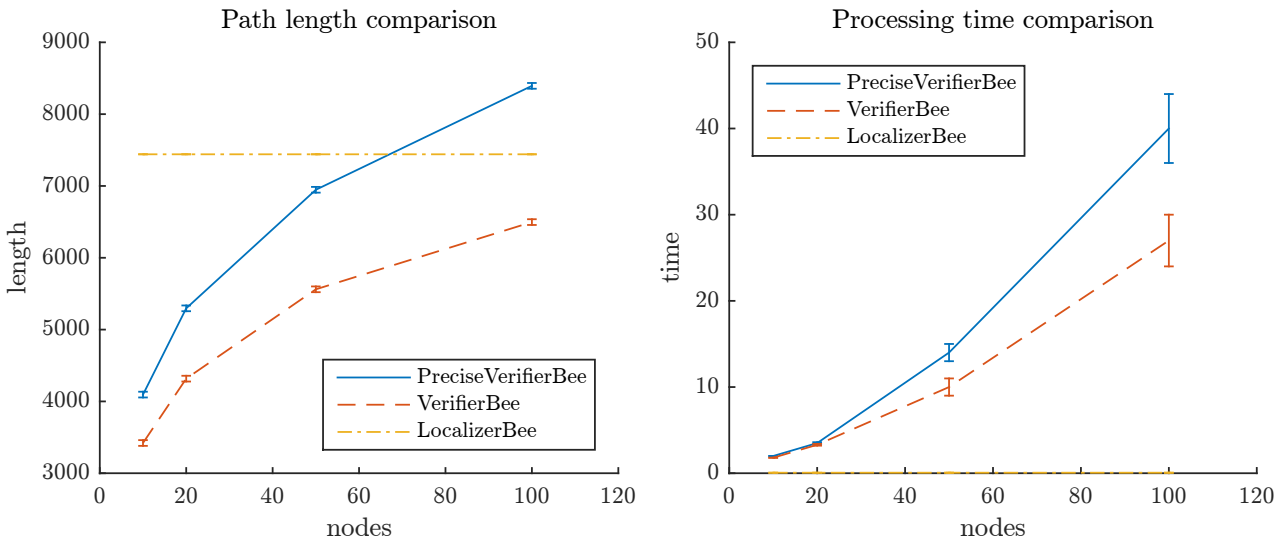
PRECISEVERIFIERBEE is an extension of VERIFIERBEE, which respects the additional d_{min} and α_{min} constraints. In this way, it guarantees a bound on the positioning error, at a cost of a longer path. The general structure of the VERIFIERBEE algorithm (Algorithm 1) [117] remains the same, except that the initial path construction and the iterative improvement phases have to take into account the two additional constraints.

In the initial path construction phase, the vertices of the minimal verifiable triangle must be at $\max\{2r_{\text{tol}}, d_{\text{min}}\}$ from the center. By geometry, this is the smallest distance with which the regular verifiable triangle respects both the r_{tol} and the d_{min} constraints. Note that, since the minimal verifiable triangle is a *regular* triangle, it enjoys the best possible angular aperture ($\beta_{\text{min}} = 60°$), and thus the β_{min} constraint is always fulfilled. In the iterative improvement phase, the freedom space of the waypoints has to be narrowed to take into account the additional constraints.

5.8 Experimental Evaluation

We implemented LOCALIZERBEE, VERIFIERBEE and PRECISEVERIFIERBEE with the MATLAB language and we tested their path lengths. We run VERIFIERBEE and PRECISEVERIFIERBEE on a number of prior positions randomly distributed on a deployment area $Q = 1000 \times 1000\,\text{m}^2$. For PRECISEVERIFIERBEE, we fixed a positioning precision of $\epsilon_L = 5\epsilon_s$. We repeated such a simulation for 35 different random sets of prior positions. We report our results in Fig. 5.7.

Fig. 5.7a shows the average length of the planned paths with respect to the number of nodes, compared with the length of the path produced by LOCALIZERBEE with the same deployment area. Note that the length of the paths produced by VERIFIERBEE and PRECISEVERIFIERBEE grow logarithmically with the number of nodes. VERIFIERBEE produces always the shortest paths, because it enjoys the prior knowledge of the positions and at the same time it does not have to fulfill any precision guarantee. As the number of nodes grows, PRECISEVERIFIERBEE becomes significantly worse than VERIFIERBEE, and with 100 nodes it even becomes worse than LOCALIZERBEE.



(a) Average Path length (m).

(b) Average Processing time (s).

Figure 5.7: LocalizerBee, VerifierBee and PreciseVerifierBee comparisons.

This means that guaranteeing a positioning precision is quite an expensive requirement when the nodes are many. With 100 nodes or more, it could be convenient to relax the precision guarantee (i.e., $\epsilon_L > 5.0\epsilon_s$). The paths produced by the proposed algorithms are feasible by commercial drones with a single battery charge. For example, the Phantom 4 Professional drone is capable of running about 13 km in practical tests[2]. Our path planning algorithms produce shorter paths (max 8.4 km with PreciseVerifierBee and 100 nodes). This allows us to use cheaper drones or to perform additional tasks in the same flight.

Fig. 5.7b shows the average processing time. The processing time of LocalizerBee is negligible with respect to the other two algorithms. The slowest algorithm (PreciseVerifierBee) takes roughly 40 seconds to run with 100 nodes. This should be fully acceptable for an off-line computation. VerifierBee takes longer to run with respect to PreciseVerifierBee. This is because both algorithms follow a local search heuristic, but VerifierBee has to respect less constraints, and thus it can do more improvement steps before stopping on a local minimum. On the other hand, PreciseVerifierBee finds a local minimum earlier, and then stops.

Conclusion In this work, we explored the approach of using drones to securely localize a set of devices by means of verifiable multilateration. We proposed three path planning algorithms for secure positioning and secure position verification: LocalizerBee, VerifierBee and PreciseVerifierBee.

[2] http://myfirstdrone.com/phantom-4/dji-phantom-4-real-world-range-test/

Chapter 6

Introduction to the Directional Antennas[1]

6.1 Introduction

Directional antennas are used in several applications including satellite communications, terrestrial microwave communications, VHF and UHF terrestrial TV transmission, cellular communication, and rural mesh networks [123].

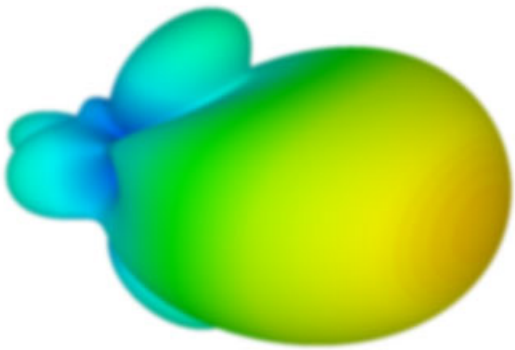


Figure 6.1: A classical directional pattern.

Antenna directionality focuses transmission power in a particular direction and improves communication range while simultaneously reducing interference with nearby antennas. However, a drawback is that a priori knowledge of the location of the intended radio receiver and, in some cases, the ability to steer the antenna or switch beams in the relevant direction is required to form connections between pairs of nodes. A larger issue is of network-wide connectivity which becomes especially important in applications such as disaster management and military battlefield

[1]Part of this work has been published to ALGOSENSORS 2015 - 11th International Symposium on Algorithms and Experiments for Wireless Sensor Networks

communications that require the rapid setup of a wireless multihop (mesh) network [124, 125]. In order to benefit from the ability of directional antennas to focus power in such settings it is critical to understand how a connected network can be formed using highly directional antenna beams.

While there has been a lot of discussion on the issue of capacity in highly directional antenna-based networks, there are few works in the literature that have addressed the question of connectivity. Notable among them is [126] which addresses this question, but under an idealized model of directional transmission that assumes a sector shaped area of transmission (with some back lobes) with uniform power transmitted throughout the sector. In this work we approach directional transmission in much greater generality and provide a result that holds for a large family of gain functions. We demonstrate how these results help us find the optimal power for connectivity in the dense mesh network setting. Using the gain function as our primary mode of describing the directionality of an antenna, we present a theorem that helps us determine the optimal power for connectivity for all gain functions that satisfy some moderate conditions.

Another important novelty is that our main result on connectivity is completely rigorous. A disadvantage seen in [126] is that they assume that if all the antennas are positioned randomly and their antennas are oriented randomly then the edges between nodes are formed independently. This assumption clearly does not hold, as we will show in detail in Section 6.3. Our mathematical results do not need the independence assumption. Hence we claim to present the first fully rigorous analysis of connectivity in dense mesh networks built with directional antennas.

We assume that nodes are deployed randomly in a finite circular area, and that each node is equipped with a directional antenna whose orientation is initially fixed randomly and kept fixed thereafter.

6.2 Related work

Connectivity in mesh networks using omnidirectional antennas has been studied in depth since the seminal work in [127]. They proved that for m nodes with omnidirectional antennas randomly placed in a disc of unit area, if transmission power for all nodes was set such that each node could communicate with any other node in a circular vicinity of area $(\log m + c(m))/m$, then the network is asymptotically connected with probability 1 if and only if $c(m) \to \infty$ [127]. Our connectivity result, Theorem 5 is an analog of this result for the more complex setting where the antennas are highly directional. Our work on connectivity benefits from the general theorem proved in [128].

Connectivity was widely studied within the omnidirectional model in mobile ad hoc net-

works [129], in thin finite strips [130], under a physical model for interferences [131], and when nodes are active independently with a certain probability [132, 133].

Several authors have studied connectivity of mesh networks equipped with steerable directional antennas in contrast to our work which considers non-steerable antennas. In [134] sensors deployed on a unit line and unit square with steerable directional antennas are considered. Given a set of nodes on a plane, each with a directional antenna, modeled as a sector, the authors in [135] investigated the problem of orienting the antennas to get a connected network.

Another work models the communication area of a steerable directional antenna as a wedge of infinite area which captures its directionality [136] and shows that a sixty degree directional antenna suffices to form a connected network for arbitrarily located nodes. In [137] the problem of connectivity through simulations when each node is equipped with several different directional antennas oriented uniformly in a circular fashion has been studied. In [138] the problem of placement of wireless sensor nodes, with a view to ensuring connectivity and coverage is considered. In our work node placement is random. In [139] a scenario of nodes deployed over a finite area and equipped with linear and circular antenna arrays used for random beamforming is considered, demonstrating that increasing directionality (i.e., setting a narrow beamwidth) leads to larger connected components. Our theoretical results support their experimental findings on connectivity.

6.3 Modeling directional mesh networks

The power received by a receiving antenna, P_{R_x}, at distance r from a transmitting antenna that is transmitting at wavelength λ with power P_{T_x} is described by the Friis transmission equation:

$$P_{R_x} = P_{T_x} G_{R_x} G_{T_x} \left(\frac{\lambda}{4\pi r} \right)^2, \tag{6.1}$$

where G_{R_x} and G_{T_x} are the receiver's and transmitter's gains and depend on the orientations of the two antennas. For highly directional antennas these gains can be very high since these antennas tend to concentrate their beams in one direction. Gain is formally defined as the ratio of the power radiated in a given direction per unit solid angle to the average power radiated per unit solid angle (see [140, 141]).

Although the gain function depends on both the polar and azimuthal angles in 3 dimensions we will assume for ease of presentation that the gain function $G : [-\pi, \pi] \to \mathbb{R}_+ \cup \{0\}$ is defined over two dimensions, i.e., depends only on the azimuthal angle. We note that our methods are general and can be transposed to 3 dimensions with suitable modifications. We assume that our

gain function has the following properties:

- Directionality: $G(\psi) = 0$ for $|\psi| \geq \pi/2$.
- Symmetry around angle of orientation: $G(\psi) = G(-\psi)$.
- Monotonicity: $G(\psi) > G(\psi')$ whenever $|\psi| < |\psi'|$.

The assumption that G takes non-zero values only in $[-\pi/2, \pi/2]$ neglects back-lobe transmission, which is a simplification we make for ease of presentation. From these properties we can additionally deduce that $G(\cdot)$ reaches its maximum value at 0. Also $G(\cdot)$ is not an invertible function, since it is not one-to-one. So we follow the convention, similar to that of inverse trigonometric functions, that $G^{-1}(x)$ is a positive valued function, i.e., if $G(\psi) = x$ then we say that $G^{-1}(x) = |\psi|$. Also, by the reciprocity principle it is known that the receiver gain and transmitter gain of an antenna are identical. In this work we will deal with settings where all antennas are considered identical to each other and so we will consider only one single gain function at a time.

6.3.1 A realistic directional fading model

In a realistic antenna setting, the gain decreases as we move away from the angle of orientation of the antenna. In this work we will work with a family of gain functions that satisfy this property. We will refer to this model as the *directional fading model* or simply the *fading model*. This family of functions, which has been mentioned in the antenna theory literature as being of particular interest [140, 141], is:

$$G_f^n(\psi) = \begin{cases} G_f^n(0)\cos^n(\psi) & 0 \leq |\psi| \leq \frac{\pi}{2} \\ 0 & |\psi| \geq \frac{\pi}{2} \end{cases} \tag{6.2}$$

where n takes even values, and the f in the subscript of G_f is to indicate the "fading" model and differentiate it from the ideal model we will also study (see below). The angle ψ is relative to the angle of orientation of the antenna. Since, by the definition of gain, the integral of gain over the unit sphere should be 4π, we can compute the normalization constant $G_f^n(0)$ for this family. It turns out that:

$$G_f^n(0) = \frac{(n+1)^2}{\pi H(n/2)^2}, H(k) = \frac{k+1}{4^{k+1}} \cdot \binom{2(k+1)}{k+1} \tag{6.3}$$

Putting $n = 2, 4, 6$ in Eq. (6.3) we get $G_f^{(2)}(0) = 5.09$, $G_f^{(4)}(0) = 9.05$, $G_f^{(6)}(0) = 13.038$. It is worthy to note that in general $2n+1$ is a reasonable approximation of $G_f^n(0)$ as n grows.

From now on, we simply denote the realistic gain function $G_f^n(\cdot)$ by $G(\cdot)$.

6.3.2 The ideal directional model

In this section we show how to translate the 3D model into a 2D ideal model, as done in literature [126]. As a theoretical counterpoint we introduce a simple idealised directional gain function that captures the idea of a beam of width 2θ centered at the angle of orientation. The gain everywhere is a uniform non-zero value within this beam is and zero everywhere outside. We call this the *ideal directional model* or just the *ideal* model. We denote the ideal gain function $G_i^\theta(\cdot)$, using the subscript i for "ideal" to differentiate it from the fading model above. This gain function can be explicitly computed by integrating the uniform gain over the surface of the sphere centered at the antenna and equating this value to 4π. By doing this we find:

$$G_i^\theta(\psi) = \begin{cases} \frac{2}{1-\cos(\theta)} & 0 \leq |\psi| \leq \theta \\ 0 & |\psi| > \theta \end{cases} \tag{6.4}$$

In the omnidirectional case, under the assumption of uniform unit gain in all directions, the authors in [142] showed that in the setting where m nodes are distributed uniformly at random in a unit disc and if each node can communicate with another node at distance r from it, then, the random graph thus formed is connected with probability tending to 1 as $m \to \infty$ if and only if the radius within which two nodes can communicate is:

$$r_o(m) = \sqrt{\frac{\log(m) + c(m)}{m\pi}} \tag{6.5}$$

where $c(m) \to \infty$ as $m \to \infty$. In the following when the number of nodes m is understood, we will often just use r_o to denote this radius.

Restating this in terms of power, using the Friis transmission equation, we can say that if P_R^* is the minimum received power required for the signal to be correctly received, then, since $G_{R_x} = G_{T_x} = 1$, the omnidirectional transmission power required is:

$$P_T^o = P_R^* \left(\frac{4\pi r_o}{\lambda}\right)^2 \tag{6.6}$$

In this work we will use this value of P_T^o as a scaling constant for the transmission power used, and r_o as a scaling constant for distances. In particular we will say that the transmission power used by our directional antennas is $P_T^d = \alpha P_T^o$.

We will use α as a parameter to tune the antenna transmission power for the rest of this work. To find the furthest distance, $r_G(\alpha)$, that an antenna u with gain function $G(\cdot)$ and power parameter α can communicate we have to find the largest x such that the power received by an antenna v which is at distance x from the transmitting antenna u is at least P_R^*, i.e., we have to find x such that:

$$\max_{\beta_1,\beta_2 \in [-\frac{\pi}{2},\frac{\pi}{2}]} P_T^d G(\beta_1) G(\beta_2) \left(\frac{\lambda}{4\pi x}\right)^2 \geq P_R^* \tag{6.7}$$

where β_1 is the angle between the ray defining the angle of orientation of the transmitter and the line segment $u \rightarrow v$ and β_2 is defined analogously for the receiver (see Fig. 6.2). Note that the gain function $G(\cdot) \neq 1$ in the directional model.

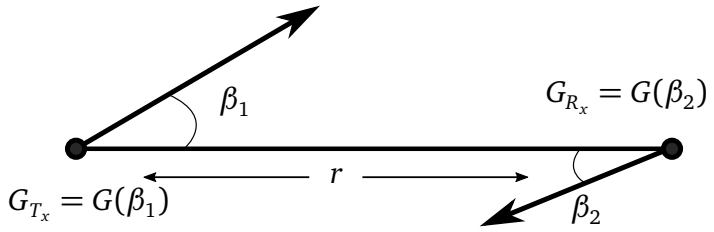


Figure 6.2: Connecting transmitter to receiver.

Solving this by putting the values of P_R^* and P_T^d, and observing that $G(\cdot)$ is maximized at $G(0)$ by definition, we get that:

$$r_G(\alpha) = \sqrt{\alpha} \cdot G(0) \cdot r_o \tag{6.8}$$

Hence by varying α we can control the distance to which the connections can be made. Note that the maximum distances for the two models can be derived by using the values of $G_f^n(0)$ and $G_i^\theta(0)$.

Unlike in the simple Random Geometric Graph (RGG) model studied in [142], connectivity between two antennas in the directional setting does not depend only on the distance between them; it also depends on their angles of orientation. We now study the situation where the antennas are located in the 2D plane and each antenna picks its angle of orientation uniformly at random from $[0, 2\pi]$.

Assuming that the receiver has fixed its angle of orientation (β_2 relative to the line joining receiver to transmitter) we compute the probability of connectivity at distance r by integrating over the range of values of the angle of orientation of the transmitter, β_1, within which the received power is at least P_R^*. This gives us:

$$g_G(r) = \int_{-G^{-1}\left(\frac{r^2}{\alpha r_o^2 \cdot G(0)}\right)}^{G^{-1}\left(\frac{r^2}{\alpha r_o^2 \cdot G(0)}\right)} \frac{1}{2\pi^2} \cdot G^{-1}\left(\frac{r^2}{\alpha r_o^2 \cdot G(\beta_1)}\right) d\beta_1 \tag{6.9}$$

The above function is non-trivial to compute in the fading model, but in the ideal directional model, under the gain function $G_i^\theta(\cdot)$ it reduces to

$$g_{G_i^\theta}(r) = \begin{cases} \frac{\theta^2}{\pi^2} & 0 < r \leq \sqrt{\alpha} \cdot \frac{2}{1-\cos(\theta)} \cdot r_o \\ 0 & \text{otherwise} \end{cases} \tag{6.10}$$

This is simply the probability that the receiver lies in a randomly chosen sector of radius $r_i(\alpha)$ with angle 2θ centered at the transmitter and vice-versa.

We also compute the probability, γ_G, that a node u connects with another node v that is placed uniformly at random in the disc of radius $r_f(\alpha) = \sqrt{\alpha}G(0)r_o$ centered at u in the realistic fading model. This quantity is going to be critical in our study of network connectivity. Conditioning on the position of u and integrating over the disc we get:

$$\gamma_G = \int_{x=0}^{\sqrt{\alpha}r_o G(0)} g_G(x)\frac{2x}{\alpha r_o^2 G(0)^2}dx \tag{6.11}$$

An important point to note here is that γ_G does *not* depend on α as long as $\alpha > 0$. This can be seen by changing variables in Eq. (6.11), replacing x with z where $x = \sqrt{\alpha}r_o z$.

For the ideal model we compute the probability, γ_{G_i}, that a node u connects with another node v that is placed uniformly at random in the disc of radius $r_i(\alpha) = \sqrt{\alpha}G_i^\theta(0)r_o$ centered at u. By substituting in Eq. (6.11) the probability of connectivity at distance x, i.e., $g_{G_i^\theta}(x)$ given by Eq. (6.10), we get $\gamma_{G_i} = \theta^2/\pi^2$.

6.3.3 A random graph model

We model a mesh network of directional antennas as a random geometric graph, $H = (V, E)$, whose nodes are distributed uniformly at random in a unit disc in $\mathbb{R}^2$. Each node $u \in V$ is equipped with a directional antenna that chooses its angle of orientation ξ_u uniformly at random from $[0, 2\pi]$ independently of all other nodes. The other parameters of the model are a power level α as defined in Section 6.3 and a gain function $G(\cdot)$.

For convenience we will use the following notation to refer to random graphs modeling networks using the directional fading and ideal directional model:

- DF-RGG(m, n, α): a random graph formed as above on m nodes with $G = G_f^n(\cdot)$ and power parameter α, briefly DF-RGG when the parameter values are understood.
- DI-RGG(m, θ, α): a random graph formed as above on m nodes with $G = G_i^\theta(\cdot)$ and power parameter α, briefly DI-RGG.

To show that the edge-independence assumption does not hold, let us consider the simpler ideal model. Assume there are three nodes x, y and z which are placed such that their pairwise distances are all equal to some $r > 0$, i.e., they are placed at the vertices of an equilateral triangle of side length r. Consider a value of θ that is smaller than 30 degrees and an α large enough to ensure that each pair can communicate if the antenna orientations are correct. For a given pair of

nodes, say x, y, the probability that they are connected is θ^2/π^2. But clearly the probability of all three pairs being connected is 0 which is less than θ^6/π^6 which is what it would have been if the probabilities of the edges being formed were independent. Hence, we find that the independence assumption does not hold and so the theory developed under this assumption cannot be used in this case. We will now show how this problem can be handled.

6.4 Connectivity

In this section we show that highly directional antennas achieve network connectivity at a much lower power level than omnidirectional antennas. This is a somewhat counterintuitive result that we feel has major implications for the design of mesh networks.

We now present our main theorem on connectivity. The key factor in this theorem is the probability of connectivity γ_G associated with an antenna with gain function G. As we showed in Section 6.3, this probability is independent of the transmission power and hence is a property of the antenna model alone and depends only on the gain function G. Our main theorem is:

Theorem 5. *Suppose we are given a set V of m nodes distributed uniformly at random in a unit disc B of $\mathbb{R}^2$ and each node is equipped with an antenna with gain function G that is: (a) non-zero in $[-\pi, \pi]$, (b) symmetric around the angle of orientation and, (c) monotonically decreasing away from the angle of orientation. Assume that each antenna has transmission power that allows it to transmit to a distance of $r > 0$ in its direction of maximum gain. Denote by γ_G the probability that two nodes that lie within distance r of each other are connected.*

We construct a random graph model D-RGG(m, G, r) by placing edges between each pair of points that can communicate with each other, and for this we have that:

$$P(\text{D-RGG}(m, G, r) \text{ is connected}) \to 1$$

as $m \to \infty$ if and only if:

$$\pi r(m)^2 \gamma_G = \frac{\log m + c(m)}{m} \tag{6.12}$$

where $\lim_{m\to\infty} c(m) = \infty$ as $m \to \infty$.

Proof. It comes directly from the results on vertex-based model proposed in [128]. □

We note that the optimal radius suggested by Theorem 5 is simply the optimal radius for omnidirectional antennas given in [127] scaled by a factor of $1/\sqrt{\gamma_G}$. This implies that the radius of connectivity is *larger* than that for omnidirectional antennas, since $\gamma_G < 1$, and appears to run counter to our claim that random directional mesh networks require lower power. However, as

we have already seen the directionality of an antenna means that it can achieve a much larger transmission range, at least in the direction of orientation, and so we will find that the power required is much lower than that required for omnidirectional antennas.

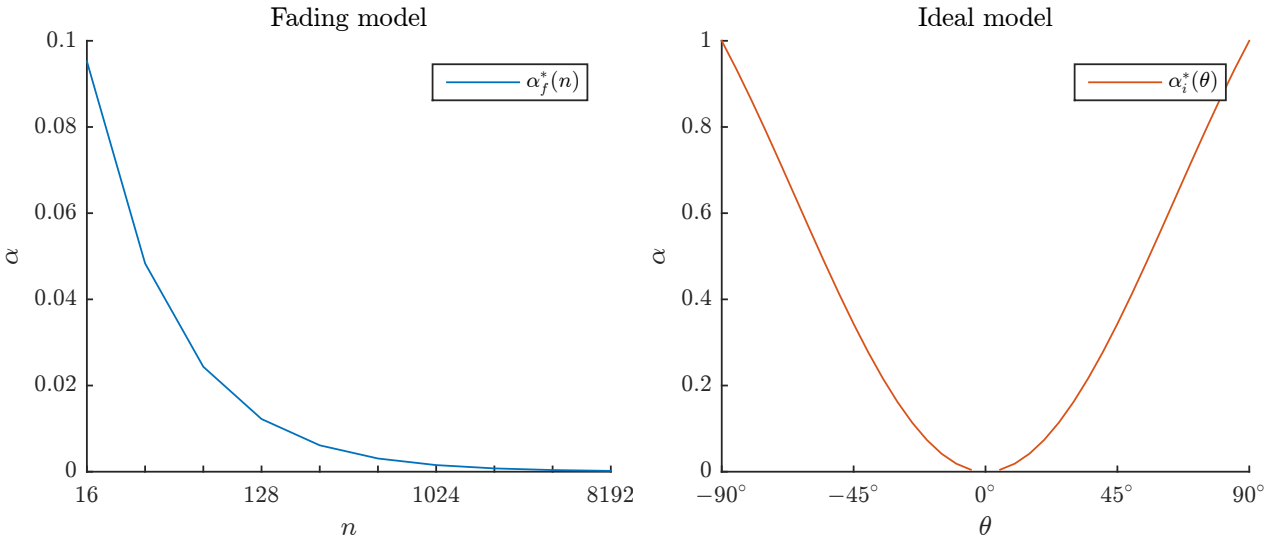
6.4.1 Optimal power for connectivity

From Theorem 5 we deduce that the optimal radius r_d of connectivity of the directional model with gain function G is given by $r_d = r_o/\sqrt{\gamma_G}$. The power level α that reaches the maximum distance $\sqrt{\alpha}G(0)r_o$ equal to r_d will be called the *optimal power level* α^* and is given in the fading and ideal model by, respectively:

$$\alpha_f^*(n) = \frac{1}{\gamma_{G_f} G_f^n(0)^2} \tag{6.13}$$

$$\alpha_i^*(\theta) = \left(\frac{\pi(1-\cos(\theta))}{2\theta}\right)^2 \tag{6.14}$$

In Fig. 6.3, after computing $\gamma_f(n)$ numerically for $n = 16, 32, 64, \ldots, 8192$, we plot $\alpha_f^*(n)$ versus n (see Fig. 6.3a) and $\alpha_i^*(\theta)$ versus θ (see Fig. 6.3b).



(a) The fading model (parameter n).

(b) The ideal model (parameter θ).

Figure 6.3: Optimal power level vs model parameters.

It is worth noting that α^* depends on n in the fading model and on θ in the ideal model. Since the gain in the direction of orientation is a measure of how "directional" the antenna beam is, i.e., how concentrated the signal is in the direction of orientation, the inverse relationship of the

optimal power level to $G(0)^2$ implies that the power level required for connectivity decreases as the directionality of the antenna increases.

n	$\theta(n)$ (degrees)	γ_{G_i}	$\alpha_i^*(n)$	γ_{G_f}	$\alpha_f^*(n)$
16	16.74	0.008652	0.0519	0.009640	0.0952468
32	11.88	0.004357	0.0263	0.004896	0.0483396
64	8.41	0.002186	0.0132	0.0024673	0.024355
128	5.95	0.001095	0.0066	0.0012385	0.0122246
256	4.21	0.000548	0.0033	0.00620	0.00612419
512	2.98	0.000274	0.0016	0.00031	0.00306508
1024	2.10	0.000137	8.34e-04	0.00015	0.00153329
2048	1.49	6.86e-05	4.17e-04	7.76e-05	0.00076683
4096	1.05	3.42e-05	2.08e-04	3.88e-05	0.000383462
8192	0.74	1.71e-05	1.04e-04	1.94e-05	0.000191743

Table 6.1: The parameter n, associated angle $\theta(n)$ and corresponding optimal power levels and connectivity probabilities.

It is not a priori clear how to determine which of the two models, ideal or fading, is more power efficient. In order to compare them, we propose to study the *half-power beamwidth* (or, simply, the *halfbeam*) for antennas with realistic gain function [141].

For an antenna of parameter n, the halfbeam is defined as the angle 2χ between the two directions in which the gain $G_f^n(\chi)$ is one half the maximum value, that is, χ such that:

$$G_f^n(\chi) = \frac{1}{2} G_f^n(0) \cos^n(0)$$

Solving the above equation, we obtain that the halfbeam of an antenna of parameter n is the angle:

$$2\chi = 2\cos^{-1}\left(\sqrt[n]{1/2}\right)$$

Thus, we associate the fading model whose gain function has parameter n to the ideal model of parameter (see Tab. 6.1):

$$\theta(n) = \cos^{-1}\left(\sqrt[n]{1/2}\right)$$

With this correspondence, we report the optimal power levels in Table 6.1: we compute $\alpha_i^*(\theta(n))$ by recalling $\gamma_{G_i} = \theta^2/\pi^2$ and using Eq. (6.4). After computing γ_G by numerical in-

tegration (see Eq. (6.11)), we derive $\alpha_f^*(n)$ using Eq. (6.2). Note that the values of $\alpha_i^*(n)$ in Tab. 6.1 zoom into Fig. 6.3b since $\theta(n)$ lies in $[0.74, 16.74]$.

In Tab. 6.1 we report the connectivity probabilities of the fading model with different values of the parameter n and those of the associated DI-RGG, i.e., the ideal model with parameter $\theta(n)$. As we see, they almost coincide thus validating the engineering intuition that guided us in making this association.

This connectivity probability is for a pair of points but when we come to network-wide connectivity the models differ: the optimal power level for DI-RGG is about half of that for the corresponding version of DF-RGG.

This is because each DI-RGG antenna covers at a smaller area (i.e., halfbeam) than the one considered in DF-RGG but with a better (uniform) gain value. This shows that for network connectivity the halfbeam assumption is overly optimistic and gives us lower power levels than required.

Nevertheless for all values of n the optimal power for the fading model is double that of the ideal model, and we can state as a rule of thumb that $\alpha_f^*(n) = 2\alpha_i^*(\theta(n))$. This is an important input for the design of a connected directional mesh in which the directional antennas transmit at power level at most α.

6.5 Experimental Evaluation

In this section we experimentally test our results on connectivity in directional meshes. We implemented the algorithm in C++. We followed the communication model for the DF-RGGs and DI-RGGs described in Section 6.3.

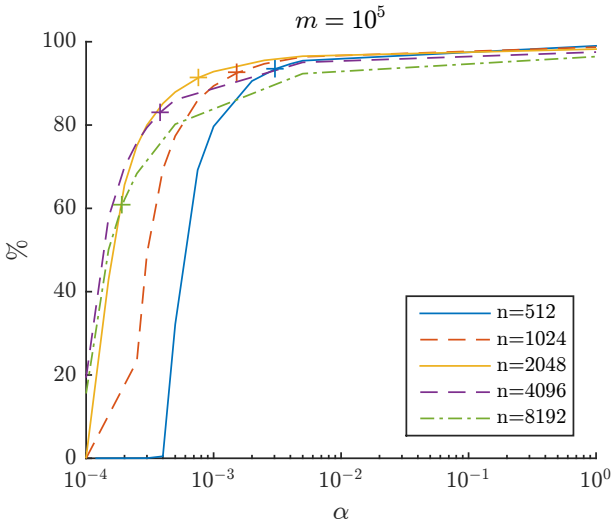
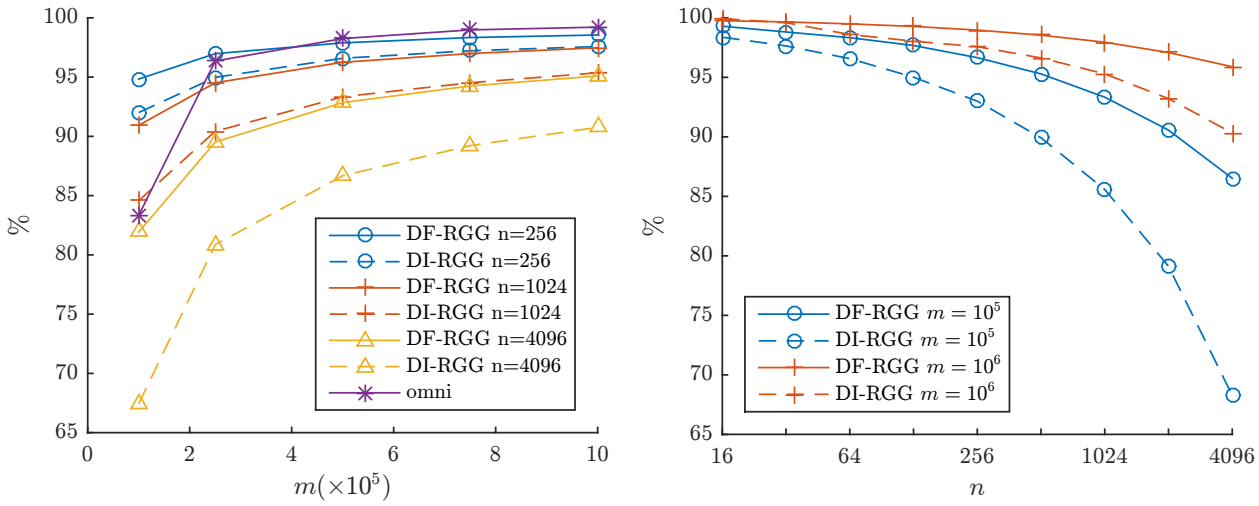


Figure 6.4: The percentage of connectivity versus α in the fading model.

Our main metric in this study is what we call the *percentage of connectivity* or *connectivity percentage*, which is defined as the percentage of nodes in the largest connected component. First we validate our main result on the optimal power level for the fading model. Fig. 6.4 shows the percentage of connectivity versus power level α for several values of the fading parameter n. For each value of n, the optimal power level $\alpha_f^*(n)$ is highlighted with a small cross. As one can see, whenever $n \leq 4096$, the optimal power level derived in Eq. (6.13) is very accurate. Indeed, at $\alpha_f^*(n)$, the percentage of connectivity reaches the maximum value and after that, it remains stable. In other words, extra power would not significantly improve the connectivity. For $n = 8192$, $\alpha_f^*(8192)$ is less accurate since the percentage of connectivity increases for $\alpha > \alpha_f^*(8192)$. This eventually shows that the connectivity probability is slightly overestimated in such extreme value of n.



(a) m varies (b) n varies

Figure 6.5: The percentage of connectivity.

The remaining experiments test the percentage of connectivity in DF-RGG and DI-RGG at the optimal power level α^*, reported in Tab. 6.1.

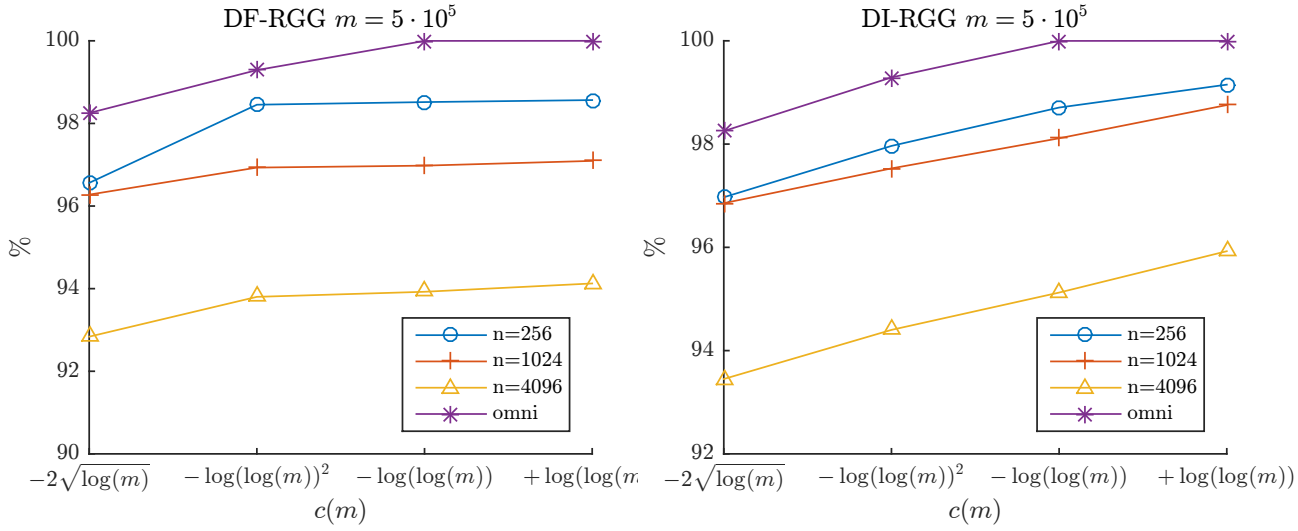
Fig. 6.5 shows that the percentage of connectivity achieved in a directional mesh is high and comparable to that of an omnidirectional mesh, although the power used by directional models is well below P_T^o which is conventionally set to 1 in our experiments. It also appears that for a more directional model to achieve a high connectivity percentage, we need a higher density than we need for a less directional model. Nonetheless, it is interesting to point out that when m is small, moderate directionality may achieve higher connectivity than omnidirectional networks, i.e., reaching further nodes within a (sufficiently wide) sector is more effective for achieving

connectivity than reaching nodes that do not lie as far but are located all around the antenna.

We then verified whether the power level derived by Eq. (6.13) and Eq. (6.14) is necessary for achieving connectivity. For this purpose, we varied the connectivity radius in Eq. 6.5 below the optimal threshold using:

$$c(m) = \{-\log\log(m), -\log^2\log(m), -2\sqrt{\log(m)}\}$$

Changing $c(m)$, the radius reduces from r_o to r, and the directional optimal power level is scaled by factor $F = (\frac{r}{r_o})^2$. The scale coefficients F used in Fig. 6.6 for the three values of $c(m)$ are $\{0.64984, 0.41382, 0.37443\}$. We take $m = 5 \cdot 10^5$ here.



(a) in DF-RGG (b) in DI-RGG

Figure 6.6: The percentage of connectivity vs $c(m)$.

We note in Fig. 6.6 that the more we decrease the power level, the greater the loss in connectivity. The trends of the connectivity curves are the same for all values of n, sharpening for higher values of n.

Conclusions In this work we have argued that connected mesh networks can be built using directional antennas and that such mesh networks can operate with much lower power than mesh networks built with isotropic omnidirectional antennas. We have also demonstrated how a simple idealised gain function can be used to approach mesh network design where the antennas have a more realistic and complex gain function.

Chapter 7

Precise Localization with Directional Antennas[1]

7.1 Introduction

In this work we present a localization algorithm called DIR that uses a drone as the mobile anchor, but this time the drone is equipped with directional antennas instead of an omnidirectional one.

Figure 7.1: A classical directional antenna.

Due to the use of the directional antennas, there is no need to first roughly estimate the sensor position. Namely, the area where the sensor can reside when it has collected three drone's beacons with different orientations is very limited, and thus a single trilateration is sufficient to precisely localize the sensor.

[1]Part of this work has been published to ICDCN 2018 - Proceedings of the 19th International Conference on Distributed Computing and Networking

7.2 Model and Assumptions

We consider a network of n sensors, (or simply, n nodes), deployed in a rectangular area Q of size $Q_x \times Q_y$, where $Q_x \geq Q_y$ without loss of generality. For the sake of simplicity, we assume the area is obstacle free and the ground is flat so as the drone is always in the line of sight with the nodes to be measured. Our main goal is to localize with a user-required precision each sensor inside the deployment area using a drone equipped with directional antennas. We assume that the drone can measure its altitude with negligible error, i.e., considering $\gamma_h = 0\,\text{m}$ and also $\gamma_d = 0\,\text{m}$.

In our proposed solution, each non-localized node performs a single trilateration and applies the least-squares-error method to find its estimated position. The drone is equipped with directional antennas. Each antenna is assumed to transmit the beacon in a circular sector, centered at the antenna position, of radius r, beamwidth 2θ and orientation ψ. Fig. 7.2 depicts a sector centered at w with direction ψ. In the following, a sector will be uniquely identified by its center and its orientation because we assume that all the sectors have the same radius and the same beamwidth.

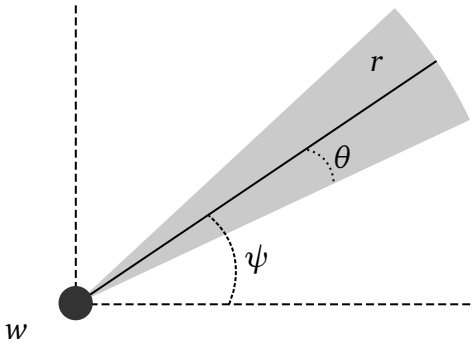


Figure 7.2: The sector parameters.

In our solution the directional antennas cover six different sectors, as illustrated in Fig. 7.3.

From now on, we express each orientation ψ as a pair: *type* (up, dw, hor) and *polarity* ($+$, $-$), as reported in Tab. 7.1, where dw denotes down and hor indicates horizontal. Each sensor saves the information received from a certain orientation in its local register R. The matching between the register positions and the orientations is also reported in Tab. 7.1.

With respect to the available hardware we can have different solutions to implement the set of directional antennas. The drone can send at the same time the beacons in all six orientations using an array of directional antennas [143]. The drone can also emulate the array of directional antennas by rotating a single orientation antenna or just rotating itself if the antenna is fixed. In the latter case, the drone has to pause its flight remaining in hovering mode thus spending more energy than in the former case [144]. Instead, if the drone is equipped with an array of

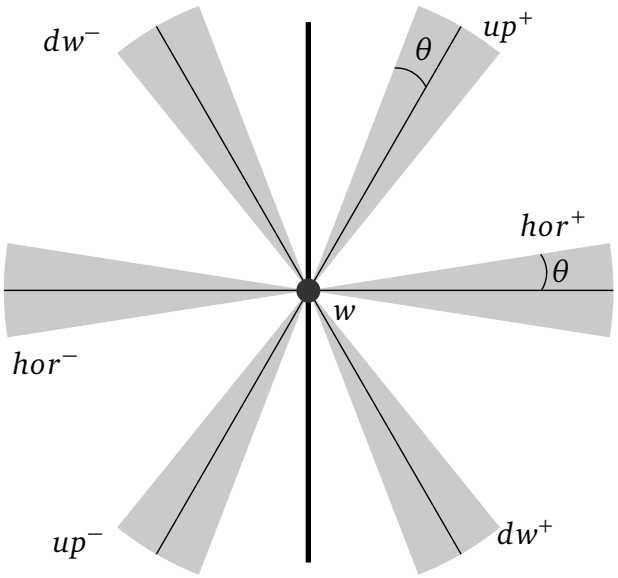


Figure 7.3: Possible antenna orientation.

$R[d]$	ψ	type	polarity
0	0	hor	+
1	$(1/3)\pi$	up	+
2	$(2/3)\pi$	dw	−
3	π	hor	−
4	$(4/3)\pi$	up	−
5	$(5/3)\pi$	dw	+

Table 7.1: Antenna orientation and register R.

directional antennas (hence more expensive hardware), it can slow down instead of hovering for measurements. Note that the sensors on the ground are equipped with omnidirectional antennas.

The drone takes measurements at pre-established points of the static path. The projections of such points on the ground are called *waypoints*. To take a measurement, the drone acts as follows. At the waypoint, by means of its directional antennas, the drone sends beacons with a unique identifier which depends on the coordinates of the waypoint, the current timestamp and the orientation used to send such a beacon. Each sensor on the ground that can hear a beacon replies to the drone with an ack message that contains its ID, the current timestamp, and the identifier of the beacon received. The antenna (for simplicity, we say the drone) computes the distance between itself and the node using the round-trip time of the beacon message and then sends a message with the computed distance to the sensor. Once a sensor has collected at least three measurements from different orientations, it can locally apply the trilateration algorithm and discovers its position.

7.3 The DIR algorithm

In this section, we describe the proposed directional localization algorithm DIR. From now on, node P is short form for the sensor that resides at point P in Q. Since, in principle any point P in Q is candidate to contain a sensor, with a little abuse of notation, we denote P indistinctly as the point P or the node P. Moreover, we use the notation $\overline{PQ}$ to denote the distance between two points P and Q.

The aim of the DIR algorithm is to obtain *precise localization* for each point P in Q. We define the precise localization as follows:

Definition 2. *The node P is precisely localized if the drone chooses three ranging waypoints $w_1(P)$, $w_2(P)$ and $w_3(P)$ for P such that they satisfy the following constraints (see Section 2.5 for a quick check of the terminology):*

1. *the d_{max} constraint, which controls the reachability for each point P in Q: $\overline{w_i(P)P} \leq d_{max}$ for $i = 1, 2, 3$;*

2. *the d_{min} constraint, which controls the ground precision ϵ_d for P in Q: $\overline{w_i(P)P} \geq d_{min}$ for $i = 1, 2, 3$;*

3. *the β_{min} constraint, which controls the collinearity of the drone with P: $\beta_{min} \geq \pi/3 - 2\theta$, where $\theta = \arctan\left(\frac{I_w/2}{d_{min}}\right)$ and I_w is the minimum distance between any two waypoints;*

4. *the non-linearity constraint which controls the collinearity among waypoints:* $w_1(P)$, $w_2(P)$ *and* $w_3(P)$ *cannot belong to the same straight line.*

7.3.1 Generating the Static Path

The static path D traversed by the drone during its mission is formed by vertical and horizontal scans and it is depicted as a dashed line in Fig. 7.4. The drone's mission starts outside the deployment area at $F = (-F_x, 0)$, where $F_x = d_{\min}/2$. The first vertical scan runs parallel to the left border of Q. The last vertical scan runs parallel to the right border of Q up to $E = (Q_x + F_x, y_E)$, where y_E can be either 0 or Q_y depending on the parity of the number of vertical scans. The path finishes at E, but then the drone goes back to its home F following the direct line between E and F. Each vertical scan has length Q_y. The vertical scans are connected by horizontal segments that run on the border of Q.

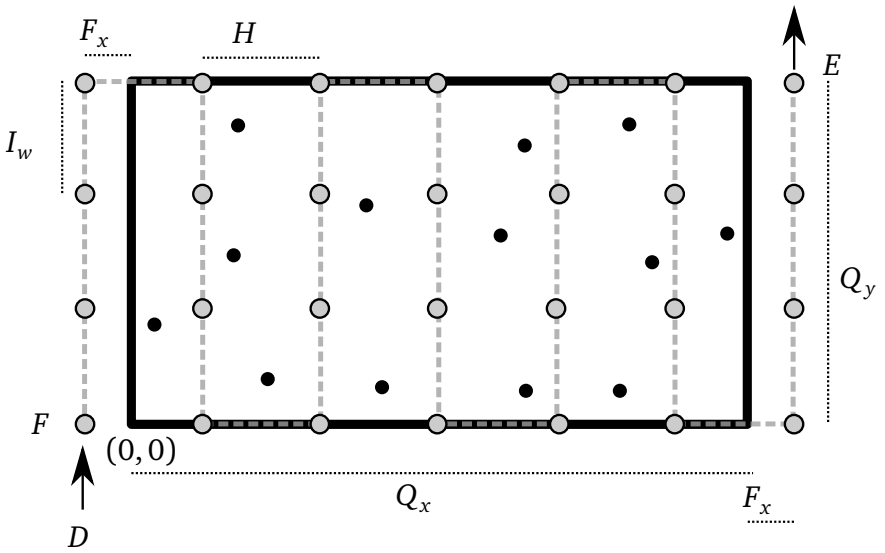


Figure 7.4: The static path (dashed line) D and the deployment area Q.

In Fig. 7.4, the gray dots represent the waypoints, while the black dots are the sensors inside Q. Here I_w is called *inter-waypoint distance*. The measurements are taken only from the vertical scans, and thus the waypoints reside on the vertical scans only. From each vertical scan S, the points of Q that we can precisely measure using any type of orientation are those at distance at least $d_{\min}/2$ and at most $(d_{\max} - 2I_w)/2$ from the scan S. In fact, although the antennas of type hor can precisely measure the points at distance from $d_{\min}$ up to $d_{\max}$ from S, the most stringent limits for the precise measurements come from the antennas of type up and $down$. Therefore, any two consecutive vertical scans are fixed at distance no greater than at distance greater than $(d_{\max} - d_{\min} - 2I_w)/2$. From now on, the *inter-scan distance* is denoted by H. Having fixed the first and the last scan

respectively at $-F_x$ and $Q_x + F_x$, the length of H is fixed in order to evenly distribute $Q_x + 2F_x$ in stripes of width as tight as possible to the maximum value $(d_{\max} - d_{\min} - 2I_w)/2$. In this way, the whole area is covered without wasting path length. Finally, observe that the left and right stripes of Q of width $d_{\min}/2$ adjacent to each vertical scan S are not measured by S to satisfy the $d_{\min}$ constraint. The left and right stripes adjacent to S are then measured by the scan that precedes S and the scan that follows S.

7.3.2 Designing the Drone's Mission

Given the altitude h of the drone, fixed I_w, we compute $d_{\min}$ and the beamwidth 2θ where $\theta = \arctan\left(\frac{I_w/2}{d_{\min}}\right)$. During the flight, for each waypoint, the drone sends the message beacon along the six different orientations: up^+, up^-, dw^+, dw^-, hor^+, and hor^-. Algorithms 2 and 3 in the following describe the drone's and sensors' behavior during the mission. Specifically, when a sensor receives the drone message from orientation ψ, for $\psi = d \cdot \pi/3$ and $d = 0, \ldots, 5$, it first checks whether the d-th location of its register R (see Tab. 7.1) is empty or not. In the first case, the sensor is receiving for the first time the beacon from the orientation d. In the second case, i.e., the sensor has already heard that beacon from orientation d, the message is ignored. When the sensor receives the beacon for the first time, the sensor replies to the drone with an ack message. The drone infers the distance from the time of the round-trip message, and sends to the sensor the measurement, that the sensor stores in the register. Once the sensor has received three measurements from different orientations, it applies the trilateration procedure to compute its position.

Algorithm 2 Drone behavior

```
1:  for all w ∈ W do                                   ▷ W = Waypoints set
2:      SENDBEACON(ID_b, t_d), ∀d = 0...5              ▷ ID_b = beacon ID
3:      P_resp ← RECVACKFROMSENSOR()
4:      for all id ∈ P_resp do
5:          grdist ← COMPDIST(ID_s, t_d, t_s)          ▷ ID_s = sensor ID
6:          if grdist ≥ d_min then
7:              SENDDIST(ID_s, dist)
8:          end if
9:      end for
10: end for
```

It is worthy to note that our algorithm is suitable to localize sparse wireless sensor networks.

Algorithm 3 Sensor behavior

```
1: ranged ← 0
2: while ranged < 3 do
3:     b = WAITBEACON()
4:     d = b.orientation
5:     if R[d] = 0 then
6:         dist ← SENDACK(t_s)
7:         R[d] ← dist
8:         ranged ← ranged + 1
9:     end if
10: end while
11: TRILATERATION(ranged)
```

Indeed, if many nodes simultaneously acknowledge the drone's beacon, interferences may occur. For handling a dense network, a possible solution is that only few nodes are authorized to reply to the drone's beacon, and later the sensors localized will act as anchor nodes for the remaining non-localized sensors in Q.

7.4 Analysis

In the following we analyze the localization mission. First, we prove that it is possible to precisely measure any point in Q. Then, we study how to set the DIR parameters to achieve the a-priori required localization precision.

7.4.1 Correctness of Algorithm

Each node has to collect at least three measurements in order to perform the trilateration. The measurements need to satisfy four constraints: $d_{\min}$, $d_{\max}$, $\beta_{\min}$ and *non-linearity* (see Definition 2).

To respect the *non-linearity* constraints, it is sufficient to choose three waypoints from two different scans.

As regards to the $\beta_{\min}$ constraint, we observe that to be as close as possible to the best (largest) possible minimum angle at $\pi/3$, the three waypoints must have different types of orientations. For this reason, the trilateration starts only when the sensor has three elements in its register. In Theorem 7 we will prove that the algorithm DIR yields $\beta_{\min} = \pi/3 - 2\theta$.

The $d_{\min}$ and $d_{\max}$ constraints are proved in Theorem 6.

Theorem 6. *Fixed the half beamwidth*

$$\theta = \arctan\Big(\frac{I_w/2}{d_{min}}\Big), \tag{7.1}$$

each point of Q can be measured by the three different orientations hor, up and dw verifying the d_{min} and d_{max} constraints.

Proof. Let first consider the antenna's orientation hor^+ with $\theta = \arctan\Big(\frac{I_w/2}{d_{\min}}\Big)$. As observed from Fig. 7.6a, each sector with orientation hor^+ is able to cover at least a rectangular of height I_w from distance $d_{\min}$ up to distance $d_{\max}$ (see Fig. 7.6a) if θ satisfies Eq. (7.1). Since the first scan of our path is at distance $F_x = d_{\min}/2$ from Q, it is easy to prove that any point in Q can be covered (thus, measured) satisfying the $d_{\min}$ and $d_{\max}$ constraints of Definition 2 using antenna's orientation hor^+. The same holds for the orientation hor^- because the last scan is outside Q, and precisely $F_x = d_{\min}/2$ beyond Q.

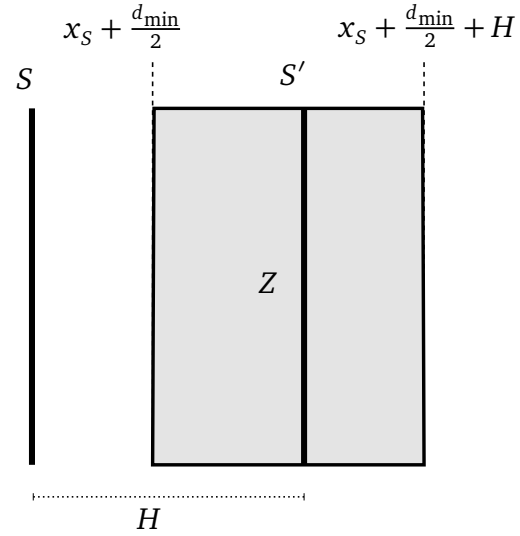


Figure 7.5: The stripe Z of Q parallel to scan S.

For the other orientation of the antennas, consider a scan S and the antennas at its waypoints. We aim to prove that the orientations dw and up are able to cover the stripe Z (see Fig. 7.5) of Q parallel to S that starts at distance $d_{\min}/2$ and finishes at distance $d_{\min}/2 + H$ from S, without leaving any uncovered space.

Consider now the antenna's orientation dw^+. To prove that they cover Z, we will show that the sector centered at the waypoint w_1 (see Fig. 7.6b) intercepts the segment T_1P_1 of length $\overline{T_1P_1} \geq I_w$ on the line parallel to S and at distance $d_{\min}/2$ from S. The same sector intersects the segment T_2P_2 of length $\overline{T_2P_2} \geq I_w$ on the line parallel to S and at distance $d_{\min}/2 + H$ from S.

Let us start with $\overline{T_1P_1}$. By applying simple trigonometric rules to the triangles $w_1Q_1T_1$ and $w_1P_1T_1$, where $Q_1 = (d_{\min}/2, y_{w_1})$, we obtain

$$\overline{T_1P_1} = \overline{Q_1T_1} - \overline{Q_1P_1}$$

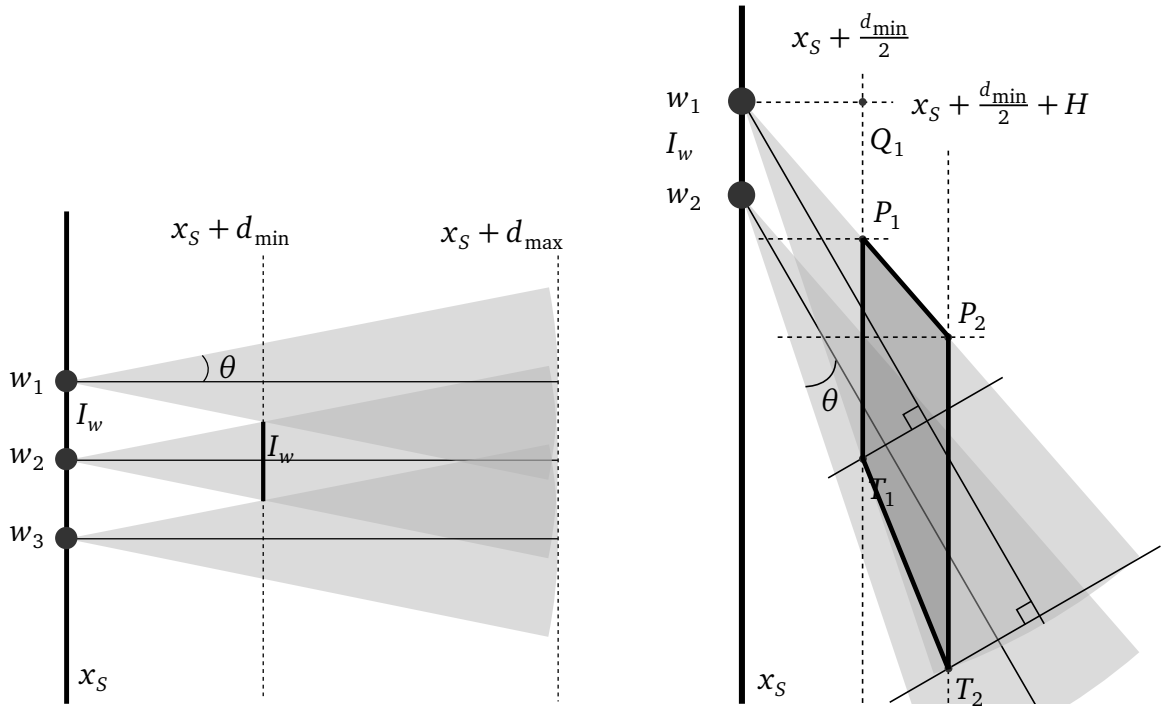
where

$$\overline{Q_1T_1} = \frac{d_{\min}}{2}\cot\Big(\frac{\pi}{6} - \theta\Big),$$

$$\overline{Q_1P_1} = \frac{d_{\min}}{2}\tan\Big(\frac{\pi}{3} - \theta\Big).$$

After algebraic manipulations we find $\overline{T_1P_1} > I_w$ when θ satisfies Eq. (7.1). Then comparing the two similar triangles $w_1T_1P_1$ and $w_1T_2P_2$, we learn that $\overline{T_2P_2} > \overline{T_1P_1} > I_w$. Hence, repeating the same reasoning for all the waypoints of S, we conclude that the sectors with orientation dw^+ centered at the waypoints on the static path S cover Z.

Not only all the sensors that fall in Z are covered, but also they can be measured from the antennas with orientation dw^+ satisfying the $d_{\min}$ and $d_{\max}$ constraints. Indeed, the slant distance from w_1 to any point $P = (x_P, y_P)$ that falls in Z when projected on the ground is equal to the double of the distance $x_P - x_S$ from S to P. Thus, since $d_{\min} \leq 2(x_P - x_S) \leq d_{\max} - 2I_w$, the $d_{\min}$ and $d_{\max}$ constraints are verified.



(a) Horizontal pattern.

(b) Down pattern.

Figure 7.6: The antenna beam.

A similar reasoning can be repeated for the other remaining orientations.

The previous discussion has proved that a largest part of Q can be reached by all the six different orientations. However, on the border of Q some orientations can be missing. We divide the analysis in four different cases, as illustrated in Fig. 7.7.

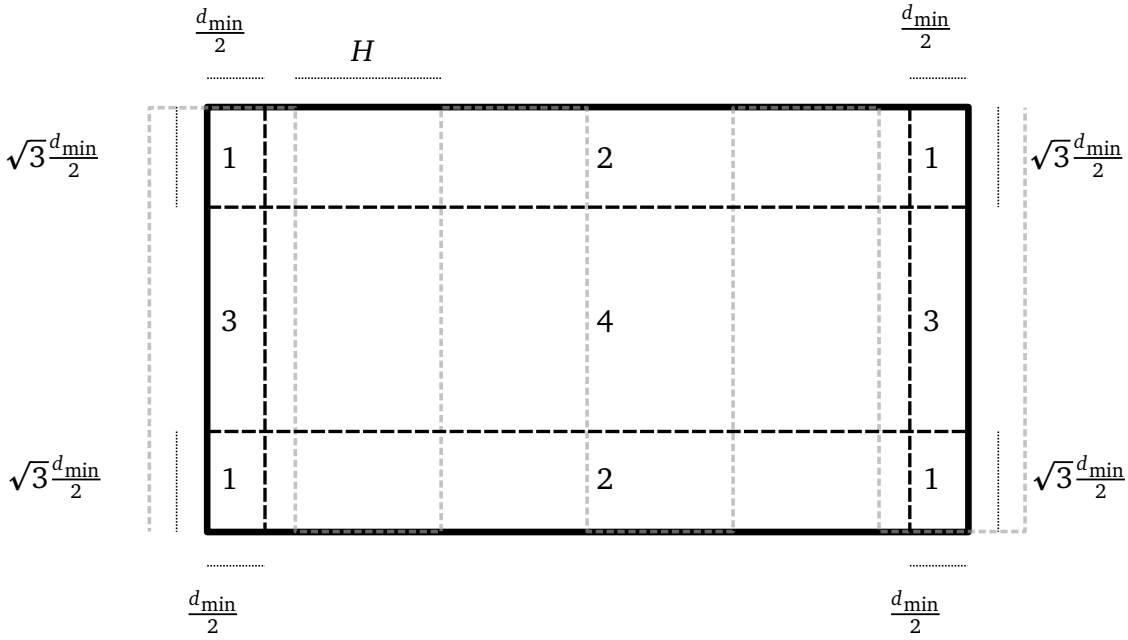
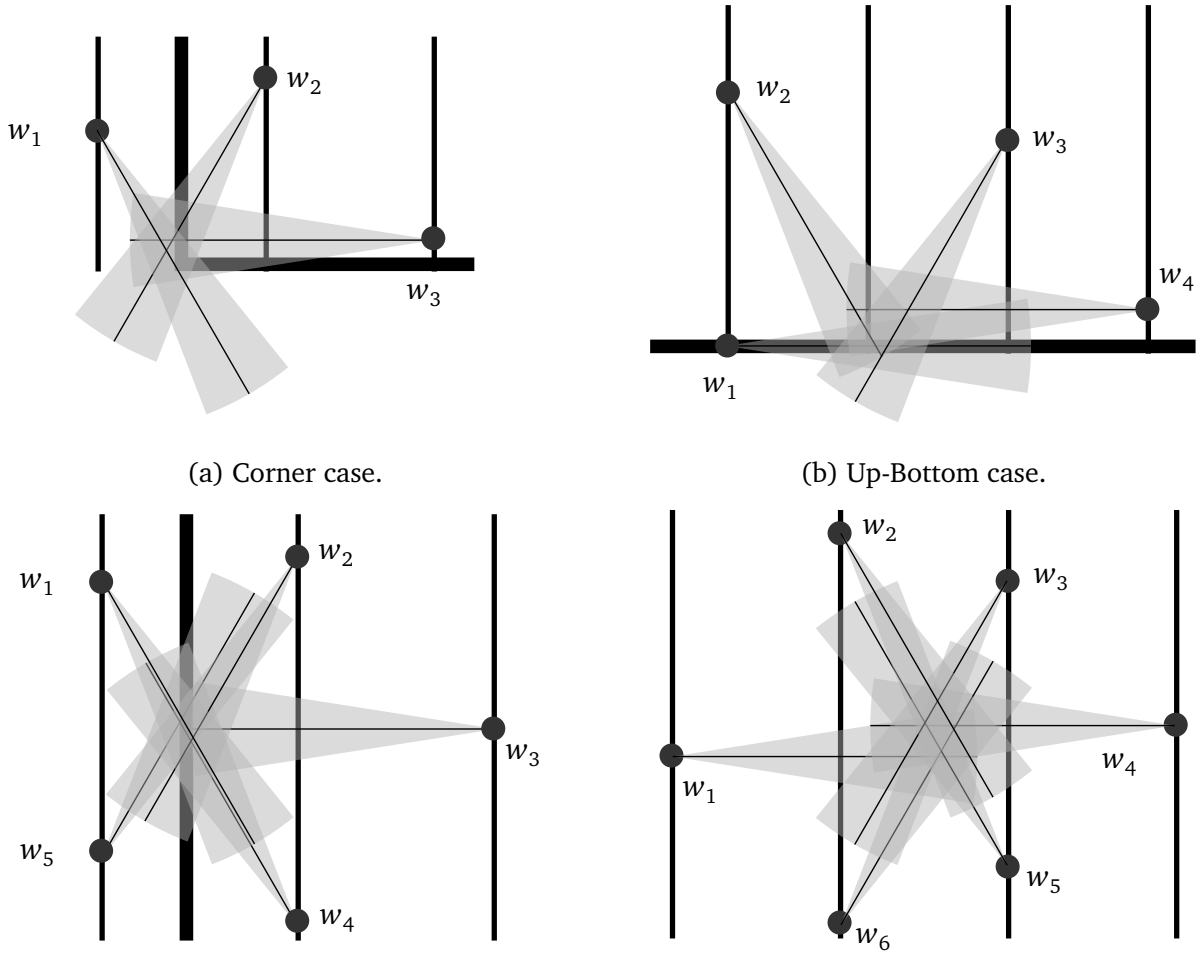


Figure 7.7: Ranging cases summary.

CASE 1: A node $P = (x_P, y_P)$ placed near the bottom-left corner of Q, with $x_P < d_{\min}/2$ and $y_P < \sqrt{3}d_{\min}/2$, can be ranged only by the three waypoints, as depicted in Fig. 7.8a, i.e., dw^+, hor^- and up^-. Since $F_x = d_{\min}/2$, node P cannot not be ranged from a horizontal beam hor^+ along the first scan because the $d_{\min}$ constraint is not satisfied. Hence the sensor has to wait until the drone passes through w_3 for collecting the hor type of orientation. A similar reasoning holds for the other areas labeled 1 in Fig. 7.7.

CASE 2: A node $P = (x_P, y_P)$ placed near the bottom border of the area (except corners), with $x_P \geq d_{\min}/2$ and $y_P < \sqrt{3}d_{\min}/2$, can be ranged by four waypoints, as shown in Fig. 7.8b, i.e., dw^+, $hor^{+/-}$ and up^-. For the drone it is impossible to reach P from up^+ or dw^+. Nevertheless, as can be seem in Fig. 7.8b, the sensor can be always reached by the three different types of orientations and thus it can collect sufficient measurements for localizing itself. A similar reasoning can be repeated for the other areas labeled 2 in Fig. 7.7.

CASE 3: A node P placed in the area of Q labeled 3 near the left border of the area (except corners), with $dw^{+/-}$, hor^- and $up^{+/-}$, can be ranged up to five waypoints, as shown in Fig. 7.8c. The sensor will be localized as soon as it has collected three different types of orientations. Like the first case, to have a measure at distance $d_{\min}$ in the hor type of orientation, the sensor has to



(a) Corner case.

(b) Up-Bottom case.

(c) Left-right case.

(d) General case.

Figure 7.8: Ranging cases.

wait until the drone passes through w_3.

CASE 4: Finally, a node P placed in the middle of Q, in the zone labeled 4, could be ranged by six waypoints, as shown in Fig. 7.8d, i.e., $dw^{+/-}$, $hor^{+/-}$ and $up^{+/-}$. Since after measurements from the three different types of orientations have been collected, the trilateration procedure starts, after three measurements in the best case, and, after five measurements, in the worst case.

From the analysis of the Cases 1, 2, 3 and 4 and from the previous discussion on stripe Z, the correctness of Theorem 6 follows. □

Note that Theorem 6 implies that the beamwidth 2θ increases with I_w. Thus if we decrease the number of waypoints, although the energy consumption due to hovering time of the drone is reduced, the transmission power increases because a larger beamwidth has to be used. Hence the selection of I_w is crucial for many aspects (energy, precision) of our algorithm.

As explained, the localization error strongly depends on the position of the three waypoints from which the point is ranged. From Eq. (3.9), the error in the position of the sensor is minimum when for each sensor $\beta_{\min} = \pi/3$.

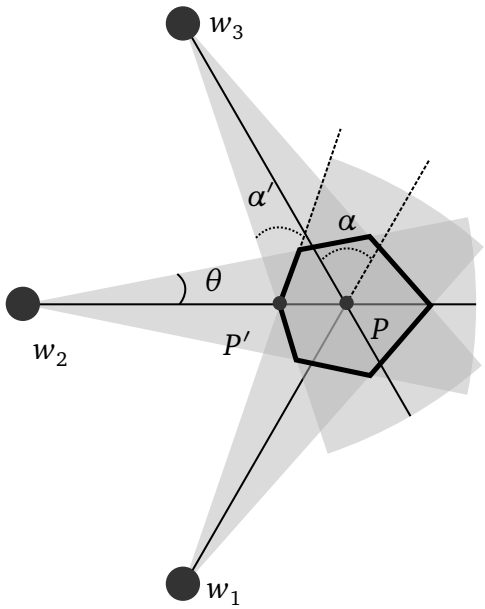


Figure 7.9: The polygon measured by up^+ in w_1, hor^+ in w_2, and dw^+ in w_3.

In our localization technique, the minimum angle at P is $\pi/3$ if and only if the node P resides at the intersection of the orientations of the three sectors centered at the waypoints $w_1(P)$, $w_2(P)$ and $w_3(P)$ (see Fig. 7.9). Although it is not possible to achieve for every point P that the minimum angle is $\pi/3$, we can claim the following result:

Theorem 7. *Fixed the half beamwidth θ according to Theorem 6, our localization technique with directional antennas guarantees:*

$$\beta_{min} \geq \frac{\pi}{3} - 2\theta$$

Proof. Consider a point P along with its three ranging waypoints w_1, w_2 and w_3 which belongs to two scans, say S_1 and S_2. W.l.o.g., let w_1 be the center of the sector dw^+, w_2 the center of the sector up^+ and w_3 the center of hor^- (see Fig. 7.10).

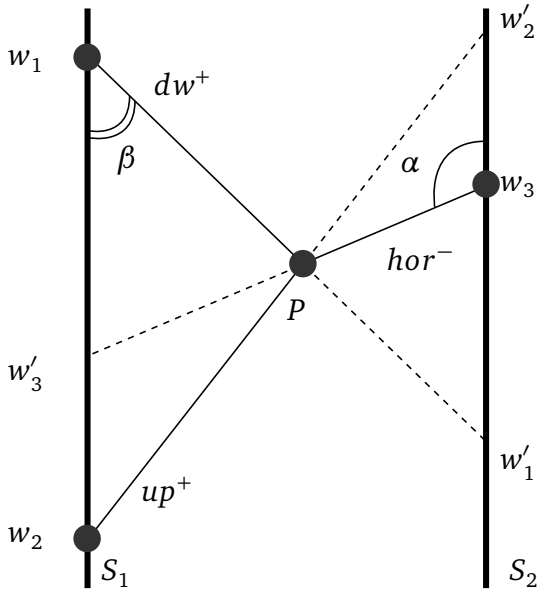


Figure 7.10: Relation between $\beta_{\min}$ and θ.

Let w'_i for $i = 1, 2, 3$ be at the intersection of the extensions of the line w_iP and of the opposite scan. The turn angle at P is divided in six angles by the lines w_iP and their extensions. The width of the angles at P depends on the angles formed by the lines w_iP or their extensions w'_iP and the scans. Since the line w_iP, for $i = 1, 2, 3$, belongs to the sector of w_i, the angles formed at the scans yield:

$$\frac{\pi}{2} - \theta \leq \alpha \leq \frac{\pi}{2} + \theta$$

if w_i is the center of a sector of type hor, and

$$\frac{\pi}{6} - \theta \leq \beta \leq \frac{\pi}{6} + \theta$$

if w_i is the center of a sector of type up/dw. By simple geometric considerations, the minimum angle at P is no smaller than:

$$\left\{\pi - \frac{\pi}{2} - \frac{\pi}{6} - 2\theta, 2\frac{\pi}{6} - 2\theta\right\} = \frac{\pi}{3} - 2\theta$$

Hence, $\beta_{\min} \geq \frac{\pi}{3} - 2\theta$. □

Fig. 7.11 depicts the heatmap of minimum angle referred to the example shown in Fig. 7.9. The polygon represents the intersection of the three circular sectors, that is the points of Q that are measured by w_1, w_2 and w_3. For each point P, a light gray color means a good minimum angle at P while a dark gray means a bad angle. The worst point is the leftmost point P' of polygon whose $\beta_{\min} = \alpha$ in Fig. 7.9. Note that the lines w_3P' and w_1P' form with the scan that contains w_1 and w_3 angles of width $\pi/6-\theta$, and thus $\alpha = 2\pi/6-2\theta$. In the example, $\theta = \arctan\frac{I_w/2}{d_{\min}} = 10$ degrees and the heatmap registers angles greater than $60-20 = 40$ degrees, according to Theorem 7.

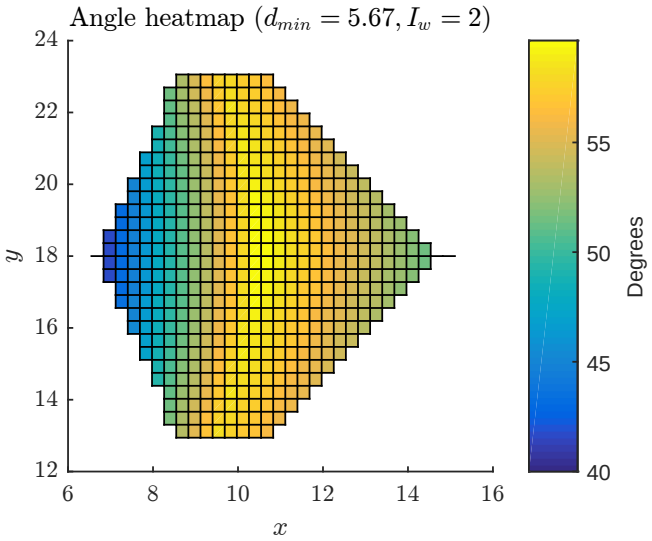


Figure 7.11: Minimum angle expressed (in degrees).

We are now in a position to rewrite Eq. (3.9) as a function that depends on h, $d_{\min}$ and I_w and state our main result:

Theorem 8. *The localization precision can be expressed as:*

$$\epsilon_L = \epsilon_s \cdot \frac{\sqrt{1+\frac{h^2}{d_{min}{}^2}} \cdot 2\sqrt{1+\left(\frac{I_w/2}{d_{min}}\right)^2}}{1-\sqrt{3}\left(\frac{I_w/2}{d_{min}}\right)} \tag{7.2}$$

Proof. By Theorem 7, $\frac{\beta_{\min}}{2} \geq \frac{\pi}{6}-\theta$. The largest error occurs when the angle $\frac{\beta_{\min}}{2}$ assumes the minimum value, i.e., $\frac{\pi}{6}-\theta$. Recalling that $\theta = \arctan\left(\frac{I_w/2}{d_{\min}}\right)$, calculating $\sin(\frac{\pi}{6}-\theta)$ and substituting in Eq. (3.9), it holds:

$$\epsilon_L = \epsilon_s \cdot \frac{\sqrt{1+\frac{h^2}{d_{\min}{}^2}}}{\sin(\frac{\pi}{6}-\theta)} = \epsilon_s \cdot \frac{\sqrt{1+\frac{h^2}{d_{\min}{}^2}} \cdot 2\sqrt{1+\left(\frac{I_w/2}{d_{\min}}\right)^2}}{1-\sqrt{3}\left(\frac{I_w/2}{d_{\min}}\right)}$$

□

To achieve the user-required localization precision ϵ_L, fixed h and r, we can invert Eq. (7.2) to find pairs of values $d_{\min}$, I_w that guarantee ϵ_L. Tab. 7.2 gives $d_{\min}$ values for different input values of ϵ_L fixed $I_w = \{2, 5, 10\}$, $h = \{15, 30\}$ and $r = 150$.

In conclusion, given h and the communication range r, the value of I_w can be selected so that the drone's mission is uniquely defined and ensures the localization precision of Eq. (7.2).

		ϵ_L(m)		
h(m)	I_w(m)	0.3	0.6	0.9
15	2	16.75	7.35	5.33
	5	22.43	10.62	8.31
	10	33.25	16.41	13.43
30	2	30.06	12.61	8.72
	5	35.28	15.77	11.64
	10	44.86	21.25	16.62

Table 7.2: The values $d_{\min}$ for DIR.

7.5 Experimental Evaluation

We have implemented the new DIR localization algorithm in MATLAB. We simulate a localization mission by deploying at random n nodes, with $n = 50, 100, 250$ and 500, on a 500×500 m^2 map (at most one sensor in every square of size 25 m). In all experiments, we set $r = 150$ m which is the average communication range of the IR-UWB DecaWave antenna [92]. Moreover, we use as the maximum altitude $h = 60$ m in order to follow the European Drone Regulations for civil drones [145].

We simulate the measurement in DIR and OMNI as follows. Since the drone is simulated, we only know the exact ground distance between the waypoint and the sensor. We can calculate using the altitude the exact slant distance. Thus, to simulate the slant measure we proceed as follows: fixed the slant precision $\epsilon_s = 0.1$ m, as claimed by DecaWave IR-UWB [92], we compute the slant distance s measured by the drone by adding a random generated slant error $s_{\text{err}} \in [-\epsilon_s, \epsilon_s]$ to the exact slant distance node-drone. Then we compute the ground measure d as: $\sqrt{s^2 - h^2}$. Note that d at this point includes the ground error.

We compare the performance of DIR with OMNI [115] and of RANGE-FREE (briefly, RF) [29, 146]. Recall the OMNI algorithm in [115] tessellates the area with diamonds. Each diamond is

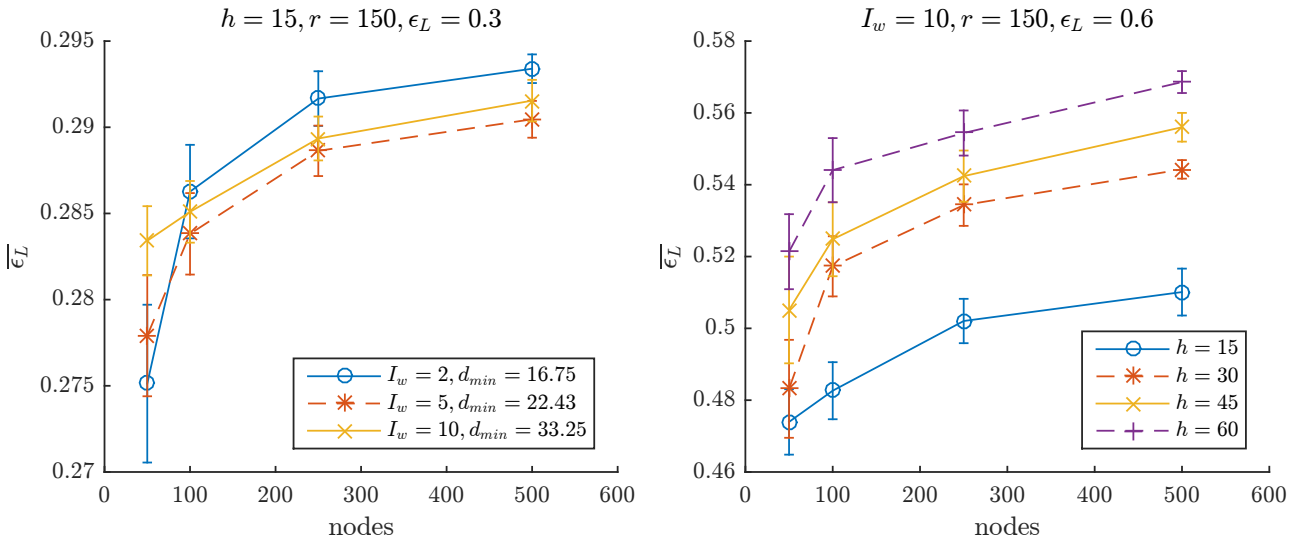
associated with an optimal triple of waypoints. So the sensors first compute their rough position $\hat{P}$ using the first three beacons that can receive. From $\hat{P}$, each sensor finds the diamond where it resides and repeats the trilateration using the optimal triple associated with a such diamond. Also OMNI guarantees a user-required precision but the expression of ϵ_L in terms of $d_{\min}$, I_w is:

$$\epsilon_L = \epsilon_s \frac{\sqrt{\left(1+\frac{h^2}{d_{\min}{}^2}\right)(1+\tan^2(\gamma))}}{\tan(\gamma)} \tag{7.3}$$

where

$$\tan(\gamma) = \frac{\frac{d_{\min}}{2}}{\sqrt{3}\frac{d_{\min}}{2} + 2I_w}. \tag{7.4}$$

For $\epsilon_L = \{0.3, 0.6, 0.9\}$, we derive the values of $d_{\min}$ for DIR using Eq. (7.2) (Tab. 7.2) and for OMNI using Eq. (7.3) (Tab. 4.1). It is interesting to note that the threshold $d_{\min}$ is always smaller in DIR than in OMNI because the angle in $\beta_{\min}$ constraint is larger in DIR than in OMNI.



(a) $\overline{\epsilon_L} = 0.3$ and I_w varies.

(b) $\overline{\epsilon_L} = 0.6$ and h varies.

Figure 7.12: The error bound $\overline{\epsilon_L}$ in Algorithm DIR (meters).

We generate 35 different sensor deployments. For each deployment, we simulate the localization mission. In order to evaluate the localization precision of each simulated mission, we evaluate the *experimental position precision* $\epsilon_L(P)$ for each node P by comparing the real position of P with the estimated position computed according to Eq. (3.8). For each mission M, we record the *worst experimental localization precision*, that is, $\epsilon_L(M) = \max_{P \in Q}\{\epsilon_L(P)\}$. Then, we compute the *error bound* $\overline{\epsilon_L} = \sum_{i=1}^{35} \epsilon_L(M_i)/35$ as the *average* value of the worst bound on the position error in all the missions. We also compute the 95%-confidence interval of the data reported in the experiments. In our experiments we are interested in comparing ϵ_L with $\overline{\epsilon_L}$.

Fig. 7.12 plots the error bound $\overline{\epsilon_L}$ of the DIR algorithm. The error bound in Fig. 7.12 is always smaller than the user-defined localization precision. Not only is $\overline{\epsilon_L} \leq \epsilon_L$, but also the maximum error in each mission is below the user-defined localization precision, i.e., $\epsilon_L(M) \leq \epsilon_L$.

In Fig. 7.12a, we can also see that $\overline{\epsilon_L}$ in general becomes worse when $d_{\min}$ increases and I_w increases. Hence for the precision, it is better to have smaller values of I_w. However, smaller values of I_w increase the number of waypoints, and hence the energy consumption due to the extra measurements and the possible hovering mode. Finally, Fig. 7.12b plots the precision $\overline{\epsilon_L}$ for $r = 150\,\text{m}$ at different altitudes when the user-defined localization precision is $\epsilon_L = 0.6$. According to Eq. (3.1), the localization error increases with the altitude. Nonetheless, it ensures the required precision.

Note that in our experiments, the values of I_w and $d_{\min}$ are set in order to achieve a pre-established precision.

Fig. 7.13 provides a comparison between the precision of OMNI and DIR. The a-priori localization precision ϵ_L is ensured with both the algorithms. The OMNI algorithm is a little bit more precise than DIR, probably because OMNI uses two position estimations. However, one can note that the precision of OMNI algorithm strongly depends on the quality of the first position estimation. If the first estimation of the position is so poor that it computes a wrong diamond –for example in an adverse context– the second estimation can only be poor. To validate this observation, in Fig. 7.14, we report the error bound $\overline{\epsilon_L}$ when the error of the first trilateration is accentuated up to $\hat{P}_{\text{err}}$. When $\hat{P}_{\text{err}} = 3I_w$, it is most likely that the first estimation returns a wrong diamond. In this case, the localization error $\overline{\epsilon_L}$ blows up significantly.

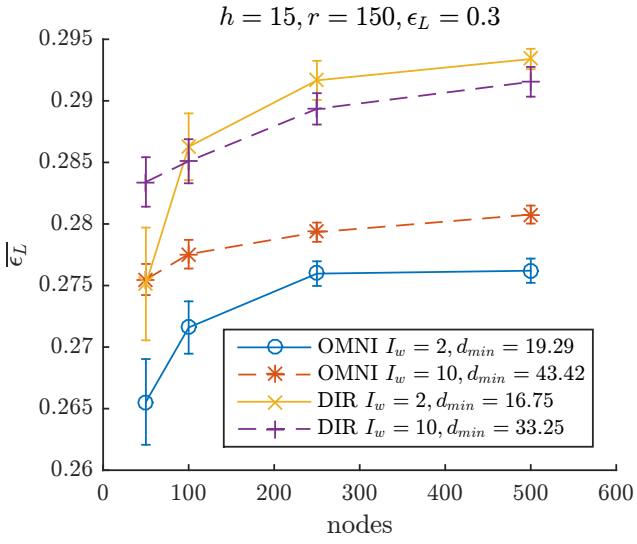


Figure 7.13: Comparing the error bound $\overline{\epsilon_L}$ (m) for OMNI and DIR when $\epsilon_L = 0.3$.

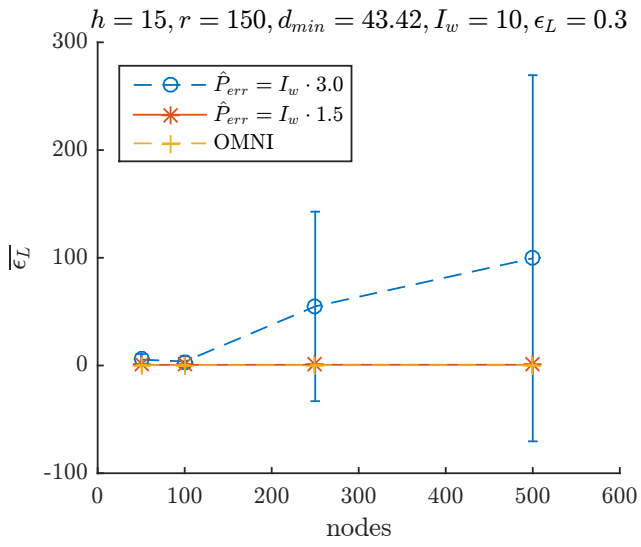


Figure 7.14: The error bound $\overline{\epsilon_L}$ (m) for OMNI when the first estimation $-\hat{P}_{\text{err}} \leq \hat{P} \leq \hat{P}_{\text{err}}$.

Fig. 7.15 compares the length of the static path of algorithms OMNI and DIR. They have more or less the same number of vertical scans, but each vertical scan in OMNI goes beyond the deployment area. Precisely, in OMNI, each vertical scan has length $2F_y + Q_y$, where $F_y = (d_{\max} - I_w)\sqrt{3}/2$, whereas the length is Q_y for DIR. Fig. 7.15 shows that the static path of DIR is better than that of OMNI. We reduce the path length from 8.2 km ($I_w = 2, d_{\min} = 19.29$) and 11.3 km ($I_w = 10, d_{\min} = 43.42$) in OMNI to 6.0 km ($I_w = 2, d_{\min} = 16.75$) and 7.7 km ($I_w = 10, d_{\min} = 33.25$) in DIR. The gain in the path length of DIR versus OMNI is above 30%. The gain is higher when $d_{\max} = \sqrt{r^2 - h^2}$ assumes large values. Note that the path length in OMNI is not monotonic decreasing because we use a fixed inter-scan value without resizing it as we do in DIR.

Finally, to compare our range-based algorithm with a range-free localization algorithm, we extend [29] by replacing the mobile anchor with a drone. In the RF algorithm [29], the drone follows a random path. Each sensor listens to the drone until it detects two chords, and localizes itself at the intersection of the two lines perpendicular at the middle point of such chords. Implementing the RF algorithm we set as in [29]: 1. the minimum chord length $r = 150$ m, 2. the minimum angle between chords 15 degrees, and 3. the beacon interval $b_i = \{1\,\text{m}, 2\,\text{m}\}$.

Since the RF algorithm was not designed to guarantee any error precision and since RF has been designed for a mobile anchor that moves on the ground and not for a drone, we cannot compare the precision of RF with that of OMNI and DIR. We just report that localization error that we experimentally found when the parameters (minimum chord, minimum angle, beacon interval) of RF are set as in [29]. In such a case, RF achieves a localization error of approximately 1 m.

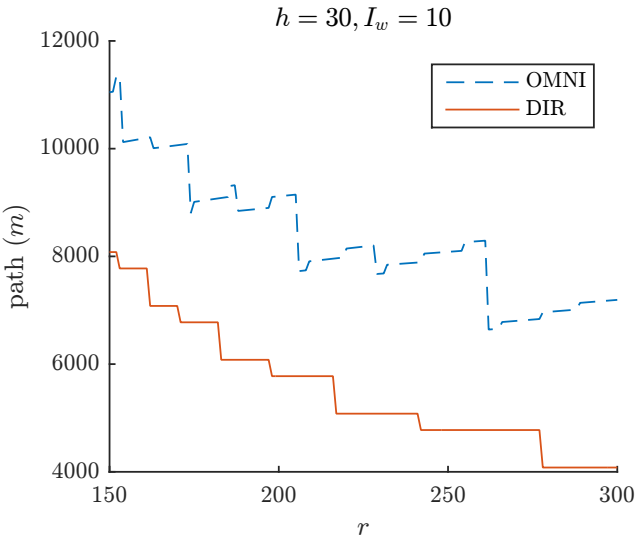


Figure 7.15: Path length in OMNI and DIR algorithms (m).

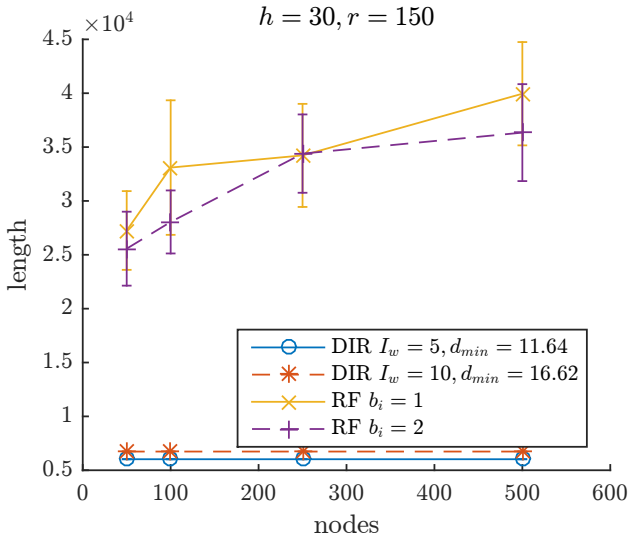


Figure 7.16: The path length of DIR and RF when experimentally $\epsilon_L = 1\,\text{m}$.

We have compared in Fig. 7.16 the length of the static path of DIR and RF assuming to know the total number of nodes to be localized. The path traversed by the drone while executing the RF algorithm is four times or more that of DIR and as such it cannot be traversed with a single battery charge of a customary drone. Moreover, observe that the length of the path of DIR does not depend on the number n of nodes, while the length of RF increases with n (see Fig. 7.16). Moreover, as already discussed, in general RF does not know when to stop the drone if no assumptions are done on the overall number of nodes and on the number of localized nodes. However, one has to recall that RF is a range-free algorithm and sensors do not require any special hardware to perform the measurements.

Conclusion In this work, we presented a localization technique that replaces multiple fixed anchors nodes with a flying drone using directional antennas. We have shown that our proposed algorithm DIR is able to guarantee the localization precision required by the user and it provides a shorter path than the existing algorithms OMNI and RF. The new algorithm is slightly less precise that OMNI, but it requires a single trilateration.

Conclusions

In this thesis we have seen that the combined utilization of two technologies, i.e., drones and directional antennas, led us to obtain satisfactory results on the field of localization. In order to achieve these results, we started from the analysis of measurement errors affected by a flying anchor like a drone, and we learned that the localization error can be bounded just following appropriate rules (Chapter 3). Then, we conduced a campaign of localizations using a drone equipped with an omnidirectional antenna, showing that the user-defined localization precision can be obtained just playing with a single parameter, i.e., the minimal ground distance (Chapter 4). After this, we focused on secure localization, and we noted that there is a price to pay in order to obtain a certain level of security, in terms of time and path length (Chapter 5). Finally, we studied and simplified the model of a directional antenna from a real complex fading model to an ideal simple model (Chapter 6). These results guided us to run another campaign of localizations using an array of directional antennas. We proved that using directional antennas we obtain a lighter localization algorithm (i.e., each node is localized just once), without loosing the ability of preserving the precision required by the user in the localization (Chapter 7).

In conclusion, this thesis gives me a solid base to continue my research in this challenging area. In particular, as open future problems, I wish to investigate the density of the network that can be supported by our OMNI and DIR algorithms when the drone takes the measurements while flying. I will also plan to evaluate our solutions when the drone ranges the nodes while hovering. In this case, the terrestrial nodes' density becomes less critical, but energy may become an issue.

I wish to be in a position to acquire more antennas and thus to complete the theoretical analysis with the experiments in a slightly larger test-bed.

Recent works exploit the usage of a swarm of drones with the purpose of sensor localization. Considering that an off-the-shelf drone has a limited battery life, it is crucial to rely to multiple vehicles if the sensing area starts to be large enough. Anyway, in the immediate future, drones will change our normal daily life.

Publications

- Amitabha Bagchi, Francesco Betti Sorbelli, Cristina Maria Pinotti, and Vinay Ribeiro. Connectivity of a dense mesh of randomly oriented directional antennas under a realistic fading model. In *LNCS*, volume 9536, pages 13–26, 2015

- Cristina M Pinotti, Francesco Betti Sorbelli, Pericle Perazzo, and Gianluca Dini. Localization with guaranteed bound on the position error using a drone. In *Proceedings of the 14th ACM International Symposium on Mobility Management and Wireless Access*, pages 147–154. ACM, 2016

- Pericle Perazzo, Francesco Betti Sorbelli, Mauro Conti, Gianluca Dini, and Cristina M Pinotti. Drone path planning for secure positioning and secure position verification. *IEEE Transactions on Mobile Computing*, 16(9):2478–2493, 2017

- Francesco Betti Sorbelli, Sajal K. Das, Cristina M. Pinotti, and Simone Silvestri. Precise localization in sparse sensor networks using a drone with directional antennas. In *Proceedings of the 19th International Conference on Distributed Computing and Networking, ICDCN*, 2018

- Francesco Betti Sorbelli, Sajal K. Das, Cristina M. Pinotti, and Simone Silvestri. On the accuracy of localizing terrestrial objects using drones. In *IEEE International Conference on Communications, ICC 2018*, 2018. Accepted for publication

Bibliography

[1] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[2] Victor V Zhirnov and Daniel JC Herr. New frontiers: Self-assembly and nanoelectronics. *Computer*, 34(1):34–43, 2001.

[3] Brett Warneke, Matt Last, Brian Liebowitz, and Kristofer SJ Pister. Smart dust: Communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, 2001.

[4] Stephan Olariu, Ashraf Wada, Larry Wilson, and Mohamed Eltoweissy. Wireless sensor networks: leveraging the virtual infrastructure. *IEEE network*, 18(4):51–56, 2004.

[5] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, and Gregory J Pottie. Protocols for self-organization of a wireless sensor network. *IEEE personal communications*, 7(5):16–27, 2000.

[6] Culler David, E Deborah, and Srivastva Mani. Overview of sensor networks. *Computer*, 32:41–50, 2004.

[7] Bruce Hemingway, Waylon Brunette, Tom Anderl, and Gaetano Borriello. The flock. *Computer*, 37(8):72–78, 2004.

[8] Kimiko Ryokai and Justine Cassell. Storymat: a play space for collaborative storytelling. In *CHI'99 extended abstracts on Human factors in computing systems*, pages 272–273. ACM, 1999.

[9] Joseph Polastre, Robert Szewczyk, Alan Mainwaring, David Culler, and John Anderson. Analysis of wireless sensor networks for habitat monitoring. In *Wireless sensor networks*, pages 399–423. Springer, 2004.

[10] Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler. An analysis of a large scale habitat monitoring application. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 214–226. ACM, 2004.

[11] Kirk Martinez, Jane K Hart, Royan Ong, S Brennan, A Mielke, D Torney, A Maccabe, M Maroti, G Simon, A Ledeczi, et al. Sensor network applications. *IEEE computer*, 37(8):50–56, 2004.

[12] Robert Szewczyk, Eric Osterweil, Joseph Polastre, Michael Hamilton, Alan Mainwaring, and Deborah Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, 2004.

[13] Kevin A Delin and Shannon P Jackson. The sensor web: a new instrument concept. In *Proc. SPIE*, volume 4282, 2001.

[14] David M Doolin and Nicholas Sitar. Wireless sensors for wildfire monitoring. In *Proc. SPIE*, volume 5765, pages 477–484, 2005.

[15] Mani Srivastava, Richard Muntz, and Miodrag Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 132–138. ACM, 2001.

[16] Ferruccio Barsi, Alan A Bertossi, Francesco Betti Sorbelli, Roberto Ciotti, Stephan Olariu, and Cristina M Pinotti. Asynchronous training in wireless sensor networks. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 46–57. Springer, 2007.

[17] Ferruccio Barsi, Francesco Betti Sorbelli, Roberto Ciotti, Cristina M Pinotti, Alan A Bertossi, and Stephan Olariu. Asynchronous training in sanet. In *Proceedings of the First ACM workshop on Sensor and actor networks*, pages 43–50. ACM, 2007.

[18] Francesco Betti Sorbelli, Roberto Ciotti, Alfredo Navarra, Maria Cristina Pinotti, and Vlady Ravelomanana. Cooperative training in wireless sensor and actor networks. In *QSHINE*, volume 9, pages 569–583. Springer, 2009.

[19] Ferruccio Barsi, Alan A Bertossi, Francesco Betti Sorbelli, Roberto Ciotti, Stephan Olariu, and M Cristina Pinotti. Asynchronous corona training protocols in wireless sensor and actor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(8):1216–1230, 2009.

[20] Alfredo Navarra, CM Pinotti, Vlady Ravelomanana, F Betti Sorbelli, and Roberto Ciotti. Cooperative training for high density sensor and actor networks. *IEEE Journal on Selected Areas in Communications*, 28(5):753–763, 2010.

[21] Vijay K Chaurasiya, Neeraj Jain, and Gora Chand Nandi. A novel distance estimation approach for 3d localization in wireless sensor network using multi dimensional scaling. *Information Fusion*, 15:5–18, 2014.

[22] Hua-Jie Shao, Xiao-Ping Zhang, and Zhi Wang. Efficient closed-form algorithms for aoa based self-localization of sensor nodes using auxiliary variables. *IEEE Transactions on Signal Processing*, 62(10):2580–2594, 2014.

[23] Ferdews Tlili, Abderrezak Rachedi, and Abderrahim Benslimane. Time-bounded localization algorithm based on distributed multidimensional scaling for wireless sensor networks. In *Communications (ICC), 2014 IEEE International Conference on*, pages 233–238. IEEE, 2014.

[24] Yong Ding, Chen Wang, and Li Xiao. Using mobile beacons to locate sensors in obstructed environments. *Journal of Parallel and Distributed Computing*, 70(6):644–656, 2010.

[25] Sunho Lim, Chansu Yu, and Chita R Das. A realistic mobility model for wireless networks of scale-free node connectivity. *International Journal of Mobile Communications*, 8(3):351–369, 2010.

[26] Amotz Bar-Noy, Ilan Kessler, and Moshe Sidi. Mobile users: To update or not to update? *Wireless Networks*, 1(2):175–185, 1995.

[27] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile computing*, pages 153–181, 1996.

[28] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.

[29] Kuo-Feng Ssu, Chia-Ho Ou, and Hewijin Christine Jiau. Localization with mobile anchor points in wireless sensor networks. *IEEE transactions on Vehicular Technology*, 54(3):1187–1197, 2005.

[30] Gang Lu, Gordon Manson, and Demetrios Belis. Mobility modeling in mobile ad hoc networks with environment-aware. *Journal of Networks*, 1(1):54–63, 2006.

[31] Kezhong Liu and Ji Xiong. A fine-grained localization scheme using a mobile beacon node for wireless sensor networks. *Journal of Information Processing Systems*, 6(2):147–162, 2010.

[32] Liu Xiao, Wang Jin-kuan, and Wang Yun. A novel localization algorithm based on received signal strength for mobile wireless sensor networks. In *Microwave and Millimeter Wave Technology, 2008. ICMMT 2008. International Conference on*, volume 1, pages 92–95. IEEE, 2008.

[33] Zhengyong Zhang, Zhi Sun, Gang Wang, Rong Yu, and Sunliang MEI. Localization in wireless sensor networks with mobile anchor nodes [j]. *Journal of Tsinghua University (Science and Technology)*, 4:021, 2007.

[34] Adem Tuncer and Mehmet Yildirim. Dynamic path planning of mobile robots with improved genetic algorithm. *Computers & Electrical Engineering*, 38(6):1564–1572, 2012.

[35] Dimitrios Koutsonikolas, Saumitra M Das, and Y Charlie Hu. Path planning of mobile landmarks for localization in wireless sensor networks. *Computer Communications*, 30(13):2577–2592, 2007.

[36] Rui Huang and Gergely V Zaruba. Static path planning for mobile beacons to localize sensor networks. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*, pages 323–330. IEEE, 2007.

[37] Javad Rezazadeh, Marjan Moradi, Abdul Samad Ismail, and Eryk Dutkiewicz. Superior path planning mechanism for mobile beacon-assisted localization in wireless sensor networks. *IEEE Sensors Journal*, 14(9):3052–3064, 2014.

[38] Jinfang Jiang, Guangjie Han, Huihui Xu, Lei Shu, and Mohsen Guizani. Lmat: Localization with a mobile anchor node based on trilateration in wireless sensor networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6. IEEE, 2011.

[39] Chia-Ho Ou and Wei-Lun He. Path planning algorithm for mobile anchor-based localization in wireless sensor networks. *IEEE Sensors Journal*, 13(2):466–475, 2013.

[40] Kyunghwi Kim and Wonjun Lee. Mbal: A mobile beacon-assisted localization scheme for wireless sensor networks. In *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pages 57–62. IEEE, 2007.

[41] Naser Hossein Motlagh, Tarik Taleb, and Osama Arouk. Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives. *IEEE Internet of Things Journal*, 3(6):899–922, 2016.

[42] Barry J O'Brien, David G Baran, and Brian B Luu. Ad hoc networking for unmanned ground vehicles: Design and evaluation at command, control, communications, intelligence, surveillance and reconnaissance on-the-move. Technical report, ARMY RESEARCH LAB ADELPHI MD COMPUTATIONAL AND INFORMATION SCIENCES DIRECTORATE, 2006.

[43] Chad C Haddal and Jeremiah Gertler. Homeland security: Unmanned aerial vehicles and border surveillance. *LIBRARY OF CONGRESS WASHINGTON DC CONGRESSIONAL RESEARCH SERVICE*, 2010.

[44] Rachel L Finn and David Wright. Unmanned aircraft systems: Surveillance, ethics and privacy in civil applications. *Computer Law & Security Review*, 28(2):184–194, 2012.

[45] Anibal Ollero and L Merino. Unmanned aerial vehicles as tools for forest-fire fighting. *Forest Ecology and Management*, 234(1):S263, 2006.

[46] Jane K Hart and Kirk Martinez. Environmental sensor networks: A revolution in the earth system science? *Earth-Science Reviews*, 78(3):177–191, 2006.

[47] Seng W Loke. The internet of flying-things: Opportunities and challenges with airborne fog computing and mobile cloud in the clouds. *arXiv preprint arXiv:1507.04492*, 2015.

[48] YU Jie, Nachiappan Subramanian, Kun Ning, and David Edwards. Product delivery service provider selection and customer satisfaction in the era of internet of things: a chinese e-retailers' perspective. *International Journal of Production Economics*, 159:104–116, 2015.

[49] Anouck R Girard, Adam S Howell, and J Karl Hedrick. Border patrol and surveillance missions using multiple unmanned air vehicles. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 620–625. IEEE, 2004.

[50] Gur Kimchi, Daniel Buchmueller, Scott A Green, Brian C Beckman, Scott Isaacs, Amir Navot, Fabian Hensel, Avi Bar-Zeev, and Severan Sylvain Jean-Michel Rault. Unmanned aerial vehicle delivery system, February 21 2017. US Patent 9,573,684.

[51] Sung-Suk Choi and Eung-Kon Kim. Building crack inspection using small UAV. In *17th International IEEE Conference on Advanced Communication Technology (ICACT)*, pages 235–238, 2015.

[52] Vijay K. Shah, Shameek Bhattacharjee, Simone Silvestri, and Sajal K. Das. Designing sustainable smart connected communities using dynamic spectrum access via band selection. In *ACM BuildSys, Delft, the Netherlands*, 2017.

[53] Amateur Related Satellite Services Tech. Rep. M.2171 Mobile, Radio Determination. *Characteristics of unmanned aircraft systems and spectrum requirements to support their safe operation in nonsegregated airspace*, 2017 (accessed October 20, 2017). http://www.itu.int/pub/R-REP-M.2171-2009/fr.

[54] Roland Everett Weibel. *Safety considerations for operation of different classes of unmanned aerial vehicles in the national airspace system*. PhD thesis, Massachusetts Institute of Technology, 2005.

[55] Andrew D Zeitlin and Michael P McLaughlin. Safety of cooperative collision avoidance for unmanned aircraft. *IEEE Aerospace and Electronic Systems Magazine*, 22(4):9–13, 2007.

[56] Brian Capozzi and Juris Vagners. Evolving (semi)-autonomous vehicles. In *Proc. of the 2001 AIAA Guidance, Navigation and Control Conference*, 2001.

[57] Kamesh Namuduri, Yan Wan, and Mahadevan Gomathisankaran. Mobile ad hoc networks in the sky: State of the art, opportunities, and challenges. In *Proceedings of the second ACM MobiHoc workshop on Airborne networks and communications*, pages 25–28. ACM, 2013.

[58] C Witefeld. Communication-aware service platform for collaborative uav applications. *Talk at UAV-g, Commun. Networks Inst., Faculty Elect. Eng. Inform. Technol., TU Dortmond Univ., Dortmund, Germany*, 2013.

[59] Chuck Johnson, James H Griner, Kelly J Hayhurst, Robert J Shively, Maria Consiglio, Eric Muller, James Murphy, and Sam Kim. Unmanned aircraft systems (uas) integration in the national airspace system (nas) project subcommittee final. *Proceedings of the IEEE*, 2012.

[60] Imad Jawhar, Nader Mohamed, Jameela Al-Jaroodi, and Sheng Zhang. Data communication in linear wireless sensor networks using unmanned aerial vehicles. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 492–499. IEEE, 2013.

[61] Eric W Frew and Timothy X Brown. Airborne communication networks for small unmanned aircraft systems. *Proceedings of the IEEE*, 96(12), 2008.

[62] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. ACM, 2003.

[63] Kevin Fall, Keith L Scott, Scott C Burleigh, Leigh Torgerson, Adrian J Hooke, Howard S Weiss, Robert C Durst, and Vint Cerf. Delay-tolerant networking architecture. *Proceedings of the IEEE*, 2007.

[64] Jamal Nazzal Al-Karaki. Infrastructureless wireless networks: cluster-based architectures and protocols. 2004.

[65] Lav Gupta, Raj Jain, and Gabor Vaszkun. Survey of important issues in uav communication networks. *IEEE Communications Surveys & Tutorials*, 18(2):1123–1152, 2016.

[66] Jean-Daniel Medjo Me Biomo, Thomas Kunz, and Marc St-Hilaire. Routing in unmanned aerial ad hoc networks: A recovery strategy for greedy geographic forwarding failure. In *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, pages 2236–2241. IEEE, 2014.

[67] Rostam Shirani. *Reactive-greedy-reactive in unmanned aeronautical ad-hoc networks: A combinational routing mechanism*. PhD thesis, Carleton University, 2011.

[68] Ozgur Koray Sahingoz. Networking models in flying ad-hoc networks (fanets): Concepts and challenges. *Journal of Intelligent & Robotic Systems*, 74(1-2):513, 2014.

[69] Kuldeep Singh and Anil Kumar Verma. Applying olsr routing in fanets. In *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*, pages 1212–1215. IEEE, 2014.

[70] Md Hasan Tareque, Md Shohrab Hossain, and Mohammed Atiquzzaman. On the routing in flying ad hoc networks. In *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*, pages 1–9. IEEE, 2015.

[71] Shashank Kumar Singh. A comprehensive survey on fanet?: Challenges and advancements. *Int. J. Comput. Sci. Inf. Technol*, 6(3):2010–2013, 2015.

[72] Rodolfo Barros Chiaramonte and Kalinka Regina Lucas Jaquie Castelo Branco. Collision detection using received signal strength in fanets. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 1274–1283. IEEE, 2014.

[73] Samil Temel and Ilker Bekmezci. On the performance of flying ad hoc networks (fanets) utilizing near space high altitude platforms (haps). In *Recent Advances in Space Technologies (RAST), 2013 6th International Conference on*, pages 461–465. IEEE, 2013.

[74] Ilker Bekmezci, Ismail Sen, and Ercan Erkalkan. Flying ad hoc networks (fanet) test bed implementation. In *Recent Advances in Space Technologies (RAST), 2015 7th International Conference on*, pages 665–668. IEEE, 2015.

[75] Ovidiu Vermesan and Peter Friess. *Internet of things-from research and innovation to market deployment*, volume 29. River Publishers Aalborg, 2014.

[76] Courtney S Sharp, Omid Shakernia, and S Shankar Sastry. A vision system for landing an unmanned aerial vehicle. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1720–1727. Ieee, 2001.

[77] Pedro J Garcia-Pardo, Gaurav S Sukhatme, and James F Montgomery. Towards vision-based safe landing for an autonomous helicopter. *Robotics and Autonomous Systems*, 38(1):19–29, 2002.

[78] Stefan Hrabar and Gaurav S Sukhatme. Omnidirectional vision for an autonomous helicopter. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 558–563. IEEE, 2003.

[79] Srikanth Saripalli, James F Montgomery, and Gaurav S Sukhatme. Vision-based autonomous landing of an unmanned aerial vehicle. In *Robotics and automation, 2002. Proceedings. ICRA'02. IEEE international conference on*, volume 3, pages 2799–2804. IEEE, 2002.

[80] Benjamin Coifman, Mark McCord, M Mishalani, and Keith Redmill. Surface transportation surveillance from unmanned aerial vehicles. In *Proc. of the 83rd Annual Meeting of the Transportation Research Board*, 2004.

[81] J David Irwin. *The industrial electronics handbook*. CRC Press, 1997.

[82] Jacob Fraden. *Handbook of modern sensors: physics, designs, and applications*. Springer Science & Business Media, 2004.

[83] Byungjin Lee, Jongmin Lee, Young Jae Lee, and Sangkyung Sung. Development of cellular data network enabled autonomous rotary uav. In *Control, Automation and Systems (ICCAS), 2013 13th International Conference on*, pages 719–722. IEEE, 2013.

[84] Sameer Qazi, Ali Shuja Siddiqui, and Asim Imdad Wagan. Uav based real time video surveillance over 4g lte. In *Open Source Systems & Technologies (ICOSST), 2015 International Conference on*, pages 141–145. IEEE, 2015.

[85] Dusit Niyato and Ekram Hossain. Wireless broadband access: Wimax and beyond-integration of wimax and wifi: Optimal pricing for bandwidth sharing. *IEEE communications Magazine*, 45(5), 2007.

[86] Bjoern Dusza and Christian Wietfeld. Performance evaluation of ieee 802.16 e mobile wimax for long distance control of uav swarms. In *Wireless Information Technology and Systems (ICWITS), 2010 IEEE International Conference on*, pages 1–4. IEEE, 2010.

[87] Harald Skinnemoen. Uav & satellite communications live mission-critical visual data. In *Aerospace Electronics and Remote Sensing Technology (ICARES), 2014 IEEE International Conference on*, pages 12–19. IEEE, 2014.

[88] Dawei Ma and Shizhong Yang. Uav image transmission system based on satellite relay. In *Microwave and Millimeter Wave Technology, 2004. ICMMT 4th International Conference on, Proceedings*, pages 874–878. IEEE, 2004.

[89] HDIAC S. Fu and Y. Wan. *Spotlight: UAVs for disaster area communication*, 2017 (accessed October 20, 2017). https://www.hdiac.org/.

[90] José A Afonso, Ezequiel T Coelho, Paulo Carvalhal, Manuel João Oliveira Ferreira, Cristina Santos, Luís F Silva, and Heitor Almeida. Distributed sensing and actuation over bluetooth for unmanned air vehicles. *ICRA 2006*, 2006.

[91] Libei Yut, Qing Fei, and Qingbo Geng. Combining zigbee and inertial sensors for quadrotor uav indoor localization. In *Control and Automation (ICCA), 2013 10th IEEE International Conference on*, pages 1912–1916. IEEE, 2013.

[92] DecaWave. *DecaWave Ltd: ScenSor SWM1000 Module*, 2017 (accessed October 20, 2017). http://www.decawave.com/products/dwm1000-module.

[93] L Sakkila, C Tatkeu, F Boukour, Y El Hillali, A Rivenq, and JM Rouvean. Uwb radar system for road anti-collision application. In *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, pages 1–6. IEEE, 2008.

[94] Qiong Huang, Lele Qu, Bingheng Wu, and Guangyou Fang. Uwb through-wall imaging based on compressive sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 48(3):1408–1415, 2010.

[95] Cemin Zhang, Michael Kuhn, Brandon Merkl, Aly E Fathy, and Mohamed Mahfouz. Accurate uwb indoor localization system utilizing time difference of arrival approach. In *Radio and Wireless Symposium, 2006 IEEE*, pages 515–518. IEEE, 2006.

[96] Lawrence Carin, Norbert Geng, Mark McClure, Jeffrey Sichina, and Lam Nguyen. Ultra-wide-band synthetic-aperture radar for mine-field detection. *IEEE Antennas and Propagation Magazine*, 41(1):18–33, 1999.

[97] Tennessee Technological University. *Ultra-Wide-Band Research Poised to Save Lives in Rescue, Combat*, 2017 (accessed October 20, 2017). http://www.newswise.com/articles/view/512759?print-article.

[98] 3D Robotics. *Solo Specs: Just the facts*, 2017 (accessed October 20, 2017). https://news.3dr.com/solo-specs-just-the-facts.

[99] Official U.S. government. *GPS Accuracy*, 2017 (accessed October 20, 2017). https://www.gps.gov/systems/gps/performance/accuracy/.

[100] Pericle Perazzo, Francesco Betti Sorbelli, Mauro Conti, Gianluca Dini, and Cristina M Pinotti. Drone path planning for secure positioning and secure position verification. *IEEE Transactions on Mobile Computing*, 16(9):2478–2493, 2017.

[101] Yaser Al Mtawa, Nidal Nasser, and Hossam S Hassanein. Mitigating anchor misplacement errors in wireless sensor networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, pages 569–575. IEEE, 2015.

[102] Guillermo Heredia, Fernando Caballero, Iván Maza, Luis Merino, Antidio Viguria, and Aníbal Ollero. Multi-unmanned aerial vehicle (uav) cooperative fault detection employing differential global positioning (dgps), inertial and vision sensors. *Sensors*, 9(9):7566–7579, 2009.

[103] Jinfang Jiang, Guangjie Han, Chuan Zhu, Yuhui Dong, and Na Zhang. Secure localization in wireless sensor networks: a survey. *JCM*, 6(6):460–470, 2011.

[104] Alberto Compagno, Mauro Conti, Antonio Alberto D'Amico, Gianluca Dini, Pericle Perazzo, and Lorenzo Taponecco. Modeling enlargement attacks against uwb distance bounding protocols. *IEEE Transactions on Information Forensics and Security*, 11(7):1565–1577, 2016.

[105] Pericle Perazzo and Gianluca Dini. Secure positioning with non-ideal distance bounding protocols. In *Computers and Communication (ISCC), 2015 IEEE Symposium on*, pages 907–912. IEEE, 2015.

[106] Kasper Rasmussen, Mani Srivastava, et al. Secure location verification with hidden and mobile base stations. *IEEE Transactions on Mobile Computing*, 7(4):470–483, 2008.

[107] Srdjan Capkun and J-P Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(2):221–232, 2006.

[108] Yanchao Zhang, Wei Liu, Yuguang Fang, and Dapeng Wu. Secure localization and authentication in ultra-wideband sensor networks. *IEEE Journal on Selected areas in communications*, 24(4):829–835, 2006.

[109] Stefan Brands and David Chaum. Distance-bounding protocols. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 344–359. Springer, 1993.

[110] Pericle Perazzo, Lorenzo Taponecco, Antonio A D'amico, and Gianluca Dini. Secure positioning in wireless sensor networks through enlargement miscontrol detection. *ACM Transactions on Sensor Networks (TOSN)*, 12(4):27, 2016.

[111] Gianluca Dini, Francesco Giurlanda, and Pericle Perazzo. Secdev: Secure distance evaluation in wireless networks. In *Proc. 2013 International Conference on Networking and Services*, pages 207–212, 2013.

[112] Lorenzo Taponecco, Pericle Perazzo, Antonio A D'Amico, and Gianluca Dini. On the feasibility of overshadow enlargement attack on ieee 802.15. 4a distance bounding. *IEEE Communications Letters*, 18(2):257–260, 2014.

[113] Peter Corke, Ron Peterson, and Daniela Rus. Coordinating aerial robots and sensor networks for localization and navigation. In *Distributed Autonomous Robotic Systems 6*, pages 295–304. Springer, 2007.

[114] Mihail L Sichitiu and Vaidyanathan Ramadurai. Localization of wireless sensor networks with a mobile beacon. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 174–183. IEEE, 2004.

[115] Cristina M Pinotti, Francesco Betti Sorbelli, Pericle Perazzo, and Gianluca Dini. Localization with guaranteed bound on the position error using a drone. In *Proceedings of the 14th ACM International Symposium on Mobility Management and Wireless Access*, pages 147–154. ACM, 2016.

[116] Wenliang Du, Jing Deng, Yunghsiang S Han, Shigang Chen, and Pramod K Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *INFOCOM 2004. Twenty-third AnnualJoint conference of the IEEE computer and communications societies*, volume 1. IEEE, 2004.

[117] Pericle Perazzo, Kanishka Ariyapala, Mauro Conti, and Gianluca Dini. The verifier bee: A path planner for drone-based secure location verification. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*, pages 1–9. IEEE, 2015.

[118] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3):555–565, 1976.

[119] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.

[120] Christian Nilsson. Heuristics for the traveling salesman problem. *Linkoping University*, pages 1–6, 2003.

[121] Holger H Hoos and Thomas Stützle. On the empirical scaling of run-time for finding optimal solutions to the travelling salesman problem. *European Journal of Operational Research*, 238(1):87–94, 2014.

[122] David Applegate, William Cook, and André Rohe. Chained lin-kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 15(1):82–92, 2003.

[123] Kameswari Chebrolu and Bhaskaran Raman. Fractel: a fresh perspective on (rural) mesh networks. In *Proceedings of the 2007 workshop on Networked systems for developing regions*, page 8. ACM, 2007.

[124] Michael R Souryal, Andreas Wapf, and Nader Moayeri. Rapidly-deployable mesh network testbed. In *Proc. Global Telecommunications Conference (GLOBECOM 2009)*, pages 1–6. IEEE, 2009.

[125] Aniesh Chawla, Vinay Yadav, V Dev Sharma, Jitin Bajaj, Eshan Nanda, Vinay Ribeiro, and Huzur Saran. RODEO: Robust and rapidly deployable TDM mesh with QoS differentiation. In *Proc. 4th Intl. Conf. Communication Systems and Networks (COMSNETS 2012)*, pages 1–6. IEEE, 2012.

[126] Pan Li, Chi Zhang, and Yuguang Fang. Asymptotic connectivity in wireless ad hoc networks using directional antennas. *IEEE/ACM Trans. Netw.*, 17(4):1106–1117, 2009.

[127] Piyush Gupta and Panganamala R Kumar. Critical power for asymptotic connectivity in wireless networks. In *Stochastic analysis, control, optimization and applications*, pages 547–566. Springer, 1999.

[128] Amitabha Bagchi, Sainyam Galhotra, Tarun Mangla, and Cristina M Pinotti. Optimal radius for connectivity in duty-cycled wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 11(2):36, 2015.

[129] Tatiana K Madsen, Frank HP Fitzek, Ramjee Prasad, and Gerrit Schulte. Connectivity probability of wireless ad hoc networks: Definition, evaluation, comparison. *Wireless Pers. Commun*, 35(1-2):135–151, 2005.

[130] Paul Balister, Béla Bollobas, Amites Sarkar, and Santosh Kumar. Reliable density estimates for coverage and connectivity in thin strips of finite length. In *Proc. 13th Annual ACM Intl. Conf. on Mobile computing and networking (Mobicom 2007)*, pages 75–86. ACM, 2007.

[131] Olivier Dousse, François Baccelli, and Patrick Thiran. Impact of interferences on connectivity in ad hoc networks. *IEEE/ACM Trans. Netw*, 13(2):425–436, 2005.

[132] Peng-Jun Wan and Chih-Wei Yi. Asymptotic critical transmission ranges for connectivity in wireless ad hoc networks with bernoulli nodes. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC 2005)*, volume 4, pages 2219–2224. IEEE, 2005.

[133] Chih-Wei Yi, Peng-Jun Wan, Mo Li, and Ophir Frieder. Asymptotic distribution of the number of isolated nodes in wireless ad hoc networks with bernoulli nodes. *IEEE Trans. Commun.*, 54(3):510–517, 2006.

[134] Evangelos Kranakis, Danny Krizanc, and Eric Williams. Directional versus omnidirectional antennas for energy consumption and k-connectivity of networks of sensors. In *Proc. 8th Intl. Conf. Principles of Distributed Systems (OPODIS 2004)*, pages 357–368. Springer, 2004.

[135] Ioannis Caragiannis, Christos Kaklamanis, Evangelos Kranakis, Danny Krizanc, and Andreas Wiese. Communication in wireless networks with directional antennas. In *Proc. 20th Annual Symp. Parallelism in Algorithms and Architectures (SPAA 2008)*, pages 344–351. ACM, 2008.

[136] Paz Carmi, Matthew J Katz, Zvi Lotker, and Adi Rosén. Connectivity guarantees for wireless networks with directional antennas. *Comput. Geom.*, 44(9):477–485, 2011.

[137] Huiwen Xu, Hong-Ning Dai, and Qinglin Zhao. On the connectivity of wireless networks with multiple directional antennas. In *18th IEEE Intl. Conf. on Networks (ICON 2012)*, pages 155–160. IEEE, 2012.

[138] Zuoming Yu, Jin Teng, Xiaole Bai, Dong Xuan, and Weijia Jia. Connected coverage in wireless networks with directional antennas. *ACM T. Sensor Network*, 10(3):51, 2014.

[139] Christian Bettstetter, Christian Hartmann, and Clemens Moser. How does randomized beamforming improve the connectivity of ad hoc networks? In *IEEE Intl. Conf. on Communications (ICC 2005)*, volume 5, pages 3380–3385. IEEE, 2005.

[140] Constantine A Balanis. *Antenna theory: analysis and design*. John Wiley & Sons, 2012.

[141] Samuel Silver. *Microwave antenna theory and design*. Iet, 1949.

[142] Piyush Guptar and PR Kumar. Critical power for asymptotic connectivity. In *Proc. of the 37th IEEE, Conference on Decision & Control*, 1998.

[143] Hao-Tao Hu, Fu-Chang Chen, and Qing-Xin Chu. A compact directional slot antenna and its application in mimo array. *IEEE Transactions on Antennas and Propagation*, 64(12):5513–5517, 2016.

[144] Ken Goss, Riccardo Musmeci, and Simone Silvestri. Realistic models for characterizing the performance of unmanned aerial vehicles. In *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*, pages 1–9. IEEE, 2017.

[145] Enac. *UAV regulations in Italy*, 2017 (accessed October 20, 2017). https://goo.gl/vgktKd.

[146] Chia-Ho Ou and Kuo-Feng Ssu. Sensor position determination with flying anchors in three-dimensional wireless sensor networks. *IEEE Transactions on Mobile Computing*, 7(9):1084–1097, 2008.

[147] Amitabha Bagchi, Francesco Betti Sorbelli, Cristina Maria Pinotti, and Vinay Ribeiro. Connectivity of a dense mesh of randomly oriented directional antennas under a realistic fading model. In *LNCS*, volume 9536, pages 13–26, 2015.

[148] Francesco Betti Sorbelli, Sajal K. Das, Cristina M. Pinotti, and Simone Silvestri. Precise localization in sparse sensor networks using a drone with directional antennas. In *Proceedings of the 19th International Conference on Distributed Computing and Networking, ICDCN*, 2018.

[149] Francesco Betti Sorbelli, Sajal K. Das, Cristina M. Pinotti, and Simone Silvestri. On the accuracy of localizing terrestrial objects using drones. In *IEEE International Conference on Communications, ICC 2018*, 2018. Accepted for publication.