Wireless Sensor Network Planning with Application to UWB Localization in GPS-Denied Environments

by

Damien Bruno Jourdan

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2006

Author . . . .
Department of Aeronautics and Astronautics
May 22, 2006

Certified by . . . .
Moe Z. Win
Assistant Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by . . . .
Olivier L. de Weck
Assistant Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Advisor

Certified by . . . .
John J. Deyst, Jr.
Professor of Aeronautics and Astronautics
Thesis Advisor

Certified by . . . .
Nicholas Roy
Assistant Professor of Aeronautics and Astronautics
Thesis Advisor

Certified by . . . .
Eric Feron
Professor of Aerospace Engineering, Georgia Institute of Technology
Thesis Advisor

Accepted by . . . .
Jaime Peraire
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Wireless Sensor Network Planning with Application to UWB Localization in GPS-Denied Environments

by

Damien Bruno Jourdan

Submitted to the Department of Aeronautics and Astronautics
on May 22, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

Wireless Sensor Networks (WSN) have received much attention in the past 5 years, and much progress has been made in designing hardware, communications protocols, routing, and sensor fusion algorithms. The planning and deployment of the WSN, however, has been overlooked to a great extent. These are important aspects, as we show, which can result in significant gains in WSN performance and resource utilization.

We propose a comprehensive strategy for the planning, deployment, and operation of WSNs divided into 3 phases, Phases I, II, and III. This framework addresses the optimization challenges of the planning process, and takes into account the major sources of uncertainty (notably that due to the aerial deployment of the sensors), so that the WSN deployed on the ground performs as best as possible. We first present general-purpose algorithms implementing this strategy, and showcase their benefits on a few examples. In particular, a Multi-Objective Genetic Algorithm (MOGA) is proposed for the initial network planning of Phase I, and a greedy local search is used for the real-time deployment of Phase II.

We then direct our attention to a specific application, where a WSN is deployed to provide localization to an agent navigating in GPS-denied environments. The network relies on Ultra-Wideband (UWB) technology in order to provide accurate ranging. Because of its high resolution, UWB is able to provide high ranging accuracy, even in the kind of harsh environments typically found in GPS-denied areas (indoor, urban canyon, etc.). However the ranging accuracy is limited by two phenomena: the presence of positive biases in the range measurements due to NLOS propagation, and the increase of the measurement variance with distance. Given these characteristics, we derive the Position Error Bound (PEB), a lower bound on the localization accuracy of a sensor configuration.

We then develop a placement algorithm, RELOCATE, which places the sensors so as to minimize the PEB. We show that this algorithm can be used for the initial planning of Phase I, as well as when incremental planning is needed, such as during Phases II and III.

Finally a Monte Carlo localization (MCL) method is developed to fuse the range measurements from UWB sensors and the inertial measurements from an onboard IMU, so as to optimally estimate the state of an agent moving through a GPS-denied environment.

This algorithm is shown to provide high positioning accuracy, notably thanks to its ability to accurately estimate the bias embedded in the range measurements.

The benefits of using these smart algorithms is showcased at each step, in order to demonstrate the importance of optimally planning, deploying, and operating the WSN.

Thesis Supervisor: Moe Z. Win
Title: Assistant Professor of Aeronautics and Astronautics

Thesis Advisor: Olivier L. de Weck
Title: Assistant Professor of Aeronautics and Astronautics and Engineering Systems

Thesis Advisor: John J. Deyst, Jr.
Title: Professor of Aeronautics and Astronautics

Thesis Advisor: Nicholas Roy
Title: Assistant Professor of Aeronautics and Astronautics

Thesis Advisor: Eric Feron
Title: Professor of Aerospace Engineering, Georgia Institute of Technology

Acknowledgments

I would like to first thank my thesis committee. I could not have wished for a better, more supportive team of advisors who always had my best interest in mind. I'd like to thank Prof. Moe Win, the chair of my committee, for the opportunity to work with you and your great team of students, postdocs, and visiting professors. Your expert mix of research excellence and good humor made our collaboration most enjoyable. I especially have great memories from Zurich, with Mount Pilatus, lake Luzern, and Swiss-German food. I am also grateful for my collaboration with Prof. Nick Roy, which began in 2004 and ushered a very productive year, giving birth more or less to half of this thesis. Your support and encouragement were precious all along. Thanks also to Prof. Oli de Weck for your help with the first chapters of this thesis, and for helping me define this topic early on. I am grateful for Prof. Eric Feron encouraging me to make theoretical contributions. I enjoyed your good humor and our theological discussions, LIDS' hallways definitely are less lively since your departure. Last but not least, many, many thanks to Prof. Deyst. Your constant support, kindness and passion for our different projects have made these 6 years of working with you a pleasure.

I'd also like to thank the Draper Laboratory for generously funding me during my years at MIT. It gave me the unique opportunity to work on exciting projects, from UAVs to sensor networks. In particular I'd like to acknowledge Dr. Brent Appleby for his significant contribution in defining my thesis topic.

I greatly enjoyed collaborating with Prof. Davide Dardari from the University of Bologna, Italy. I learned a lot from you, and your rigor has forced me to strive more toward excellence. I also thank Dr. Chia-Chin Chong, from the DoCoMo Labs, for agreeing to being a thesis reader, along with Prof. Dardari.

Throughout my time at MIT I have been blessed with a great team of labmates. In my early PhD years, I enjoyed work and fun with Joshua Torgerson and Hyung-Il Anh. I am grateful for the friendship of Dr. Sanghyuk Park, my mentor when it came to research, academics, Korean culture and tango. Even if he is now in the cold lands of Quebec, I look forward to a life-long friendship. God bless you, Jiyoon and Adrian.

I am also grateful for Simon Nolet, and not only because he has been at MIT longer than I am and thus helps me feel still young. His expertise in Kalman filtering and estimation was of great help in my research. Finally my good friend “le” Anand, it has been such a blessing to be your labmate since our move to Stata. I am grateful that Eytan forced you to come to the office more often. It’s been a pleasure to get to know you better over the years. We shared good times, difficult “mid-PhD” crisis, and great discussions about anything. It’s one of those discussions that set the stage for what is now Chapter 6.

I’d be a fool to forget the many “non-research” persons who contributed to this thesis. Daniel, my dear friend, my fellow Frenchman–more than that, my brother in Jesus. I thank God for you, your prayers and encouragements during our Wednesday morning prayer meetings sustained me through the lows of MIT. It was great to go through much of the same experiences together: expatriation, the quals, PhD “crises,” losing our French (me to English, you to Quebecois), even marriage! Le Seigneur te bénisse, toi et Marie-Eve.

The Eastgate Bible study was my oasis every Tuesday evening. I enjoyed getting to know everyone, and God has taught me many truths through your insights. Through His Word and your prayers the Lord renewed my strength and kept me from getting discouraged. Special thanks to Bryan and David for keeping me accountable and challenging me to become a better husband.

Thank God for Park Street Church, where I was spiritually born and nurtured. Its excellent teaching and the passion and love of its staff and members have been a model for me in my 4 years as a Christian.

I’d like also to thank my family. First, forgive me for leaving you back in 2000 to come study here. Romain, it is hard to believe that when I left Paris you were only 13, and you are now 19. One of my greatest regret about coming to Boston is to miss seeing you grow up. I know that it did not keep you from growing into a talented man though, so I don’t feel so bad. I am very proud of you, brother, and look forward to seeing what your path will be in the coming years. Merci aussi à toi Maman pour ta patience et pour accepter le fait que je fasse ma vie loin de toi. Loin des yeux peut-être, mais pas loin du coeur, tu es toujours dans mes prières. Merci aussi Bernard de m’avoir considéré comme ton fils, et de m’avoir toujours encouragé à viser haut.

And the best friend award goes to...dun dun dun, Sara! Five years may not seem like such a long time, yet how full those years now look. Thanks for your patience throughout my long-winded studies! We have been through highs and lows, but since August 14, 2004 we covenanted to go through them together, no matter what. Thank you for this commitment you made to me, the commitment to love me in spite of all my weaknesses. I love you and look forward to a life with you. I am grateful that we both share the same goal, that we run the same race. Thank you for your willingness to follow God wherever it is He calls us. We may not be sure where the race will take us, but we know where it ends, praise God for that. Come, “glorify the LORD with me, let us exalt his name together” –Psalm 34:3.

Wonderful professors, great friends, an edifying Christian community, a supportive family, an excellent wife. Where does it stop? It doesn’t stop quite yet. John 1:16 says that “from [Christ’s] fullness we have all received grace upon grace.” Indeed Lord you gave me grace upon grace, blessing after blessing. You gave me more than I could imagine in terms of research, academic fulfilment, friends, and marriage. And yet you gave so much more, you gave me Yourself. And that is worth far more than all these things, nothing comes close to knowing you, my heavenly father. Thank you Jesus for making this possible. Thank you for your mercy and grace, upon grace, upon grace. I cannot repay you, so may my life be a worthy response to your love and sacrifice.

“What good will it be for a man if he gains the whole world, yet forfeits his soul?” – Matthew 16:26

“Seek first his kingdom and his righteousness, and all these things will be given to you as well.” – Matthew 6:33

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Recent military operations demonstrated the limitations of surveillance missions performed by high-altitude platforms (Unmanned Aerial Vehicle (UAV), U2, satellite), even if equipped with state of the art sensors. These limitations can for example affect target identification in the presence of a cloud cover or a tree canopy, as during NATO's operation *Allied Force* in Kosovo in 1999. Likewise aerial surveillance may only provide a limited assessment of the use of a facility, as was the case before operation *Iraqi Freedom* in 2003. Aerial surveillance proved to be insufficient for assessing whether or not certain facilities were being used for the manufacture of weapons of mass destruction. Furthermore, targets can be concealed by terrain features like caves (such as those in Afghanistan) or other man-made constructions (underground complex, decoys), so that they are hidden from high-altitude observers. In the examples just mentioned, the limitations are inherent to aerial surveillance, and they are unlikely to be resolved with any amount of improvement in the onboard-sensor technology. Today's surveillance needs are no longer satisfied by high-altitude observation. Rather, crucial information such as chemical, acoustic, biological and nuclear data about the targets is required, information which cannot be satisfactorily sensed from a high altitude.

In order to gain a clear understanding of the situation on the ground, it is becoming vital to observe from up-close, using sensing devices placed in the area of interest. For example, the U.S. Army Future Combat System (FCS) will "rely heavily on the use of remote, unattended sensors to detect, identify and track enemy targets" [75]. These sensors form a so-called Wireless Sensor Network (WSN), which provides all necessary information remotely.

The shortcomings of high altitude platforms are not limited to monitoring hostile activity. In open environments where a direct line-of-sight to aircraft or satellites is available, ground units can satisfactorily communicate with each other through these links. By combining this capability with GPS, they can also track their own position and that of other friendly forces, using for example the U.S. Army Force XXI Battle Command, Brigade-and-Below (FBCB2) [1]. However in more cluttered environments such as in dense urban areas, under-the-canopy, or indoors, harsh propagation conditions prevent ground units from efficiently communicating or accurately localizing themselves. In order to provide these critical capabilities in such environments, it is again necessary to deploy sensors on the ground, close to the agents.

Since the area in which the sensors operate is potentially hostile, they must be placed without any human physically involved, for example by dropping them from an aircraft, which can itself be a UAV. Moreover there might only be a limited number of sensors available for a specific mission, or sensors might be expensive requiring the mission to be performed with the fewest sensors possible (the sensing capabilities of miniature sensors such as Smart Dust [2] are very limited and cannot be used for missions requiring sophisticated sensing, such as cameras, microphones, etc.). Also, even if sensors are cheap and available in unlimited quantity, certain missions must still be performed with the fewest sensors in order to fulfill the objectives. As this thesis will show by example, dropping thousands of sensors randomly is often unacceptable. In addition, the mission may have objectives that conflict with one another. For example, two objectives may be maximizing the total coverage of the network and minimizing the number of sensors. Maximizing the coverage leads to layouts with more and more sensors, so that the second objective gets worse. Conversely, using few sensors typically yields a poor total coverage. In this case the objectives are conflicting, and the user must determine the trade-off he or she is willing to make between those two.

For these reasons, *the location of each sensor must be carefully determined*, so as to make the best out of every sensor and ensure the optimal performance of the network. Note also that because the sensors are deployed from an aircraft, their final ground location will have some uncertainty due to the airdrop inaccuracy (as opposed to the case where humans place the sensors manually, which they can do with very high accuracy). The aircraft deployment of sensors introduces some uncertainty in the position of the sensors.

The resulting complexity of the planning task (multiple, conflicting objectives, uncertainty) renders human instinct and expertise insufficient, and an optimization algorithm is required in order to satisfactorily solve this problem.

Once the necessary sensors are deployed on the ground, their data is transmitted back to the base in order to provide decision makers with the necessary situational awareness. In recent years researchers have given more attention to operational issues of WSNs such as energy-efficient communication [33, 84] and data fusion [14, 15]. Aspects related to the optimal operation of WSNs will be included in the proposed 3-phased planner of Chapter 2, but most of this thesis will be devoted to the issue of *sensor placement*. The only exception will be in Chapter 7 in the context of a UWB localization WSN, since those issues were not satisfactorily treated in the literature.

Such a sensor network can be used not only in military operations, but also for environmental monitoring. Sensor networks are used in Hawaii to assist the long-term study of rare and endangered species of plants [10]. In this particular project the sensors were hand-placed by human personnel, but it is easy to imagine that in some cases (e.g. rugged terrain), the sensors need to be dropped from an aircraft. Another example is that of an environmental disaster, such as a nuclear explosion, a biological leak or other scenarios where monitoring the area is required but human presence is dangerous. Sensors may be dropped in the contaminated area to monitor the evolution of the disaster. Placement algorithms are also needed for underwater sensing using hydrophones [21]. The common denominator in all these applications is a need to observe certain states from up-close, in an area that is hostile for human personnel (military or environmental threat).

In addition, an aerospace firm designing UAV's, drop vehicles or sensors could use this planning tool to conduct a sensitivity analysis in order to figure out where additional capability would be well invested. For example, with a limited research budget, one must decide whether the drop accuracy should be improved, or whether the effort should concentrate on extending the battery life, increasing the UAV payload, the sensors communication range or sensing range, etc.

1.2 Thesis Contributions and Outline

This thesis aims to provide a planning system for the automated deployment of WSNs. As mentioned above, the applications of WSNs are numerous, and the performance of the WSN is highly dependent on the position of the sensors. The network should be planned optimally, while taking into account the uncertainty due to the aerial deployment. First some general results are derived in Chapters 2 and 3.

- a planning strategy in 3 phases is proposed for robust WSN planning and deployment (Chapter 2);
- general-purpose algorithms are presented for solving Phases I and II (Chapter 3);
- although the placement of sensors and other WSN deployment issues are often neglected in the literature, we show through several case studies that planning is in fact critical for ensuring satisfactory performance of WSNs (Chapter 3).

Then in Chapters 4-7 a specific application is considered where the WSN is used to provide accurate localization to ground agents through ultra-wide bandwidth (UWB) ranging. This application extends the aforementioned blue force tracking capability to GPS-denied environments. The main contributions are as follows:

- analysis of the errors in UWB ranging and their statistical modeling (Chapter 4);
- calculation of a lower bound on the localization accuracy given a set of sensors locations. This provides the best performance achievable given the ranging errors, and is used subsequently as the objective to minimize (Chapter 5);
- derivation of a sensor placement algorithm that maximizes the localization accuracy in an area. This algorithm is shown to be optimal for a special case and to perform

very well in general (Chapter 6). In addition its flexibility makes it applicable to all 3 phases of the deployment process;

- demonstration of high accuracy localization for agents using the UWB WSN. This is achieved by an algorithm based on particle filtering, which provides a greater degree of accuracy due to its ability to estimate the ranging errors (Chapter 7).

Chapter 2

Planning Strategy: Phases I, II, and III

2.1 Wireless Sensor Networks

In this section we give a general model for WSNs, which will be used in Chapter 3. In Chapters 4-7 a WSN modeling specifically tailored to the localization application will be considered.

2.1.1 The Sensors

Sensors can be thought of as having two modules. The first is the *sensing* module, which can be of different types (seismic, acoustic, chemical, optical, etc.) and will depend on the mission at hand. The other is the wireless *communication* module, which is used to communicate with the other sensors and transmit the sensed information. Note that in some sensors the sensing module is the same as the communication module. This will be the case later on when a UWB WSN is considered. In this case UWB can be used for both ranging (the "sensing" part) and communication purposes.

In Chapters 2-3 we make the following assumptions about the sensors. First, a free space model is considered for the sensing and communication properties of the sensors, in other words:

- A sensor can sense anything within its *sensing range*, R_{Sensing};

- A sensor can transmit data to another one if the latter is within its *communication range*, R_{COMM}.

This is illustrated in Figure 2-1. We also assume, without loss of generality, that all sensors have the same sensing and communication range. These assumptions are for simplicity only. The results of Chapter 3 can easily accommodate heterogenous sensors and more realistic models, where for example the terrain influences both sensing and communication.

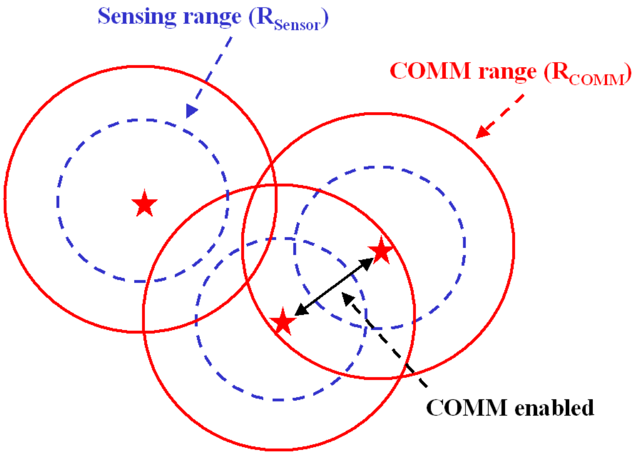
Sensing range (R_{Sensor})
COMM range (R_{COMM})
COMM enabled

Figure 2-1: The sensors have a sensing and a communication range, which can be different. Two sensors communicate if they are within R_{COMM}.

When lifetime issues are considered, the sensors' battery storage capacity will also need to be specified, along with the rate at which batteries are depleted, depending on the task performed (e.g. asleep, sensing, communicating).

The following illustrates the characteristics of an acoustic/seismic sensor [3]:

- *Weight*: $500g$;
- *Size*: $8cm \times 8cm \times 8cm$;
- *Sensing range*: $R_{\text{Sensing}} = 1000m$ for vehicles, $50m$ for people (unaffected by terrain or vegetation);
- *Communication range*: $R_{\text{COMM}} = 300m$;

- *Communication frequency*: $900MHz$;

- *Lifetime*: batteries last for 2 weeks.

2.1.2 The Network Architecture

Sensors are usually small, and therefore have limited energy. This prevents them from relaying their gathered information directly to some far-off UAV or relay satellite. For some missions this may be acceptable, for example if the WSN can perform its task in isolation from the outside world (e.g. a WSN solely devoted to relaying communications between ground units). However most missions will require transmission of the gathered data to the home base. It is therefore necessary to have at least one node with enough power to transmit data over longer distances. We call such node the *High-Energy Communication Node* (HECN). All sensors must then be able to transmit their data to this node, either directly or via hops using nearby sensors to act as communication relays. This requirement is called *communication connectivity*, and ensures that the network formed by all sensor nodes and communication links be connected. Figure 2-2 illustrates this architecture for a scenario where the WSN is used to monitor movements in a forest (sensors N_4 to N_{12}) and on a road (sensors N_1 and N_3). We will only consider scenarios with a single HECN without loss of generality.

2.1.3 The Aerial Deployment

As mentioned in Chapter 1, WSNs may need to operate in hostile or humanly inaccessible environments. The WSN therefore needs to be deployed remotely, with no human personnel involved. Our system utilizes a UAV to fly to the mission area and deploy the sensors. Drop vehicles may be available to guide the sensors in place, with different degrees of precision. Since the sensors are airdropped, their actual location once deployed will differ from their planned location. This is the *airdrop inaccuracy*. The planner will need to account for this uncertainty.

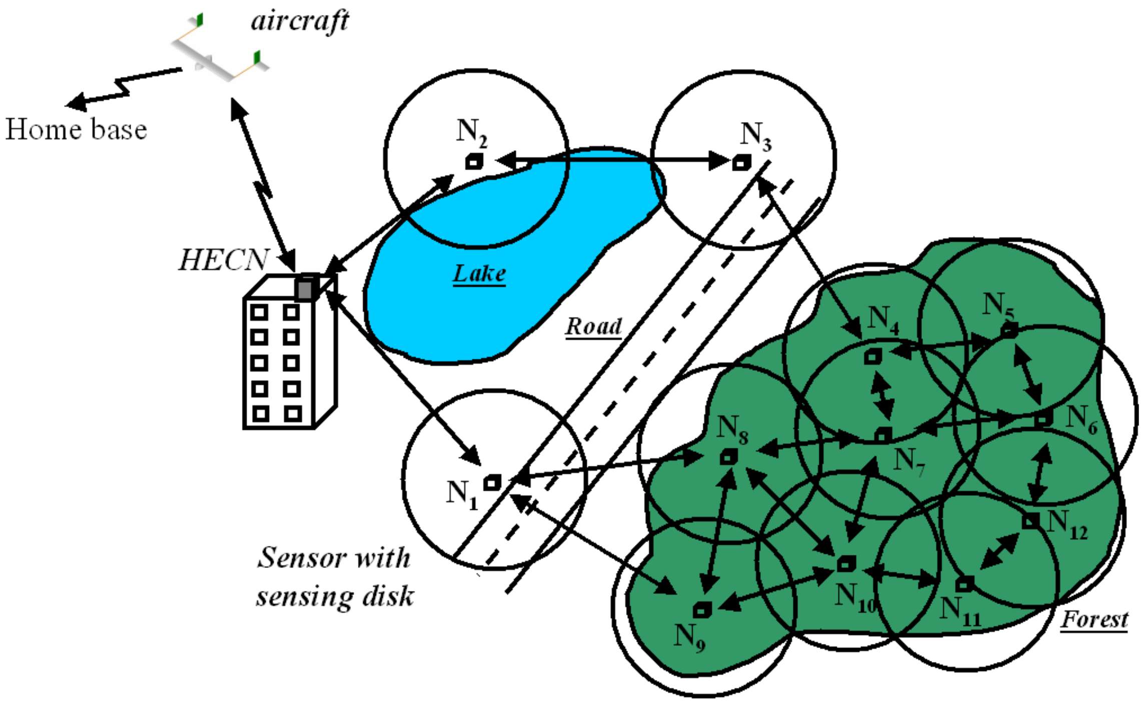

Figure 2-2: Illustration of a WSN monitoring a forest and a road. The HECN located on top of a building collects all the data gathered by the sensor, before relaying it back to the base. The sensing ranges are shown circles around each sensor.

2.2 Challenges to Optimal WSN Performance

Below is a summary of the many factors affecting the performance of WSN:

1. Once on the ground, the sensors must efficiently operate and interact (e.g. networking, information fusion);
2. The physical location of the sensors strongly influences the performance of the WSN;
3. Multiple, conflicting objectives may need to be taken into account. These include network coverage, lifetime, survivability, or the number of sensors;
4. The objectives often are nonlinear or discrete, and sometimes have to be calculated through simulation ("black box");
5. Uncertainties are present due to:

    - The sensors are deployed from a UAV, which introduces uncertainty in the location of the sensors. Sensors may also malfunction due to the airdrop or for other various reasons;
    - Inaccuracies in the sensors capabilities (in terms of sensing or communication ability);
    - The mission objectives may be changing with time. As the WSN performs its task, new information may change the goal of the WSN user.

6. Although we have assumed an idealized terrain model, in general the terrain will affect the sensing and communication ability of the sensors. There may also be areas where sensors should not be dropped (e.g. lake);

7. Additional constraints must be accounted for:
    - Connectivity constraints between the sensors and the HECN. The sensors also act as relay for their neighbors, in order to circulate the sensed information;
    - Payload constraints on the UAV;
    - Lifetime constraints on the UAV. If the mission area is hostile, the UAV has a probability of being shot down at each pass over the area, so the deployment of the sensors must be as fast as possible.

These challenges will often interact with one another. For example the fact that the aircraft carrying the sensors has a limited payload implies that it is impossible to drop thousands of sensors randomly over the area of interest, hoping the necessary coverage and communication connectivity to arise by chance. Moreover one objective may be to minimize the number of sensors deployed.

In light of the complexity of the task, an automated planning tool is necessary to plan the WSN deployment.

2.3 Literature Review

Several areas of research are relevant to the WSN planning problem.

A large body of research was devoted to the Base Station (BS) location problem for cellular phone networks in the 1990's. BS antennas must be placed so that every user is covered by the network, i.e., so that their communications can be relayed by at least one BS antenna.

Each BS typically has to be located above the users, i.e. on rooftops or on dedicated towers, which in both cases is a costly enterprize. There is a strong incentive to achieve full coverage with the minimum number of stations. In addition these stations often operate in urban environments, where wave propagation is complex. The resulting complexity of this problem led network designers to develop optimization algorithms in order to help automate the planning task. In spite of its similarities with the WSN planning problem however, the BS problem differs from it in several aspects. One main difference comes from the fact that in WSNs the sensors (analogous to the BS) also need to communicate with each other wirelessly (connectivity), whereas BS are connected to the backbone network via ground wires. Also, the payload constraint, and more importantly the inaccuracy of deployment are peculiar to aerial deployment (BS are placed at the exact planned location by human personnel). Finally, WSNs must be deployed faster than cellular networks, which can take months to plan. For example some military applications may require a WSN to be deployed as soon as possible, typically within a few hours. In spite of these differences, the two problems are similar enough that reviewing the methods used to plan cellular phone networks is fruitful. These range from Dynamic Programming [81] and unconstrained optimization [24], to Genetic Algorithms [45, 73] and Tabu Search [6]. Virtually every type of optimization technique was tested on this problem, many dealing with multiple objectives (though almost always blended in a single objective function, except in [73] which uses Pareto optimality), while using non-trivial communication models accounting for the terrain.

The BS location problem is part of the larger research area of Facility Location in Operations Research [32]. Here a set of demand points must be covered by a set of facilities, which corresponds in WSN to covering an area with a set of sensors. The goal is to locate these facilities so as to optimize a certain objective (e.g. minimize the total distance of demand points to their closest facility). A classic example close to the WSN problem is the Maximal Covering Location Problem (MCLP) [26, 72], where as many demand points as possible must be covered with a number s of sensors of fixed radius. It is also referred as the location-allocation problem, since each demand point must be assigned to a certain sensor. Suzuki and Drezner [91] addressed a similar problem on the continuous plane, i.e., the problem of locating s facilities to cover a whole planar region, while using the smallest possible radius. Salapaka discusses a more sophisticated version of this class of problems in [82]. A set of target points on the ground needs to be observed from above, with each target possibly requiring different sensors depending on its nature. Salapaka implements a Deterministic Annealing technique that assigns the required resources (sensors) to each aircraft and then determines the aircraft position so that the targets are covered as best as possible. Again in all these discussions, the main difference with the WSN problem is that the nodes are not required to be connected. Another problem of interest is the Facility Location-Network Design problem, where facility positions need to be determined (just as in the MCLP) and the network connecting these facilities must also be optimized. Unfortunately, in the WSN design it is impossible to decouple sensor placement and network design, since the locations of the sensors determine the network topology.

The past five years have seen a rising interest in sensor network planning, focusing mostly on optimizing the location of the sensors in order to maximize their collective coverage. Although this is a problem almost identical to the BS location problem, the work on BS location is rarely mentioned. Chakrabarty [20] used Integer Programming, while Bulusu [16], Dhillon [30] and Howard [48, 50] devised different greedy heuristic rules to incrementally deploy the sensors. Virtual force methods (VFM), often used for the deployment of robots [49], were adapted by Zou [104] for the sensor deployment. Finally the author implemented a Genetic Algorithm to solve this placement problem [55, 56], which will be described in the next chapter. Some also looked at communication connectivity problems from a theoretical point of view [10, 61], but this seems to have little value in realistic, asymmetrical terrain conditions.

More attention has been paid to adaptive techniques for the deployment of sensors, e.g. when more sensors are needed [16, 30]. This accounts for the nature of the deployment, happening in real-time, whereas the BS planning problem is assumed to be offline. As mentioned before, current work on WSN mainly focuses on the maximization of the total sensing coverage, with little or no attention given to the communication connectivity. Meguerdichian [71] assumes that the communication radius of the sensors will be much larger than the sensing radius, so that connectivity will arise naturally. But this assumption is unrealistic for two reasons. First, there exist sensors for which the sensing range is of the same order or larger than the communication range (e.g. the seismic sensors described in Section 2.1.1), so that maximizing the coverage without caring about the communication range will result in a disconnected network. Second, if the areas to be covered are disjoint, the network will be partitioned. In addition, in our WSN model the sensors must be connected not only to each other but also to a HECN. Therefore the communication connectivity requirement cannot be trivialized and both aspects of the sensors (sensing and communicating) must be taken into account for the planning of the network.

The research in WSN planning deals only with a single objective (almost always coverage) whereas it seems that other considerations are also of vital importance in the choice of the network layout (lifetime, survivability, etc.). The sensing model usually does not account for terrain. The areas considered are flat squares, and only Howard [49] and Dhillon [30] include a few obstacles obstructing communications. Finally there is no concern about aerial deployment (payload, inaccuracy).

Beginning in Chapter 4 a WSN used for localization in GPS-denied environments will be considered. A separate literature review will be performed at that time, suffice to say for now that the optimal placement of sensors in that context has received very little attention.

2.4 Proposed Strategy

In order to ensure that the actual network placed on the ground meets its planned performance in spite of the uncertainties, the planning method should not only be able to plan a network design from scratch, but it should also be adaptive, i.e., it should be able to respond to drop inaccuracy or additional sensor deployment while onsite.

The system should therefore have several “time constants”: one on the order of a few hours (initial planning of the WSN, before take-off), the other on the order of a few minutes (in-flight adaptive planning, while the aircraft is onsite and sensors are being dropped). For these reasons the planning sequence was divided into 3 phases, as described below and summarized in Figure 2-3.

Phase I is performed on the ground and generates the initial network design. It determines the type and number of sensors to be loaded inside the UAV, as well as the location of each sensor. It includes a multi-objective optimization algorithm to generate those designs under several constraints. It may also account for the uncertainty in sensor placement, or the terrain information. This phase can take on the order of several hours. Once Phase I is completed, the necessary sensors are loaded inside the UAV and the aircraft flies to the drop zone.

Phase II is initiated when the UAV reaches the mission area and starts deploying sensors. This phase is aimed at adding robustness to the deployment process, especially to cope with the uncertainty. The accuracy of the *drop vehicles* used to deploy the sensors from the UAV is characterized by a probability distribution with standard deviation σ_{DV}. Even the most accurate drop vehicles will have a non-negligible σ_{DV}. Indeed the most accurate placement is currently achieved through ballistic drops, which have a very high terminal velocity. While this is acceptable (and even desirable) for seismic sensors which must be buried in the ground, some sensors will require a softer ground impact. But a smaller terminal velocity implies a loss of accuracy. Typically, the best σ_{DV} that low-velocity impact drop vehicles can currently achieve is about $4m$, which is accurate enough to drop a sensor on a rooftop or a large tree [4]. This inaccuracy makes it important to close the loop on the deployment process. Phase II must therefore decide which sensors to deploy first, and then it must adapt the drop location of the remaining sensors depending on the result of the previous drops. This implies an ability to dynamically re-plan the optimal WSN configuration computed earlier, which must be done quickly (on the order of minutes).

Finally **Phase III** accounts for the operation and maintenance of the WSN, once all the sensors have been dropped. Optimal operation of the WSN requires algorithms to optimally route the information, conserve energy, or fuse the information. This part of Phase III is probably the area in WSN research that has received most attention. The maintenance aspect of Phase III deals with sensor placement (just like Phases I and II). As the network operates and gathers data, the mission objectives may change, or the sensing or communication model may be updated, so that additional sensors need to be deployed in order to fulfill the mission requirements.

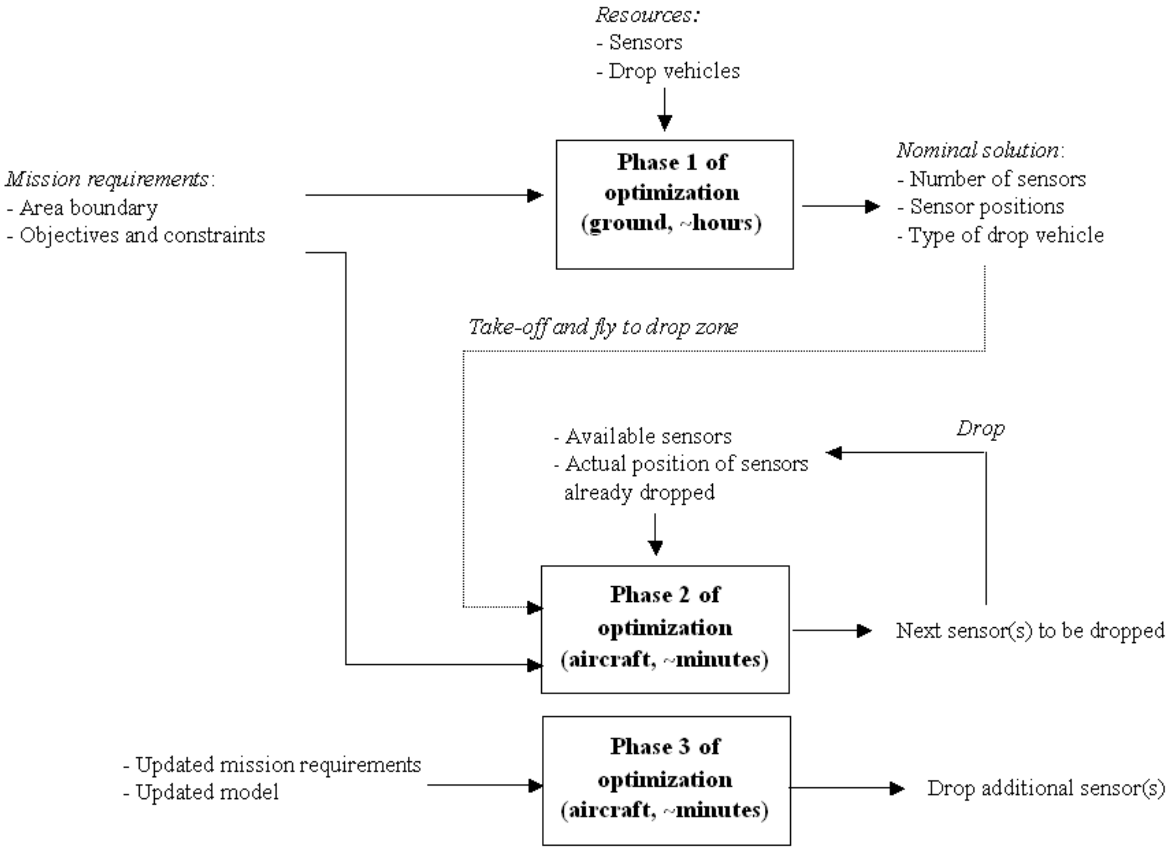
Resources:
- Sensors
- Drop vehicles
Mission requirements:
- Area boundary
- Objectives and constraints
Phase 1 of optimization (ground, ~hours)
Nominal solution:
- Number of sensors
- Sensor positions
- Type of drop vehicle
Take-off and fly to drop zone
- Available sensors
- Actual position of sensors already dropped
Drop
Phase 2 of optimization (aircraft, ~minutes)
Next sensor(s) to be dropped
- Updated mission requirements
- Updated model
Phase 3 of optimization (aircraft, ~minutes)
Drop additional sensor(s)

Figure 2-3: Diagram summarizing the 3 phases of planning and deployment.

2.5 Thesis Limitations

We end this chapter on a word about the limitations of this thesis. Because it is at the core of each of the 3 phases, the focus of this thesis will primarily be to develop *sensor placement algorithms*. In particular the next chapter will present a general purpose placement algorithm for Phase I, and Chapter 6 will present a custom-tailored placement algorithm when the WSN must provide accurate localization in GPS-denied environments. The latter algorithm will be applicable to the placement problem in Phases I, II, and III.

The results on Phase II presented in the next chapter will be mostly aimed at demonstrating the need for Phase II planning through a few examples. An approximation algorithm will be presented there, but a more thorough treatment will still be needed.

Phase III will not be treated in generality. It will only be considered in the context of the UWB localization WSN. Chapter 6 will then provide an incremental placement algorithm, and Chapter 7 an algorithm for optimally fusing the range measurements to provide accurate localization.

2.6 Summary

In this chapter a general WSN architecture was described, along with the challenges to optimal network performance. Based on those challenges a planning strategy was proposed, which was divided into 3 phases.

In the next chapter some algorithms will be developed to implement this strategy in the general case. The importance of Phases I and II will be outlined by comparing the performance of WSNs deployed using the planner versus that of WSNs deployed randomly. These studies will be performed on several case scenarios.

In Chapters 4-7 a particular application will be considered, where UWB sensors are used to provide accurate localization in GPS-denied environments. A specifically tailored algorithm will be described for the sensor placement problem of Phases I, II and III, and again the benefit of using a planning system for the network deployment will be outlined.

Chapter 3

General Results on Phase I and Phase II

3.1 Introduction

In this chapter a general purpose algorithm for Phase I is proposed. This algorithm is general in the sense that it is not tailored to any specific objective or mission, but rather works with any kind of scenario that fits the modeling of Chapter 2. It will therefore have the advantage of being versatile and able to account for all the challenges mentioned in the previous chapter. A drawback is that, because of its generality, it may not be the fastest for some types of problems. In particular Chapter 6 will present another algorithm for a specific, narrower scenario, which will prove faster and near-optimal in terms os sensor placement.

An algorithm for Phase II is then presented and is analyzed through an example. Our goal here is mainly to motivate the importance of Phase II planning.

The benefit of using a planning algorithm for Phases I and II will be showcased on several scenarios, which we now describe.

3.2 Case Scenarios

In both case scenarios considered here a flat square area of side 10 (in arbitrary units) is considered. We assume a single HECN is present, which is for convenience and does not prevent generalization. Every sensor must communicate with the HECN either directly or via hops through nearby sensors. Only connected sensors will be taken into account in the objective calculation, as the data from disconnected sensors cannot be collected by the HECN and is therefore useless.

The design variables are the x-y coordinates of the sensors in the plane, where the center of the area is taken as the origin. The vector of design variables $\mathbf{p}$ is a column vector of size $2n$, where n is the maximum number of sensors and the coordinates (x_i, y_i) of the ith sensor are stacked as shown below:

$$\mathbf{p} = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix}. \tag{3.1}$$

The goal of the optimization is to find the optimal design vector $\mathbf{p}$, given certain objectives and constraints.

Finally the sensors are assumed to be identical, with a common R_{Sensing} and R_{COMM}. Again, this assumption is for simplicity and can be relaxed in other applications.

3.2.1 Case Scenario 1 (CS1)

In case scenario 1 (CS1) the sensors must be placed so as to maximize 2 mission objectives, the network *coverage* and *lifetime* [55]. This corresponds to a mission where surveillance must be provided in an area for an extended period of time. The HECN is placed at the center of the area.

Each sensor initially has the same energy available in its battery, which then decreases by one arbitrary unit for every data transmission. The cost of sensing is not modeled, as we assume that during the mission all the sensors spend the same amount of energy on the sensing task.

Note that because of the symmetry of this scenario about the origin, similar layouts rotated about the origin will have similar objectives, but different design vectors $\mathbf{p}$.

Coverage

The *coverage* of design $\mathbf{p}$ is equal to the fraction of area occupied by the sensing disks of connected sensors:

$$\mathsf{Coverage}(\mathbf{p}) = \frac{\int_{\cup_{i\in\mathcal{C}(\mathbf{p})}\mathcal{D}(x_i,y_i)} dxdy}{\int_{\mathcal{A}} dxdy}, \tag{3.2}$$

where $\mathcal{C}(\mathbf{p})$ is the set of indices of connected sensors, $\mathcal{D}(x_i, y_i)$ is the disk of radius R_{Sensing} centered at (x_i, y_i), and $\mathcal{A}$ is the total area. This objective is to be maximized.

The following routine is used to numerically compute the coverage of a given network design $\mathbf{p}$:

1. Discretize $\mathcal{A}$ into n_{grid} grid points;
2. Initialize the number of covered point $n_{\text{covered}} = 0$;
3. For $j = 1 \ldots n_{\text{grid}}$, do:
    - If grid point j is within R_{Sensing} of one of the n sensors, then: $n_{\text{covered}} \leftarrow n_{\text{covered}} + 1$;
    - Else: $n_{\text{covered}} \leftarrow n_{\text{covered}}$.
4. Set $\mathsf{Coverage} = n_{\text{covered}}/n_{\text{grid}}$.

Lifetime

The lifetime of the network is to be maximized. In order to define the lifetime of a WSN we need to model how quickly the sensors' energy is depleted. We describe below a simple model for how the network operates. It is by no means the only valid one, but it captures the most salient feature about WSN lifetime, namely that the network topology will require some sensors to do more "work" than others, leading to shorter battery life.

In our model the sensors first gather data about the environment. Each sensor encodes its gathered data into a *data packet* which is then sent to the HECN. Once all the data packets are collected by the HECN, the whole process (called a *sensing cycle*) can start again. Some sensors will have a direct communication link to the HECN, so that their data packet can be passed directly (in one hop) to the HECN. However other sensors will not have such a direct link, instead their data packet will “hop” from one neighboring sensor to another in order to reach the HECN. These neighboring sensors will not only have to transmit their own data packet, but also that of other sensors (in which case they act as communication relay). Therefore at every sensing cycle some sensors will be required to do more data transmission than others, resulting in an asymmetric energy depletion among sensors, causing some to fail earlier than others.

We assume that the sensors initially have $\mathcal{E}$ (arbitrary) units of energy stored in their batteries initially, and each data transmission costs 1 unit of energy. A sensor can therefore survive at most $\mathcal{E}$ sensing cycles, in which case the sensor would have had to only transmit its own data packet each cycle. For a given network design $\mathbf{p}$, a connectivity matrix C can be obtained, such that $C_{i,j} = 1$ if the distance between sensors i and j is less than R_{COMM}, 0 otherwise. Note that C is of size $(n+1) \times (n+1)$, where the $(n+1)^{\text{th}}$ row and column correspond to the HECN. From this matrix we can construct a graph where each sensor is a node and an arc exists between nodes i and j if $C_{i,j} = 1$. This is illustrated on Figure 3-1 with a WSN composed of 3 sensors. Four communication links are present, shown as line segments. The corresponding connectivity matrix is a 4 by 4 matrix (3 sensors and the HECN) and is given by:

$$C = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}. \tag{3.3}$$

The only remaining piece that is needed is a routing scheme which, for each sensing cycle, describes how the data packets are directed from every sensor to the HECN. Sensors with a direct link to the HECN will obviously transmit directly to it. However when multiple hops are required to reach the HECN, the data packet may have several possible paths to get to the HECN. For example, the data of N_3 in Fig. 3-1 can reach the HECN by going either through N_1 or N_2. The routing scheme is the set of rules whereby this path is chosen. For WSN the main challenge is to route the information so as to conserve the energy of the

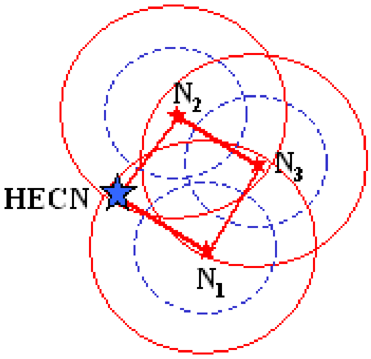

Figure 3-1: Example of a WSN with communication links shown as solid lines.

network, and many algorithms have been proposed and studied to that effect [51, 86].

We adopt a routing algorithm that maximizes the remaining energy in the sensors. At each sensing cycle, consider the connectivity graph formed by the WSN (similar to that of Fig. 3-1). If an arc exists between nodes N_i and N_j, it is split into 2 directional arcs, one from N_i to N_j and the other from N_j to N_i. Then weights are assigned to each arc depending on the remaining energy e_i in each sensor i. In particular the outgoing arcs of node i are weighted by the inverse of the node's remaining energy, or $1/e_i$. Arcs going out of nodes with little energy will thus receive a large weight. Dijkstra's algorithm is then run from every node to the HECN. This algorithm solves the single-source shortest path problem on a weighted, directed graph, so for each node it finds the route of minimum aggregate weight to the HECN [28]. This routing scheme will therefore avoid overusing nodes with little energy, if possible. Once the routing of data packets is determined for all nodes, we can count the number of data transmissions performed by each sensor. The energy of each sensor is decreased accordingly, and the process is repeated for another sensing cycle. If all of a sensor's energy is depleted, then this sensor is ignored at the next sensing cycle. This whole process enables us to simulate the WSN operation for a given configuration $\mathbf{p}$ and determine how many sensing cycles $n_{\text{cycles}}(i)$ sensor i is able to perform, for all $i = 1 \ldots n$. This process is illustrated on Figure 3-2 for the WSN of Fig. 3-1. The sensors energy is shown next to each sensor, starting with 6 units of energy each. The routing paths for each cycle are shown by bold arcs. Note that the data packets of N_3 are relayed alternatively by N_2 and N_1, so that the total number of sensing cycles is

4. It is easy to verify that if the same routing was used at each sensing cycle (e.g. packets from N_3 always go through N_2), the number of sensing cycles would be 3 instead of 4. On this simple example this routing scheme increased the WSN lifetime by 33%.

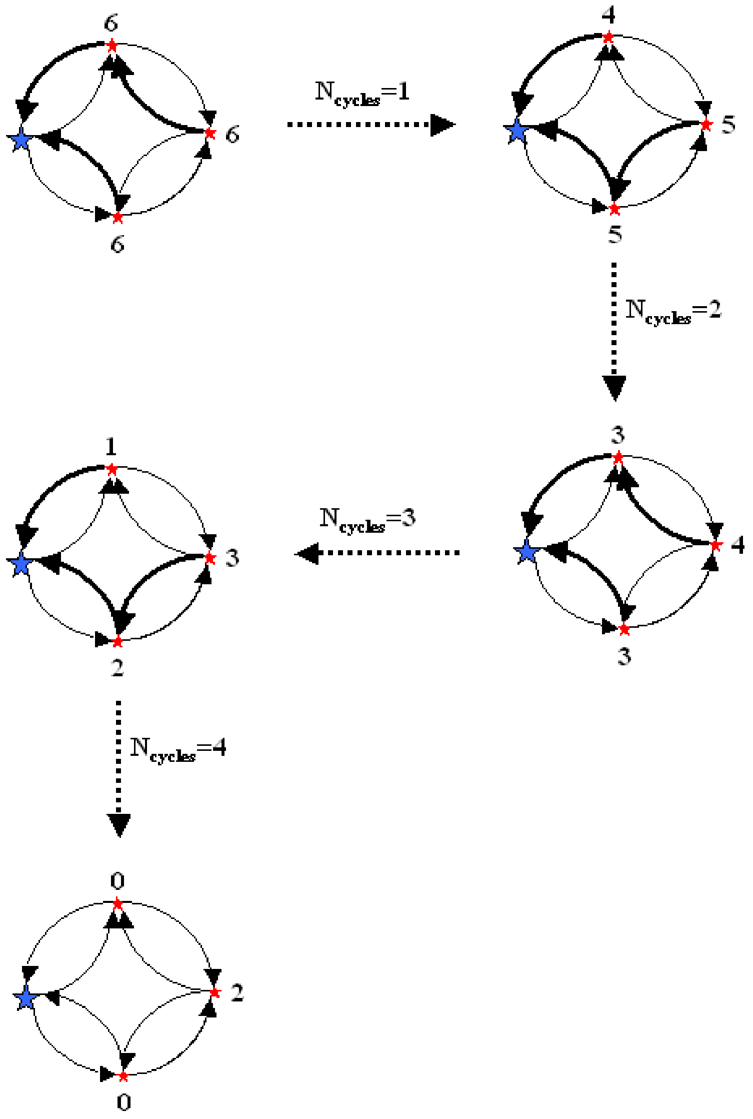

Figure 3-2: Illustration of the routing scheme on the WSN of Fig. 3-1. A total of 4 sensing cycles can be achieved, versus only 3 if a fixed routing path was chosen beforehand.

Based on this model there are many possible ways to define the lifetime of a WSN. One possibility adopted here is to say that the network is fully operational as long as all the (initially connected) sensors remain connected. Therefore the WSN will "die" according to our definition when one sensor no longer has any energy to transmit its data. The WSN will therefore operate for $n_{\text{cycles}}(\mathbf{p}) = \min_{i \in \mathcal{C}(\mathbf{p})} \{n_{\text{cycles}}(i)\}$. The lifetime is then defined as the ratio between $n_{\text{cycles}}(\mathbf{p})$ and the maximum lifetime of a sensor:

$$\mathsf{Lifetime}(\mathbf{p}) = \frac{n_{\text{cycles}}(\mathbf{p})}{\mathcal{E}}. \tag{3.4}$$

This is a conservative definition, and for practical systems one may choose another one[1]. However it captures the desired trade-off, so that coverage and lifetime, the two objectives, are competing. On the one hand the coverage objective will desire spread-out network layouts, where sensors are far apart from each other in order to minimize the overlap between sensing disks. This implies that at each sensing cycle the sensors close to the HECN will have to relay a large number of data packets from the outlying sensors. This will result in faster energy depletion for those sensors, leading to their early failure. The resulting network lifetime will be small. On the other hand, in order to get a lifetime of 1 all the sensors must communicate directly to the HECN, so that their energy is used only for their own data transmission. This implies a configuration with all the sensors clustered around the HECN, yielding a poor coverage value due to the overlap between sensing disks.

The computation of the lifetime for a given network design $\mathbf{p}$ is summarized below.

1. Compute the connectivity matrix C;
2. Initialize the energy $e_i = \mathcal{E}$ for $i = 1 \ldots n$, and the number of sensing cycle $n_{\text{cycles}} = 0$;
3. While $e_i > 0$ for all $i = 1 \ldots n$, do:
    - Form the weighted connectivity matrix C^w such that $C^w_{i,j} = 1/e_i$ for all i, j such that $C_{i,j} = 1$;
    - Run Dijkstra's algorithm from each node to the HECN;
    - For all $i = 1 \ldots n$, count how many packets p_i node i has to transmit, and update its remaining energy $e_i \leftarrow e_i - p_i$;
    - $n_{\text{cycles}} \leftarrow n_{\text{cycles}} + 1$.

---

[1] For example one could instead choose the time at which the coverage drops below 50% of its original value.

4. Set Lifetime $= n_{\text{cycles}}/\mathcal{E}$.

Note that since the lifetime is defined by the first sensor failure, the inner loop only needs to be run until a node's energy is depleted.

3.2.2 Case Scenario 2 (CS2)

In case scenario 2 (CS2) a facility served by two roads is located in a hostile area. The WSN must monitor the movements in and out of the facility [56]. For example the facility could be a terrorist training camp, or a plant suspected of manufacturing nuclear material, so that monitoring the activity could yield valuable information. The activity must be monitored along the roads (where it is most likely to occur), but also around the facility as a whole. Moreover, since the area is hostile, the survivability of the WSN depends on whether the sensors will be detected or not. The closer that sensors are placed to the facility or to the roads, the greater the probability that they will be discovered. Finally the network is deployed from an aircraft that can only carry a limited payload, so the number of sensors must be minimized.

The HECN is placed at the top right corner of the area, as shown on Figure 3-3. The circular facility is located at the center of the area, while two roads stem from it, one going East, the other North. We assume that any sensor placed inside the facility will not operate, so any such sensor will not be taken into account in the objectives calculation. A description of the objectives used in CS2 is now given.

Activity Detection

This objective measures the ability of the WSN to monitor movements in and out of the facility and is to be maximized. A series of n_{radial} equally spaced radial lines stemming from the facility are generated, which approximate the possible routes agents can take to enter or exit the facility (Figure 3-4). A sensor covers a line if the distance between them is less than R_{Sensing}, i.e., if the line crosses the sensor's sensing disk. The detection is then equal to the number of lines covered by the sensors $n_{\text{covered}}(\mathbf{p})$, divided by the total number

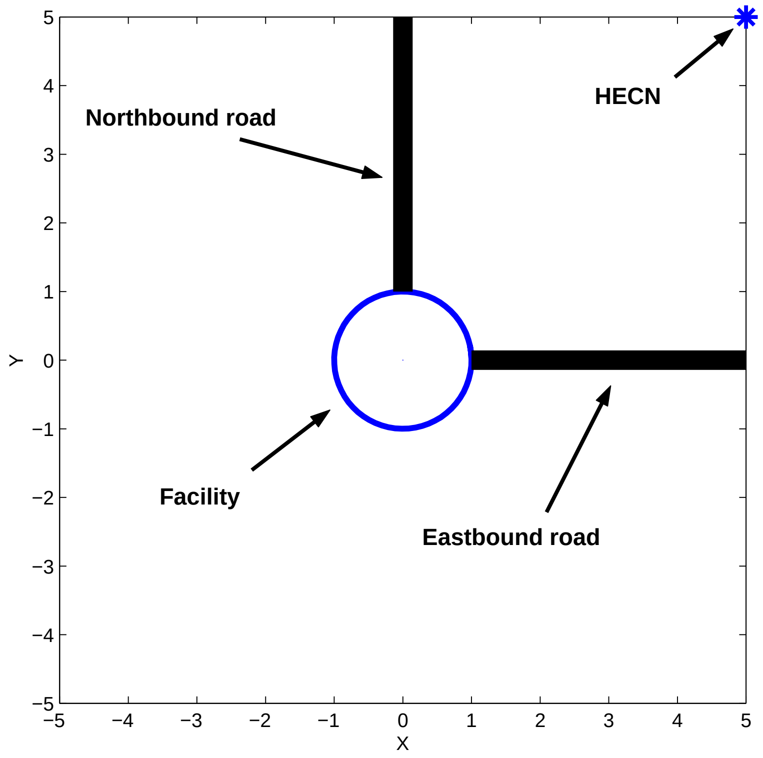

Figure 3-3: Map for CS2 with the facility, the two roads and the HECN.

of lines, so that it is between 0 and 1:

$$\text{Detection}(\mathbf{p}) = \frac{n_{\text{covered}}(\mathbf{p})}{n_{\text{radial}}}. \tag{3.5}$$

Since the facility is served by two roads, it is likely that the activity in and out of the facility will take place mostly along them. More lines can then be generated along the roads, so that these directions carry a greater weight.

Survivability

The second objective is the survivability of the network, which is to be maximized. It represents the likelihood that the sensors will *not* be found. Each point (x, y) in the area is assigned a probability of destruction $\mathbb{P}_{\text{destruction}}(x, y)$, that is, the probability that a sensor placed at (x, y) will be found and destroyed. This probability depends on the proximity of the point to the facility or the roads. Indeed we assume that most of the activity takes place

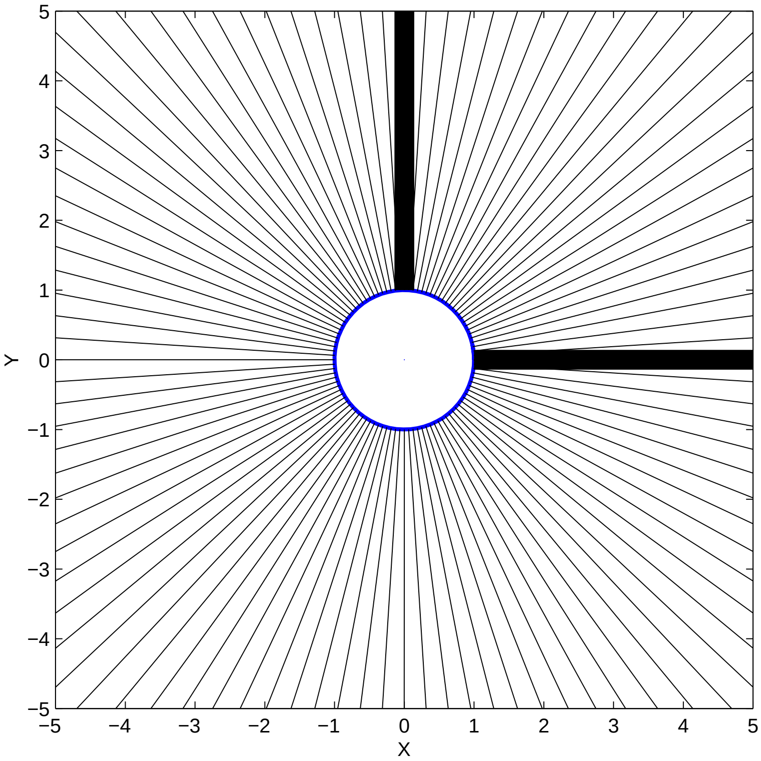

Figure 3-4: Lines used to calculate the detection objective in CS2.

on the roads or in close proximity to the facility. Therefore if a sensor is placed close to these features it is more likely to be discovered and disabled. For the purpose of this study a mapping of this probability is generated heuristically using a sum of exponential functions that decay with the distance to the features (3.6). We assume that the aggregate threat from the 3 features is the sum of 3 components: $\mathbb{P}_{\text{facility}}(x, y)$, $\mathbb{P}_{\text{RoadE}}(x, y)$, and $\mathbb{P}_{\text{RoadN}}(x, y)$, corresponding respectively to the facility, the eastbound road, and the northbound road.

$$\mathbb{P}_{\text{destruction}}(x, y) = \mathbb{P}_{\text{facility}}(x, y) + \mathbb{P}_{\text{RoadE}}(x, y) + \mathbb{P}_{\text{RoadN}}(x, y), \tag{3.6}$$

with (for a square area of side $2L$)

$$\mathbb{P}_{\text{facility}}(x, y) = \frac{1}{6}\left[1 - \frac{\sqrt{x^2+y^2} - R_{\text{facility}}}{\sqrt{2}L - R_{\text{facility}}}\right] + \frac{1}{6}e^{-\frac{1}{2}\left|\sqrt{x^2+y^2}-R_{\text{facility}}\right|^3}, \tag{3.7}$$

$$\mathbb{P}_{\text{RoadE}}(x, y) = \begin{cases} \frac{1}{2}e^{-\frac{1}{2}|y|^3} & \text{if } x \geq R_{\text{facility}}, \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

$$\mathbb{P}_{\text{RoadN}}(x, y) = \begin{cases} \frac{1}{2}e^{-\frac{1}{2}|x|^3} & \text{if } y \geq R_{\text{facility}}, \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

where $L = 5$ is half the side of the area and $R_{\text{facility}} = 1$ is the radius of the facility. This function is plotted on Figure 3-5. Note that $\mathbb{P}_{\text{destruction}}(x, y)$ is maximum at the "gates," i.e. where the roads meet the facility. The survivability of the network (as we define it) is obtained by finding the sensor with maximum probability of destruction and subtracting this value from 1:

$$\mathsf{Survivability}(\mathbf{p}) = 1 - \max_{i \in \mathcal{C}(\mathbf{p})} \left\{ \mathbb{P}_{\text{destruction}}(x_i, y_i) \right\}. \tag{3.10}$$

Note that since this objective only takes into account the sensor with highest probability of destruction, whether one or all the sensors have a high destruction probability will not matter. In this sense this objective captures the cost of having the presence of the WSN discovered at all: if one sensor is discovered, it is easy to imagine that the enemy will search for the others.

Number of Sensors

The third objective is the number of sensors n, which is to be minimized.

The first two objectives, detection and survivability, are competing. For a fixed number of sensors, in order to have more detection capability the sensors need to come closer to the facility, so that they can make a closed loop around it. However doing so decreases the survivability of the network since the sensors are close to the threats. Conversely a high survivability is obtained when all the sensors are far away from the roads and the facility, yielding a poorer coverage. Also, the more sensors the more coverage and the closer to the edges of the area the WSN can spread (yielding a good survivability). So the third objective is competing with the other two.

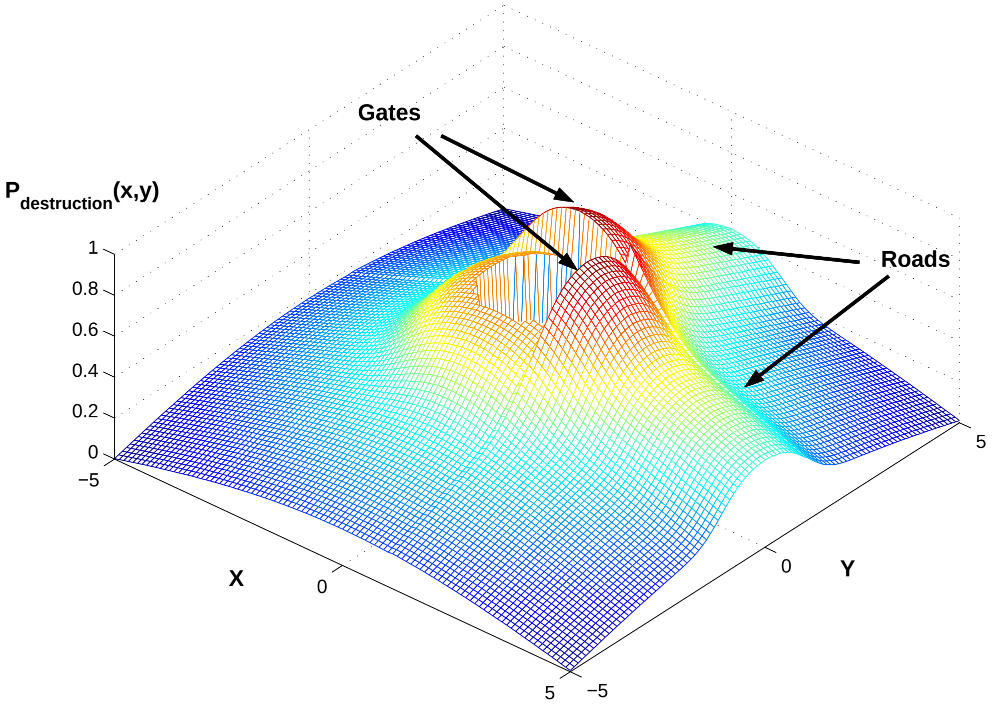

Figure 3-5: Mapping of the probability of destruction of a sensor.

3.3 Phase I: Multi-Objective Genetic Algorithm (MOGA)

In this section the sensor placement algorithm used for Phase I is presented.

3.3.1 Motivation

The Phase I optimization can be formalized as follows:

$$\max\{\mathsf{Objective}_1, \ldots, \mathsf{Objective}_\mathsf{m}\} \tag{3.11}$$

$$\text{subject to} \begin{cases} \text{the connectivity constraint} \\ \text{maximum number of sensors } n \end{cases} \tag{3.12}$$

Although the terrain is idealized as a flat surface, the design space of the WSN optimization remains highly non-linear. This is due primarily to the binary nature of the communication connectivity constraint between sensors. Moving a sensor by a small amount can cause large changes in all the objectives, especially if it becomes disconnected. This is illustrated in Figure 3-6 for the detection objective of CS2, where 4 sensors are fixed (Fig. 3-6(a)), while the fifth one is moved throughout the area in order to map the corresponding objective. Discontinuities can be observed in the detection (Fig. 3-6(b)), although only one sensor is moved. These discontinuities correspond to positions where the fifth sensor becomes disconnected from the rest of the network. As long as it is inside the communication region of one of the fixed sensors (solid circles), its coverage contributes to the overall detection when it is in locations not already covered by the other 4 sensors. But once it leaves this region, it cannot communicate anymore with the HECN so the detection drops abruptly. The maximum for the detection is found on a sharp edge of the design space, with a sudden drop on one side. The combined effect of all sensors renders these discontinuities even more severe.

In addition, the lifetime objective and the number of sensors objective vary in discrete amount, which makes the optimization harder.

The Phase I algorithm must therefore be able to cope with:

- Variety of objectives: Phase I should be flexible to accommodate diverse objectives without changing the algorithm;
- Multiple objectives, often non-analytical (coverage, detection) or discrete (number of sensors, lifetime);
- Nonlinear, nonconvex design space.

These challenges call for a heuristic method, and the Genetic Algorithm (GA) is well-suited for this problem [43]. It can work with problems where the objectives are nonlinear, discrete, or obtained through simulation ("black box"). It also has a multiple-objective variant, the Multi Objective Genetic Algorithm (MOGA), which easily handles multiple objectives through Pareto-optimality [27,36]. MOGA is also very flexible and can be applied without any modifications to different scenarios with different objectives. Among its drawbacks are the absence of convergence guarantees and a typically heavy computational cost. In spite of these, good results were obtained on the scenarios considered.

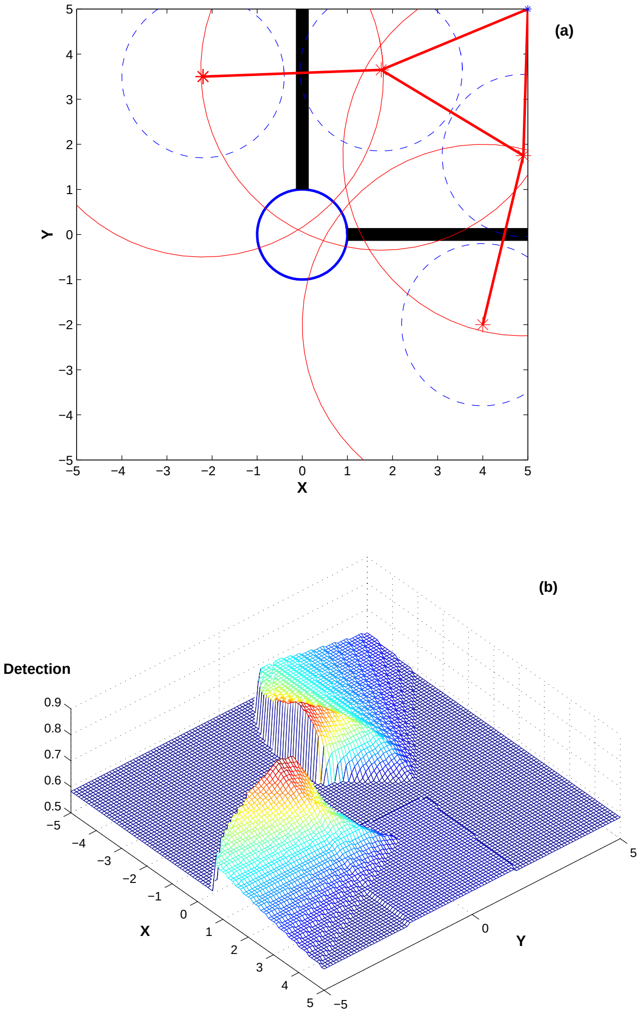

Figure 3-6: (a) Set up for CS2 with 4 sensors fixed, while the fifth is moved throughout the area. (b) Mapping of Detection as the sensor is moved throughout the area.

3.3.2 Dominance and Pareto ranking

Before describing the mechanics of MOGA, we first need to define the concepts of dominance and Pareto ranking. When a single objective is considered, different designs can be compared by simply comparing the respective objective values. However when there is more than one objective (without any *a priori* preference of one over another) such comparisons do not make sense. Indeed, what is meant by a design being "better" than another itself needs to be defined. This is what the concept of *dominance* does. Let $\mathbf{J^1} = [J_1^1, \ldots J_m^1]^{\mathrm{T}}$ and $\mathbf{J^2} = [J_1^2, \ldots J_m^2]^{\mathrm{T}}$ be the vectors[2] containing the values of the

---

[2] the superscript $^{\mathrm{T}}$ denotes the transpose.

m objectives of 2 design vectors $\mathbf{p^1}$ and $\mathbf{p^2}$.

Dominance (assuming maximization): $\mathbf{J^1}$ (weakly) dominates $\mathbf{J^2}$ if and only if:

$$J_i^1 \geq J_i^2 \quad \forall i = 1 \ldots m, \quad \text{and there exists at least one } i \text{ such that} J_i^1 > J_i^2. \tag{3.13}$$

This concept enables the comparison of *vectors* of objectives. Design $\mathbf{p^1}$ is then said to be better than $\mathbf{p^2}$ if $\mathbf{J^1}$ dominates $\mathbf{J^2}$.

It is possible to have $\mathbf{J^1}$ and $\mathbf{J^2}$ such that neither dominate the other. Take for example the vectors $\mathbf{J^1} = [1 \quad 0]^{\mathrm{T}}$ and $\mathbf{J^2} = [0 \quad 1]^{\mathrm{T}}$. The first objective value is greater for $\mathbf{J^1}$, but the opposite is true for the second objective, so that neither dominates the other. Consider now a set of objective vectors $S = \{\mathbf{J^1}, \ldots \mathbf{J^n}\}$. A subset of those vectors will not be dominated by any other. These vectors are said to be *non-dominated*. In the non-dominated set, an improvement in one objective can only be achieved at the expense of at least one other objective. The non-dominated set facilitates trade-offs between the m objectives. For a multi-objective optimization problem, the *Pareto front* is the set of all non-dominated designs, i.e., it extends the concept of optimality to the multi-objective case. The goal of the MOGA will be to generate an approximate Pareto front (APF) composed of non-dominated designs.

In the set S each individual can also be assigned a ranking equal to the number of vectors that dominate it, plus 1: this is the *Pareto ranking*. Non-dominated individuals will therefore receive a rank of 1, while the rank of dominated vectors will be 2 or more. The concept of Pareto-ranking will be used in the MOGA to compare and rank different designs.

On the *objectives graph* the n individuals can be plotted as a function of their m objectives. This is illustrated on Figure 3-7 where 2 objectives are maximized. The non-dominated points are linked by a line. The *utopia point* is the point for which every objective is maximum. This point is typically infeasible, but as designs improve during the optimization they tend to come closer to it.

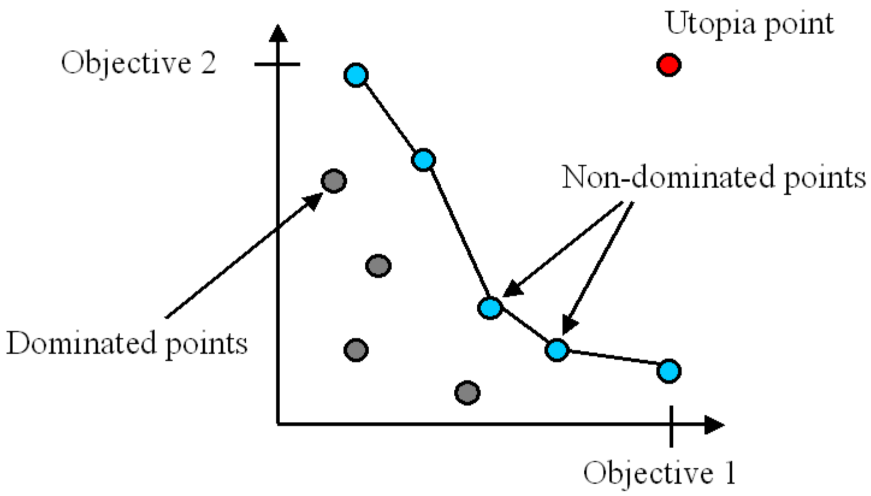

Figure 3-7: Example of an objectives graph with 2 objectives to be maximized. The utopia point is in the upper right corner, and non-dominated points (APF) are linked by a line.

3.3.3 Algorithm Description

The output of the MOGA will be a set of non-dominated layouts from which the user can choose. This is valuable information because it expresses the trade-off between the objectives. For example it quantifies the amount of additional coverage that can be attained by deploying an additional sensor, and the user, depending on his or her preference, can decide whether it is worth doing or not.

Because the MOGA is stochastic, it has a lesser tendency to get stuck in local optima than other algorithms. In particular, the non-dominated set will typically contain different network topologies: some with all sensors clustered around the HECN, some in a hub-and-spoke shape, etc. The layouts generated by the MOGA will typically be “rough” designs, and local search methods can then be used to refine them by fine-tuning the position of each sensor.

Below is the description of the GA implemented for Phase I. While the main operators are classic GA operators, it should be kept in mind that this is a specific implementation of a MOGA. For more information on different variants of GAs, the reader is referred to the literature [27,35,36,43].

A design vector $\mathbf{p}$ is called a *chromosome*[3]. Each chromosome is composed of *genes* that contain the sensors coordinates. At each step of the GA (called a *generation*), a *population* of N_{pop} chromosomes is maintained. Each chromosome is then passed through a series of genetic operators (*crossover*, *mutation*, and *selection*) to produce the next generation. The population is expected to improve as generations pass.

Crossover: Chromosomes are paired two by two at random. Each such pair is then *mated* according to the following procedure. The 2 chromosomes are first cut in 2 at a common *crossover point* chosen at random. The 4 chromosome pieces are then recombined to form 2 new chromosomes, or *children*.

Mutation: Each gene in each children is mutated with probability p_{mutation}. If a gene is selected for mutation, a new value for the coordinate is chosen at random.

Fitness Assignment: After mutation, the objectives of the children are computed and the Pareto ranking of each child and parent is computed. Each of these $2N_{\text{pop}}$ chromosomes is then assigned a *fitness* equal to the inverse of its Pareto ranking [35]. Chromosomes with higher fitness thus represent better designs.

Selection: Among the pool of parent and children chromosomes, the N_{pop} with highest fitness are selected. This enables the non-dominated individuals to be passed on to the next generation, so that the population keeps improving while maintaining its diversity.

The process is repeated on the new generation, unless the maximum number of generations is reached. This is summarized in Figure 3-8.

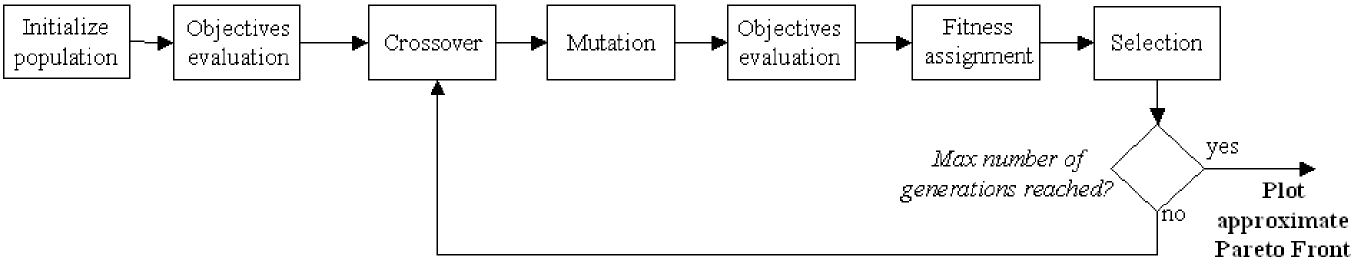

Figure 3-8: Block diagram summarizing the Genetic Algorithm procedure.

A final note regarding the MOGA for CS2: in this case the number of sensors is itself a

---

[3] Since the design variables are the coordinates of the sensors and are therefore homogenous, there is no need for encoding as is traditionally done in GAs.

variable. While the chromosomes are of constant length, equal to the maximum number of sensors, the actual number of sensors per chromosome is allowed to vary dynamically [45]. In particular, a sensor is present at the $(2i-1, 2i)$ genes (for $i = 1 \dots n$) if this pair is different from $(0, 0)$. When the initial population is generated, the number of sensors present in each chromosome (as well as the sensor coordinates) are chosen at random. Likewise, the mutation operator can affect both the coordinate of a sensor and whether or not the sensor is present.

3.4 MOGA Results for Phase I

3.4.1 Results for CS1

Figure 3-9 illustrates the results of MOGA on CS1 for 10 sensors, with $R_{\text{COMM}} = R_{\text{Sensing}} = 2$. The MOGA was run for 250 generations, with $N_{\text{pop}} = 120$ and a mutation rate of 0.1. On the objectives graph all designs generated during the 250 generations are plotted according to their objectives. The color code shows how the designs tend to evolve toward the utopia point in the upper-right corner as the generation number increases. Note that the lifetime can only take a finite number of values, so the objectives graph is composed of horizontal lines. Each of those lines corresponds to a value of lifetime, which in turn corresponds to a certain network topology, that is linked (typically) to the number of sensors directly connected to the HECN. The APF contains the non-dominated designs, which for convenience have been joined by a line. The designer needs only to look at this finite set of designs to make his decision about what layout to deploy, since all the other layouts are dominated by those (i.e., there exists at least one design on the APF for which both objectives are better). It can be seen on the APF that higher coverage is only achieved at the expense of a smaller lifetime. To illustrate these different designs, two of them are plotted on Figure 3-10, one with 5 sensors at the HECN (5-spokes design), the other with 3 (3-spokes design). It can be seen that for a 6% improvement in coverage (from 0.4647 to 0.4984), the lifetime drops 37.5% (from 0.4 to 0.25). The user may therefore decide that the marginal increase in coverage is not worth the decrease in lifetime, and may choose to go with the 5-spokes design. This insight would not have been available if only the coverage had been maximized. So not only does the MOGA provide diverse solutions for this problem, but it also helps inform the user about the available trade-off between objectives.

Note that the designs shown on Figure 3-10 could be refined by performing a local search around the nominal designs. An algorithm to do so will be presented in the section on Phase II.

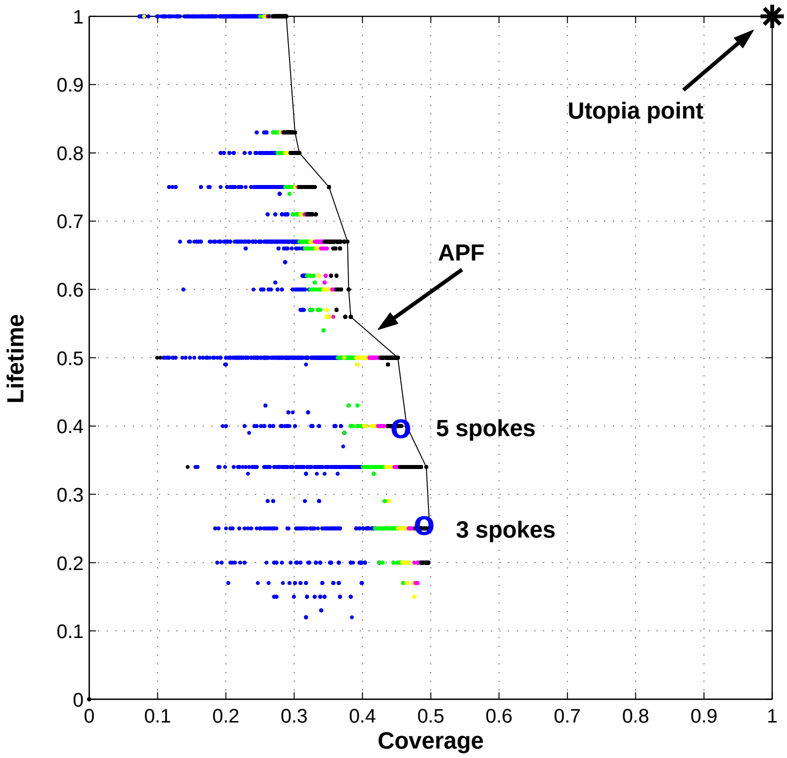

Figure 3-9: Objectives graph for CS1 with 10 sensors and $R_{\text{COMM}} = R_{\text{Sensing}} = 2$. All designs generated during the 250 generations are plotted according to the following color code. *Blue*: generations 1-50; *Green*: generations 51-100; *Yellow*: generations 101-150; *Magenta*: generations 151-200; *Black*: generations 201-250

3.4.2 Analysis of the APF

In the previous example the layout with largest coverage was a 3-spoke design. How would this change if the sensing radius was decreased, while the communication range remained constant? Intuitively, more sensors could be clustered around the HECN without having

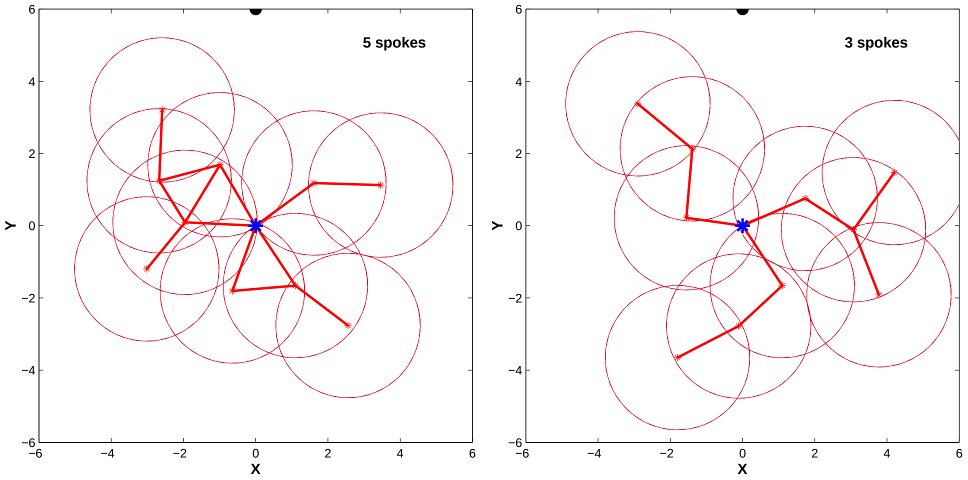

Figure 3-10: Example of non-dominated designs given by the MOGA on Fig. 3-9. One is with 5 spokes, the other with 3.

the sensing disks overlap, thus providing the same coverage with a greater lifetime: the 3-spoke design would no longer be part of the APF. In this section we briefly investigate the influence of the ratio $\rho = R_{\text{Sensing}}/R_{\text{COMM}}$ on the non-dominated designs, focusing on the layouts at the two ends of the APF.

The layout with best lifetime at the upper end of the APF is always the same, irrespective of the value of ρ. In this case all the sensors are clustered around the HECN, so that all the sensors' energy can be devoted to transmitting their own data packets, yielding a network lifetime of 1. This is true irrespective of the value of the sensing radius.

Let us then turn our attention to the design with best coverage at the bottom end of the APF. On Figure 3-11 are shown the APF designs with largest coverage for $\rho = 0.5$, 1, and 2. When $\rho = 0.5$ the sensors are arranged in a "beehive" fashion around the HECN, whereas when it is equal to 1 and 2 the sensors form a hub-and-spoke configuration, with respectively 3 and 2 spokes stemming from the HECN. It is interesting to note that in the first case hub-and-spoke layouts are not part of the APF. This is in accordance with our previous observation, since the same coverage can be obtained with a larger lifetime.

For a given number of sensors, the maximum coverage will be obtained when, if possible, there is no overlap between the sensing disks. Viewing the sensors as marbles of radius R_{Sensing}, the tightest packing is obtained when each marble touches 6 neighbors (except for the peripheral ones). This beehive configuration ensures a maximum coverage,

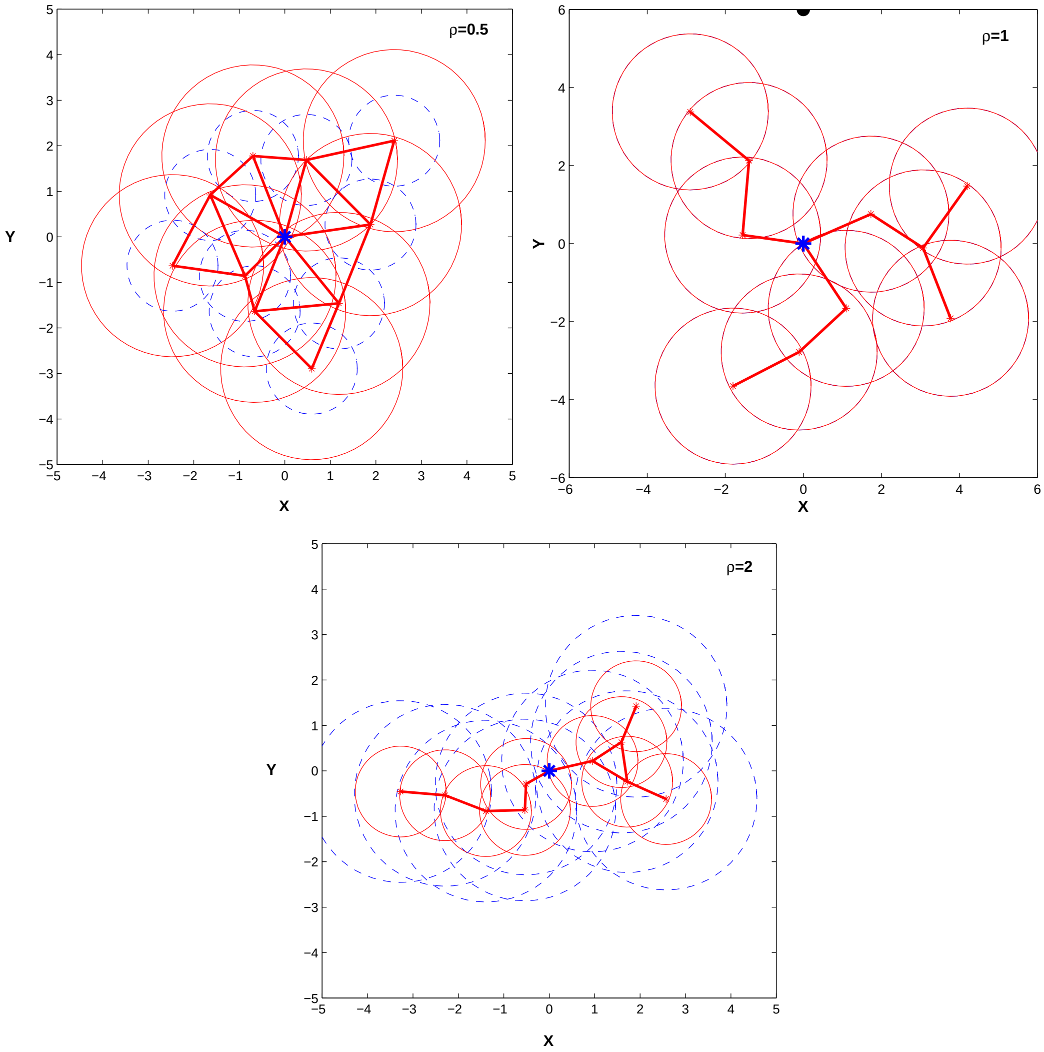

Figure 3-11: Layouts with best coverage for $\rho = 0.5$, 1, and 2. Note how the number of sensors at the HECN decreases as ρ increases.

while providing the maximum number of neighbors for each sensor, so that the lifetime is maximized. This is shown on Figure 3-12 (note the similarities between this design and that of Figure 3-11 with $\rho = 0.5$). Thus as long as $R_{\text{COMM}} \geq 2R_{\text{Sensing}}$, or $\rho \leq 0.5$, the configuration where the sensors are packed like marbles will be the bottom point of the APF.

Let us now assume that $R_{\text{COMM}} < 2R_{\text{Sensing}}$. If the sensors remained in this "beehive" configuration they would be unable to communicate with each other since they would all be outside their neighbors' communication range. Overlap between the sensing disks therefore becomes necessary, which explains the change in design observed for $\rho = 1$ and 2. Instead

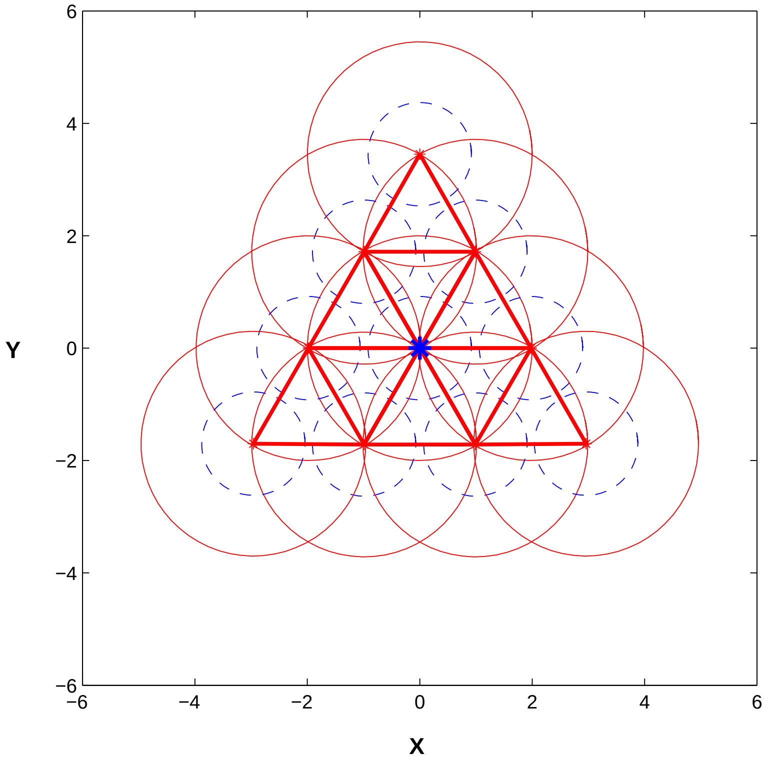

Figure 3-12: Beehive configuration where all sensors are tightly packed together (here $\rho = 0.5$).

of having 7 sensors directly connected to the HECN, these respectively have 3 and 2 sensors connected to the HECN.

As ρ increases it also seems that the number of spokes stemming from the HECN decreases. Given $\rho > 0.5$, let us perform a numerical study to approximate the number of spokes stemming from the HECN in the APF design with best coverage. For this purpose the area is assumed to be the infinite plane. We also assume that given a total number of n sensors, the layout is completely determined by the number n' of sensors directly connected to the HECN. This design is constructed as follows:

- Evenly space n' sensors around the HECN;
- Assign the remaining $n - n'$ sensors evenly between the n' sensors already in place, and place them on a straight line stemming outward.

The resulting designs are star-shaped hub-and-spoke layouts. This is an approximation since the true optimal design may have the spokes of the hub-and-spoke connect with each other (this can result in a slightly larger lifetime, as in Figure 3-2).

Following this rule we can determine the optimum number of sensors directly connected to the HECN for different values of ρ. When $\rho = 0.5$, 6 sensors can be placed at the HECN without overlap between the sensing disks. As ρ increases, fewer sensors can be placed at the HECN if overlap is to be avoided. However, overlap might be beneficial at the HECN in order to have less overlap in the spokes, as illustrated in Figure 3-13 for 3 sensors and $\rho = 1$. In this example the overlap of 3 sensors at the HECN (right) is lower than that between 2 sensors on a spoke (left). In general overlap at the HECN will therefore be allowed if it results in more overall coverage.

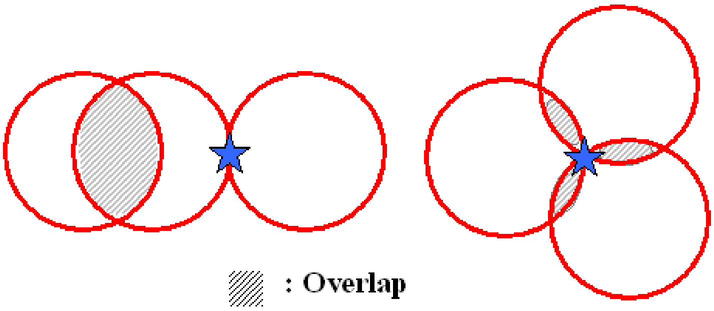

Figure 3-13: In this example a layout with 2 spokes (left) generates more overlap than one with 3 spokes (right). Although no overlap is present at the HECN, more is incurred by having 2 sensors connect on a spoke.

Let us start by considering a layout with n sensors at the HECN. Let $C_{\text{HECN}}(n)$ be the resulting coverage. There are two possibilities for the placement of an additional sensor:

- It can be added to the HECN. In that case the $n+1$ sensors are equally spaced around it, and the resulting coverage is $C_{\text{HECN}}(n+1)$;
- It can be added to a spoke. Let C_{spoke} be the corresponding additional coverage, so that the total coverage is $C_{\text{HECN}}(n) + C_{spoke}$.

The option yielding the largest coverage is chosen.

- If $C_{\text{HECN}}(n+1) \geq C_{\text{HECN}}(n) + C_{spoke}$, add the sensor to the HECN;
- Else add it to a spoke.

Note that once the second option is optimal, then all the remaining sensors will be placed on spokes: placing additional sensors at the HECN provides a decreasing amount of additional coverage as the sensors increasingly overlap. In Figure 3-13 additional sensors will be added to the spokes.

This analysis was performed numerically for several values of ρ. The optimal number of sensors at the HECN is plotted as a function of ρ on Figure 3-14. The number of sensors at the HECN starts from 6 when $\rho = 0.5$, and then decreases rapidly to 3. When the sensing radius is about twice the communication radius, the optimal layout only has 2 spokes stemming from the HECN and the network essentially degenerates into a linear array. These results, valid for any number of sensors, confirm the optimality of the layouts obtained by the MOGA. Previous results from the MOGA results indeed showed a 3-spoke design when $\rho = 1$, and a 2-spoke design for $\rho = 2$.

This result is important because it shows the influence of the sensor technology on the optimal layout. Values of ρ will indeed largely vary depending on whether a seismic or a visual sensor is employed. For example, the seismic sensor described in Section 2.1.1 had $R_{\text{COMM}} = 300m$ and $R_{\text{Sensing}} = 1000m$ for vehicles, yielding $\rho = 3.3$. The configuration of the WSN with largest coverage would therefore be a linear array. However a visual sensor can be expected to yield a ρ smaller than 1, leading to a design with at least 3 spokes.

Network designers cannot therefore have a one-size-fits-all approach to WSN design, where the same configuration would be used for all missions. As has just been demonstrated, even when the area and the mission objectives are the same, different sensors will yield different optimal designs. An automated planner such as the MOGA presented in this section is therefore necessary to optimize the use of resources. Section 3.5 will further investigate benefits of using the MOGA.

3.4.3 Results for CS2

The MOGA was run for a maximum number of 12 sensors with $R_{\text{Sensing}} = 1.8\ R_{\text{COMM}} = 4$ (i.e., the sensing range is about half the communication range). The MOGA had 300 generations, $N_{\text{pop}} = 100$, and a mutation rate of 0.1. The objectives graph is shown in

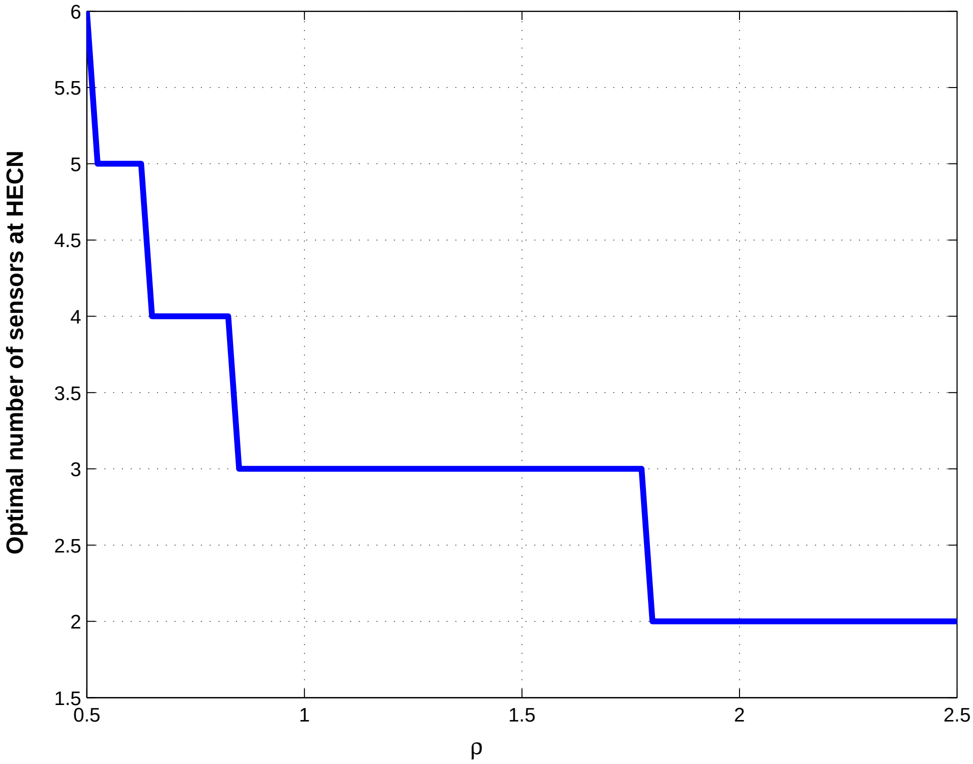

Figure 3-14: Optimal number of sensors at the HECN for CS1 as a function of ρ.

Figure 3-15, where the non-dominated designs populating the APF are shown by circles. The utopia point is where coverage and survivability are 1, and the number of sensors is 0. Figure 3-16 shows non-dominated designs with 9, 7, and 5 sensors. This illustrates the trade-off between detection, survivability, and number of sensors. This optimization took 24 minutes to complete on a Pentium 4 processor running at 1.8GHz.

These results confirm the intuition about this problem. Higher survivability is achieved by placing the sensors far away from the facility and the roads (e.g., the case with 9 sensors), but to maintain full detection capability of the movement more sensors are needed (compare the designs with 9 and 5 sensors). The cost for using fewer sensors and requiring the same full coverage is a lower survivability: it drops by 39% from that with 9 sensors ($\mathsf{Survivability} = 0.73$) to that with 5 sensors ($\mathsf{Survivability} = 0.44$). The layouts selected in Figure 3-16 all have a detection value close to 1. However it may be the case that the survivability is of great concern for a particular mission (e.g. if the WSN has to stay on site for a long time), so that some detection capability can be given up in order to gain in

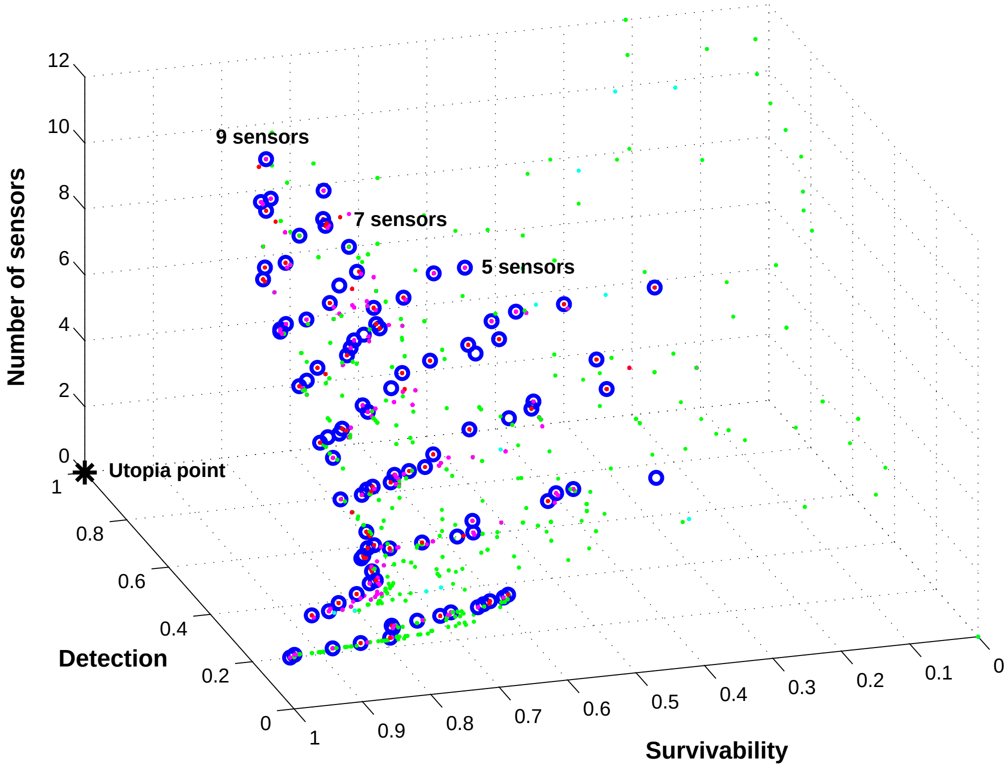

Figure 3-15: Objectives graph for CS2 obtained with a maximum of 12 sensors, $R_{\text{Sensing}} = 1.8$ and $R_{\text{COMM}} = 4$. Non-dominated designs are shown as circles, and 3 of them with 9, 7, and 5 sensors are plotted on Fig. 3-16

survivability. In particular we also note that the APF layouts with the largest survivability have the same structure as the design with 9 sensors in Figure 3-16, with the necessary number of sensors removed, starting from the lower left corner.

We finally note that this MOGA was tested on 2 other case scenarios, presented in Appendix A.

To summarize, a total of 5 different objectives were tested on 4 different case scenarios, each with different combinations of objectives. In all these examples the exact same MOGA was used to obtain different network designs from which the user can choose. The flexibility of the MOGA is evident from the variety of cases tested. The simple case scenarios treated here can serve as building blocks (or primitives) of more complex missions, while the same algorithm can be used.

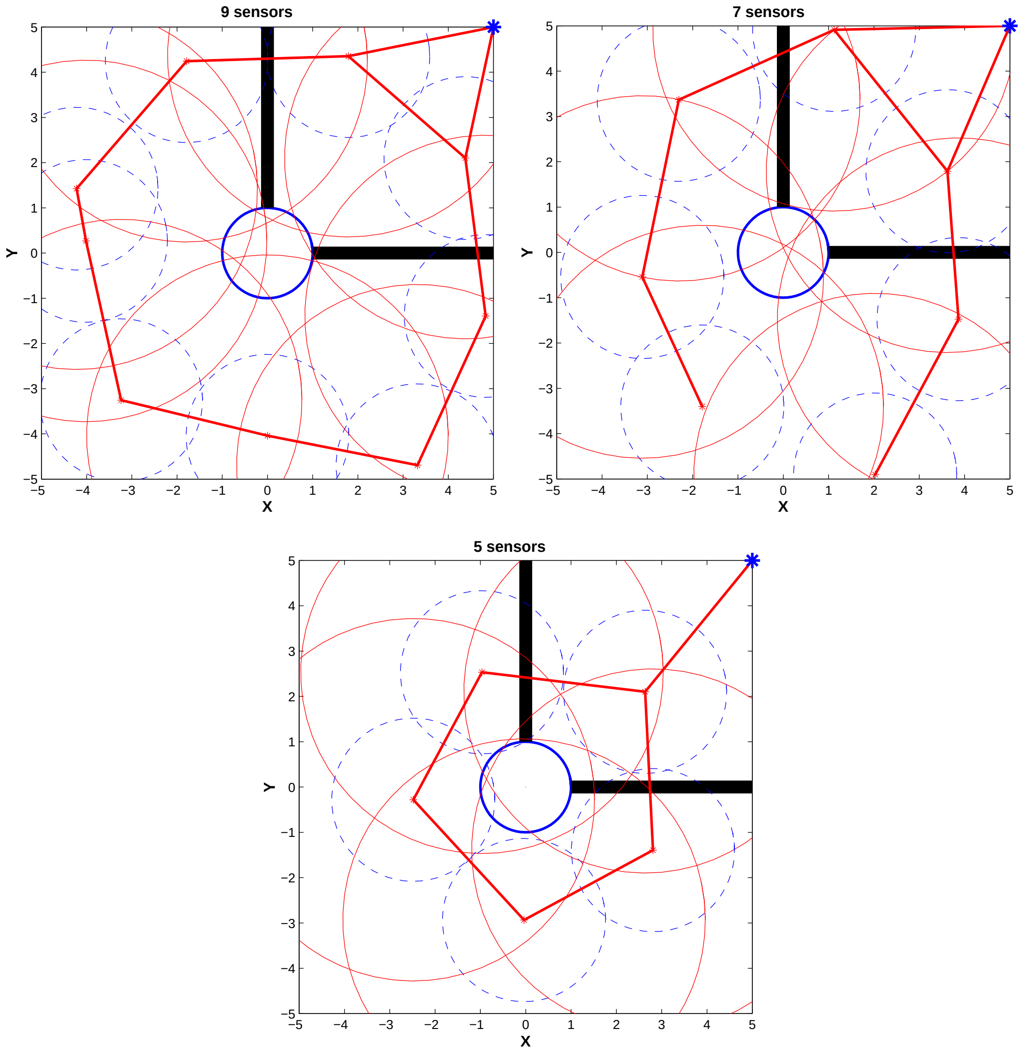

Figure 3-16: Non-dominated designs with 9, 7, and 5 sensors for CS2.

3.5 Benefits of Phase I Planning

Some important reasons for using MOGA to plan the WSN have already been outlined in the previous sections:

- The MOGA provides the user with the trade-off between objectives.
- The optimal network designs will depend on the sensors characteristics, so there is no one-size-fits-all network layout optimal in all cases.

Much of the WSN literature assumes that the sensors will be randomly deployed. In this section we perform a brief numerical comparison between the performance of the MOGA designs and that of designs obtained without planning, when the sensors are simply dropped at random.

Let us first consider the objectives graph obtained for CS1 in Figure 3-9. For a given number of sensors, the coverage and lifetime are averaged over 100 random deployments of the WSN. This is done for $n = 5, 10, \ldots, 70$ sensors. The corresponding averaged objectives values are plotted as a red line on the objectives graph of Figure 3-17. It can be seen that over 25 sensors are needed to achieve on average the same coverage than the value obtained by 10 sensors deployed according to the MOGA. Moreover, the only parameter one can control when dropping sensors at random is the number of sensors, which only gives access to a very limited portion of the objectives space. The MOGA however provides the user with much more control over where the WSN should be in the objectives space.

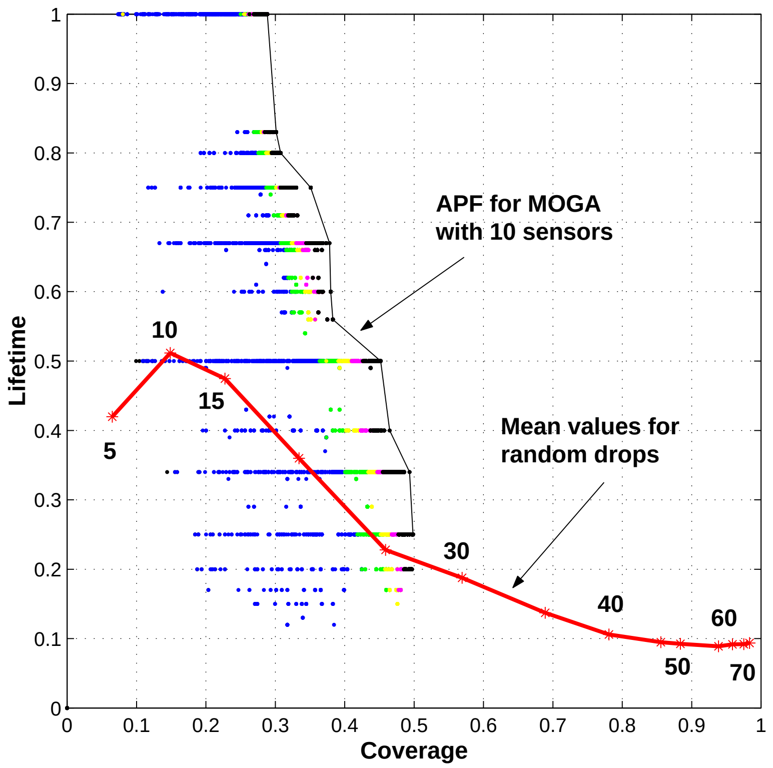

Figure 3-17: Objectives graph for CS1 with the averaged objectives obtained by randomly dropping $5, 10, \ldots, 70$ sensors.

We then consider the results from CS2, in particular the designs with 9 sensors (called D_9) and with 5 sensors (called D_5) of Figure 3-16. Just as in the previous case, these objectives are compared to those obtained via random deployments, with a varying number of sensors. The resulting objectives are plotted as a function of the number of sensors in Figure 3-18. The objectives of D_9 and D_5 are also plotted on each graph for comparison. It can be seen that to obtain designs with an average detection close to 1, about 30 sensors are needed if placed randomly, whereas both designs with 5 and 9 sensors provide a detection of 1. As far as survivability is concerned, it peaks for 8 sensors (at about 0.3) and decreases on average as the number of sensors increases. In this case deploying more sensors actually hurts the survivability since the probability of a sensor falling close to a road or the facility increases if more are dropped. On Figure 3-19 the average value of detection and survivability for each number of sensor is plotted alongside the objectives graph obtained by MOGA (the graph is projected on the 2D plane of detection and survivability). It is clear from this plot that, just like for CS1, the portion of the objectives space reached by random designs is poor.

In summary these brief examples illustrate the need for a planner. Without it most desirable regions of the design space can never be reached, even by deploying an increasing number of the resources. In fact deploying more sensors may even result in a decreases of some objective value.

These sections demonstrated the need for planning as well as validated our solution approach with MOGA. The MOGA was shown to be a flexible algorithm well-suited for the type of complex problems typically encountered during Phase I. It was shown that using a planning algorithm is necessary as it leads to significant saving of resources and achieves designs otherwise impossible, e.g. via random deployment.

3.6 Phase II: Algorithm and Studies

Phase I provides the planner with a nominal WSN design. The necessary resources are then loaded into the UAV, and the aircraft flies to the deployment zone. Referring back to the 3-phased strategy of Section 2.4, the system then enters Phase II, where the UAV starts deploying the sensors on the ground, according to the Phase I design. The following assumptions are made about the deployment process:

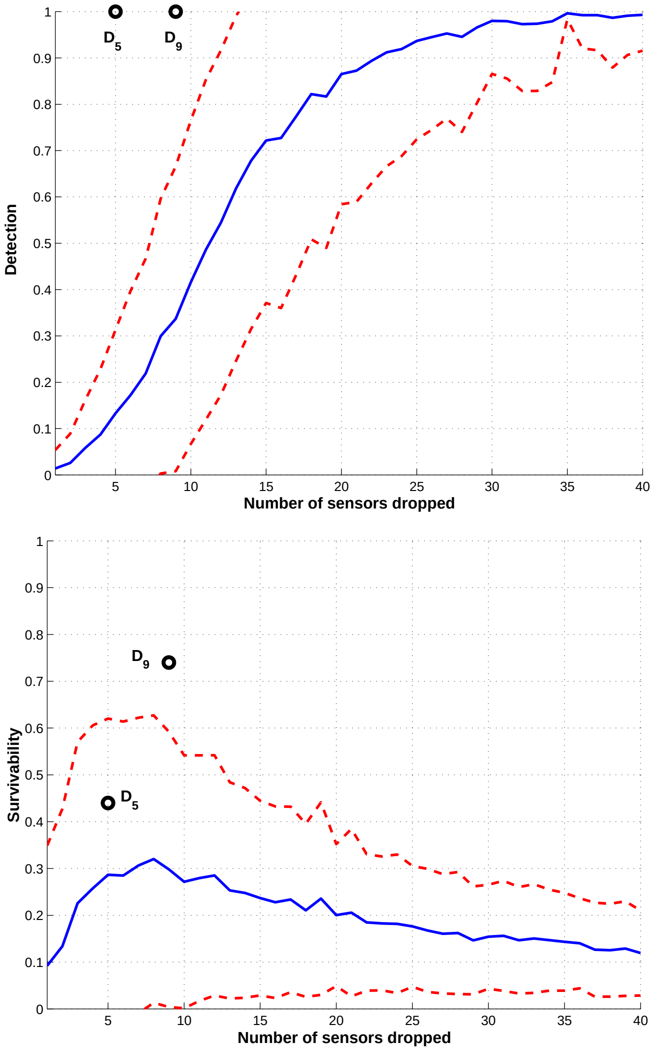

Figure 3-18: Average value of the detection and survivability objective as a function of the number of sensors dropped. The 1-σ envelop is shown in dashed, and the objectives values for D_5 and D_9 are shown for comparison.

- The UAV can make several passes over the mission area;
- At each pass it can deploy one or more sensors;
- At each pass it has a probability of being lost;
- The UAV has the ability of knowing the location of the sensors once they are on the ground (e.g. through broadcast of their GPS positions).

The task of Phase II will therefore be to decide how many sensors to drop at each pass, and where each of these sensors should be deployed. As mentioned before, the drops are

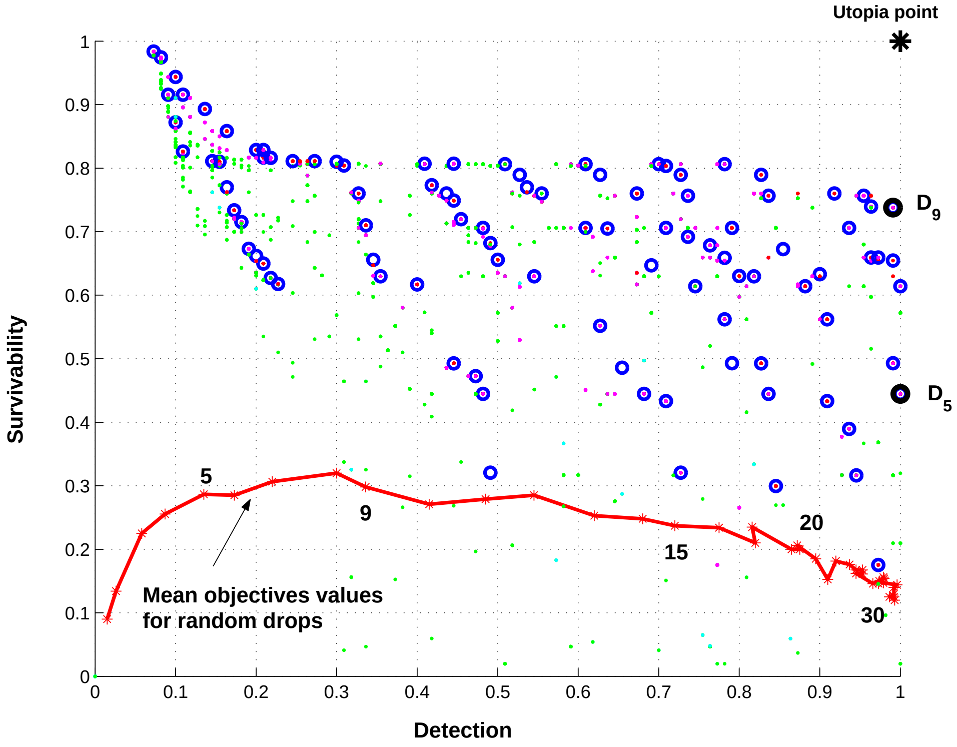

Figure 3-19: Average value of detection and survivability when sensors are placed randomly, plotted alongside the objectives graph obtained by MOGA.

guided so that sensors can be deployed at the desired location, given some inaccuracy due to the drop vehicle modeled as a zero-mean Gaussian with variance σ^2_{DV}.

This is a complex problem. Decisions must be made not only about drop sequencing, but also about the location of the drops, while taking into account the information from previous drops. In this section we will provide a simple approach to this problem, and analyze it through numerical simulations. This study will demonstrate the clear need for planning Phase II.

3.6.1 Algorithm Description

Phase II starts where Phase I left off. The nominal Phase I design would be chosen on the APF by the user, according to his or her preferences among several objectives. We restrict our study to CS1, where coverage and lifetime must be maximized. We also require that the topology of the network on the ground, i.e. the communication links between sensors, remain the same as on the Phase I design. In this way the value of the lifetime will be the same as that planned. Phase II will therefore aim at maximizing the expected value of the coverage. If CS2 was considered, then the expected survivability and detection must be maximized (assuming the number of sensors is constant). A weighted average of those two may be a good way to address this case within the framework adopted below.

It is convenient to decompose Phase II into two separate subproblems. Given a nominal layout of the sensors, with some possibly already deployed on the ground, we do the following:

- Choose a *drop sequence*, i.e. which sensors should be dropped at each pass;
- Given this drop sequence, *refine* the location of the sensors to be dropped at the next pass to account for the drop vehicle inaccuracy and the actual locations of the sensors already on the ground.

Let us first define a few variables. Let $\mathbf{p}_0$ be the nominal Phase I design, $S = \{S(1), \ldots, S(q)\}$ is the drop sequence where $S(i)$ contains the indices of the sensors to be dropped at the i^{th} drop and q is the total number of drops, $\mathbf{p}(i)$ are the coordinates of the sensors to be dropped on the i^{th} pass, and $\mathbf{p}_{\text{GND}}$ are the coordinates of the sensors already on the ground. Also, the probability that the UAV may be lost during a pass is denoted by p_{loss}. Given a drop sequence S, the refining routine of the second task of Phase II is denoted $\mathsf{REFINE}(\mathbf{p}_0, \mathbf{p}_{\text{GND}}, S(i))$ for the i^{th} pass, and it returns $\mathbf{p}(i)$.

The goal is to choose the sequence S with maximum expected coverage. Unfortunately there exists no closed-form expression for the expected coverage, so it must be approximated through Monte Carlo simulation. For a given drop sequence S, the expected coverage $\mathbb{E}\{\mathsf{Coverage}(S)\}$ is calculated as follows:

- Initialize $\mathsf{cov} = 0$;
- Repeat the following n_{MC} times:
    1. Initialize $\mathbf{p}_{\text{GND}} = \emptyset$.
    2. For drop $i = 1 \ldots q$, do:

    - Get $\mathbf{p}(i) = \mathsf{REFINE}(\mathbf{p}_0, \mathbf{p}_{\text{GND}}, S(i))$;
    - Sample a random number uniformly between 0 and 1. If it is below p_{loss} the UAV is lost, go to (3);
    - Simulate dropping $\mathbf{p}(i)$ with inaccuracy according to $\mathcal{N}(0, \sigma^2_{\text{DV}})$;
    - Add the resulting sensors positions to $\mathbf{p}_{\text{GND}}$.

  3. Calculate the coverage $\mathsf{Coverage}(\mathbf{p}_{\text{GND}})$;

  4. $\mathsf{cov} \leftarrow \mathsf{cov} + \mathsf{Coverage}(\mathbf{p}_{\text{GND}})$.

- $\mathbb{E}\{\mathsf{Coverage}(S)\} = \mathsf{cov}/n_{\text{MC}}$,

where n_{MC} is the number of Monte Carlo simulations used to calculated the expected coverage.

Not only is there no closed-form expression for the expected coverage, but the number of possible drop sequences is also exponentially large. For 3 sensors, the possible sequences are:

- Drop them all together (1 combination);
- Drop them in 2 groups ($2 \times C_3^2$ combinations);
- Drop them one by one: ($3!$ combinations).

The total number of drop sequences is 13 in this case, and for n sensors this number will grow faster than $n!$. This prohibits enumerating all drop sequences and calculating the expected coverage for each.

Drop Sequences

In the following only a subset of sequences will be evaluated, in order to illustrate the vast differences in terms of resulting expected coverage.

In particular in sequence S_1 all the sensors are dropped at the same time, while in S_2 they are dropped one by one. The sensors can also be grouped according to the number of hops h required for information to reach the HECN. This forms the basis of 2 other drop sequences. In S_3 the sensors are dropped from the HECN outward, that is the sensors with $h = 1$ are dropped first, followed by those with $h = 2$, etc. In S_4 the opposite is performed, where sensors with higher h are dropped first. Finally sensors can be grouped so that in each group, no sensor has a communication link in common. In S_5 the sensors are dropped according to those groups, so that in each group sensors are "independent" of one another.

Refining Algorithm

We first concentrate on the problem of adjusting the location of a single sensor (x_i, y_i). We assume that all the other sensors are fixed, and we want to find the location that will maximize the expected coverage. A small area around its nominal location is discretized, and for each of these candidate positions, the expected coverage is computed, for example through Monte Carlo simulation. Then the location with maximum expected coverage is chosen.

We then successively apply a routine called REFINE to every sensor in the drop, so that all their positions are adjusted. Doing so successively for each sensor presupposes that the sensors in a single drop are independent of one another. This is not true in general because sensors with overlapping sensing disks or interconnected by a communication link are not independent, so that moving one sensor will typically influence the rest of the network. To be more rigorous we should adjust the positions of all the sensors in a drop simultaneously, but this would make the problem intractable.

This routine is summarized below. The nominal Phase I design is denoted by $\mathbf{p}_0$, and the position of the sensors in the set S are to be refined. There may also be sensors already on the ground, given by $\mathbf{p}_{\text{GND}}$.

REFINE$(\mathbf{p}_0, \mathbf{p}_{\text{GND}}, S)$

- Set $\mathbf{p} = \mathbf{p}_0$;

- For each sensor i in S, do:

    1. Discretize the area around $\mathbf{p}(i)$ (the nominal location of sensor i) into q points $\{\tilde{\mathbf{p}}_1(i), \ldots, \tilde{\mathbf{p}}_q(i)\}$;

2. For each of these points $\tilde{\mathbf{p}}_j(i)$, repeat m_{MC} times:
    - Simulate dropping a sensor at $\tilde{\mathbf{p}}_j(i)$ with inaccuracy according to $\mathcal{N}(0, \sigma_{\text{DV}}^2)$;
    - In $\mathbf{p}$ replace the i^{th} nominal sensor position by this value;
    - Calculate the corresponding coverage;
3. Let $\tilde{\mathbf{p}}_k(i)$ be the points yielding the largest average coverage;
4. Set $\mathbf{p}(i) = \tilde{\mathbf{p}}_k(i)$.

- The refined design is given by $\mathbf{p}$.

This method can actually be applied to all the sensors in the network, so as to refine the whole design. This is desirable especially for the design given by Phase I, which tends to be rough as mentioned before. Running it through REFINE will fine-tune the positions of all the sensors, while accounting for the inaccuracy in drop vehicles (something to which Phase I is oblivious). Consider an example where Phase I gave a layout with 5 sensors and 3 spokes. Three layouts (Figure 3-20) are compared. In the first, the sensors are brought close to one another (within 60% of their communication range): this is a conservative option. In the second one, the sensors are stretched as far as their communication range will allow them. In the third one, REFINE is used, given that $\sigma_{\text{DV}} = 0.2$. We assume that all the sensors are dropped at the same time, and the expected coverage of the network is calculated through Monte Carlo simulation. The expected coverage values are, respectively, 0.23, 0.13, and 0.27, while the coverage values obtained through 100 simulations are sorted and plotted on Figure 3-21. The stretched configuration yields the poorest coverage because, due to the drop inaccuracy, sensors often end up being disconnected from the HECN. The coverage is much better for the more conservative design, but it can still be improved (by 17%) by running REFINE. This is explained by the fact that the REFINE routine explicitly takes the value of σ_{DV} into account, so that the refined design is neither too risky nor too conservative.

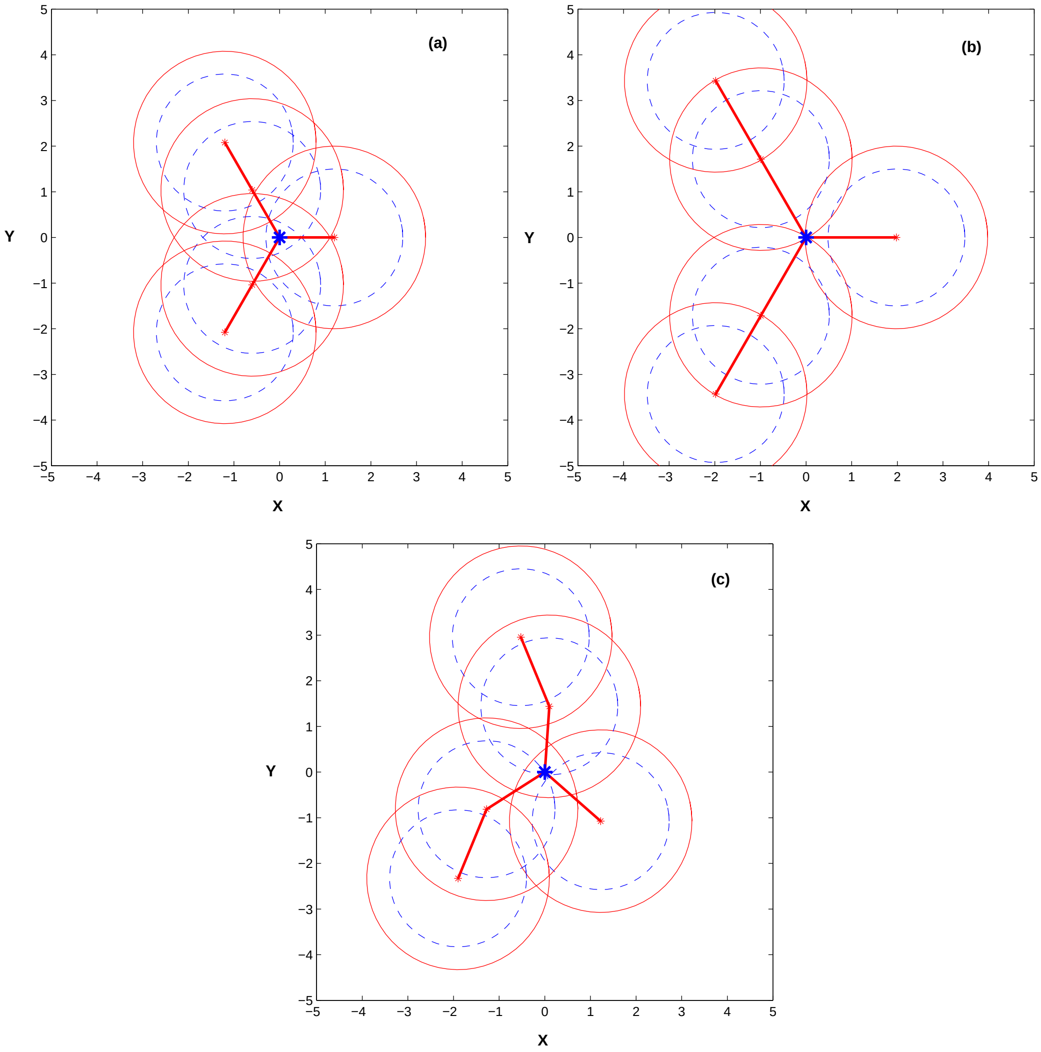

Figure 3-20: (a) Conservative design, (b) stretched design, (c) design fine-tuned using REFINE.

3.6.2 Benefits of Phase II Planning

Let us now consider the whole Phase II process. Two different designs are used in this study to show the benefits of Phase II planning. Design D_9 is a 3-spoke design with 9 sensors, as shown on Figure 3-22, while design D_{19} is a mesh of 19 sensors. For both designs, the 5 drop sequences S_1 through S_5 are tested and the resulting expected coverage are compared. The probability of UAV loss at each pass is set to $p_{\text{loss}} = 0.1$.

Results for D_9 are shown on Figure 3-23. This plot shows, for all 5 sequences, the coverage values of 100 drop simulations, sorted from worst (to the left) to best (to the

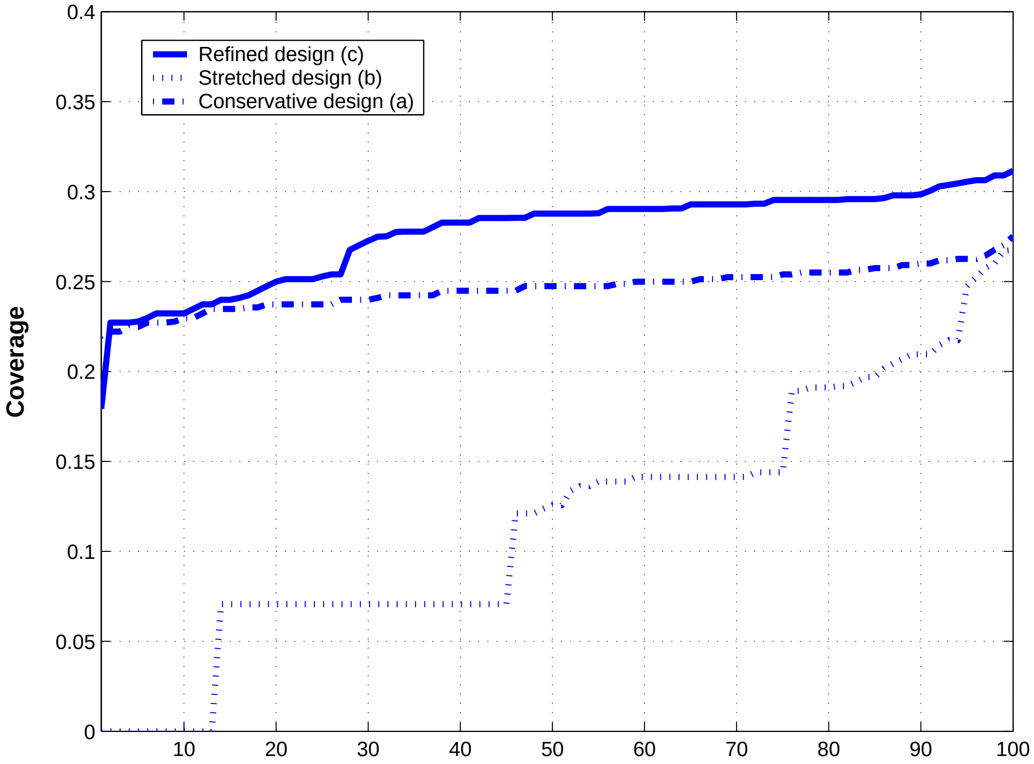

Figure 3-21: Sorted coverage values obtained through 100 Monte Carlo simulations for the 3 refined designs of Fig. 3-20.

right). It can be seen that the best sequence is S_3, when the sensors are dropped 3 by 3 outward. This strikes the best balance between the UAV survivability and the amount of feedback from previous passes. In comparison, dropping the sensors all at once (S_1) only yields mediocre results, due to the absence of feedback. The worst sequence is S_2, i.e. dropping the sensors one by one. In that case the UAV is usually lost before it had time to deploy all the sensors (the probability it survives the 9 passes is only $0.9^9 = 0.39$), so although the amount of feedback is maximum, it does not survive long enough to benefit from it.

Figure 3-24 depicts the results for D_{19} with 250 Monte Carlo simulations. This time the best sequence on average is S_1, where all the sensors are dropped at once. Since the sensors are closely clustered in the nominal Phase I design, they have a high probability of remaining connected after they are all dropped. Sequences S_3 and S_4 perform well also (actually better than S_1 in most cases, but by a small amount). However they suffer from multiple UAV passes, so that the coverage can be much worse than S_1 in some cases (left

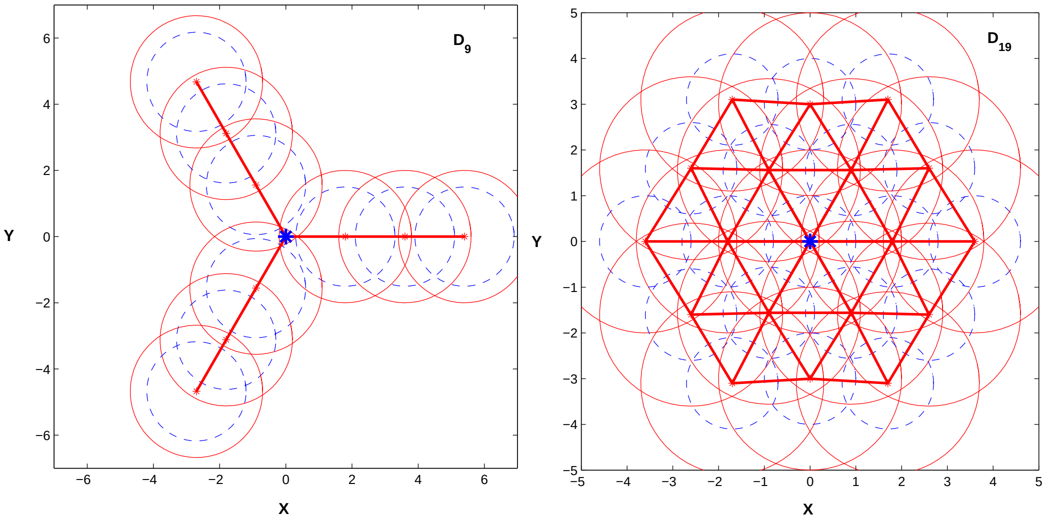

Figure 3-22: Design D_9 is a 3-spoke design with 9 sensors, while design D_{19} is a mesh of 19 sensors.

of the curves). Again, S_2 is the worst, so much so now that 19 passes have to be performed in order to fully deploy the entire network.

We draw two conclusions from this study. The first is that Phase II, can have a large impact on the performance of the WSN. With D_{19} for example, the expected coverage is over 4 times better when using sequence S_1 over S_2. *Choosing the wrong drop sequence can have a catastrophic effect on the performance of the WSN*. The second conclusion is that *the optimal drop sequence varies depending on the WSN design*. For a hub-and-spoke with 9 sensors, S_3 was the best, whereas for a mesh of 19 sensors it was better to drop all the sensors at once. There is therefore no one-size-fits-all drop sequence that will perform optimally in every case, and an algorithm like the one proposed for Phase II is a crucial piece of the planning algorithm.

3.7 Conclusion

In this chapter algorithms for Phase I and Phase II were proposed. A MOGA was shown to perform well on a variety of Phase I scenarios. Likewise, an approximate algorithm was described for Phase II, and shown to have a significant impact on the performance of the WSN once it reaches the ground.

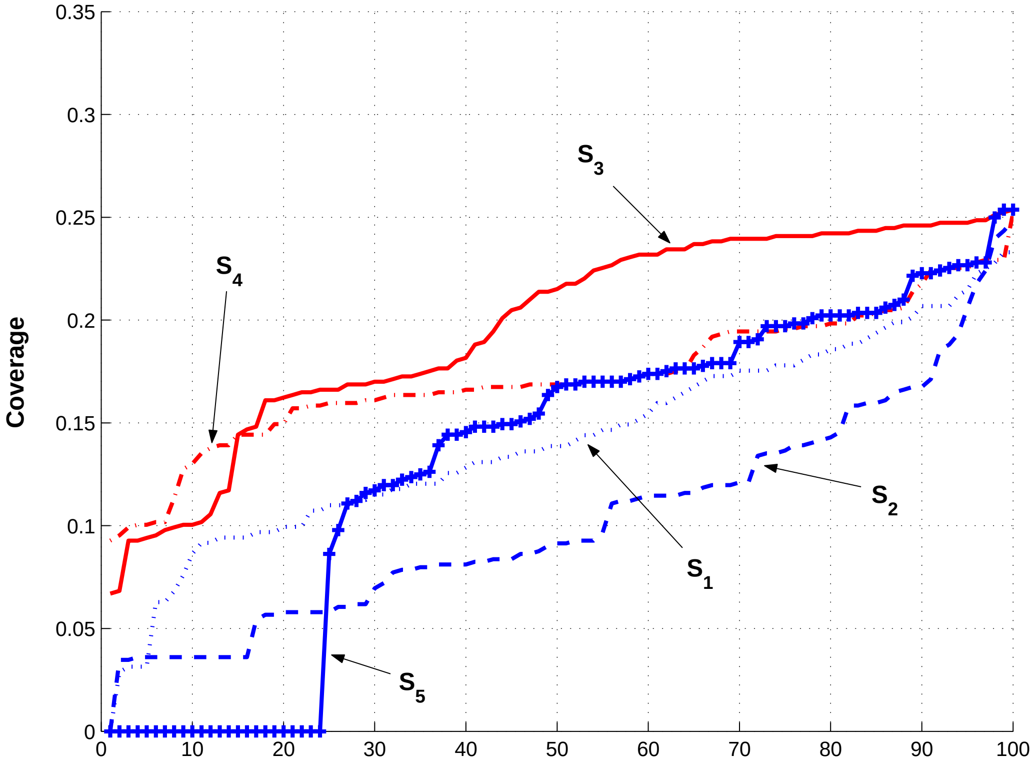

Figure 3-23: For design D_9, coverage values of 100 drop simulations, sorted from worst (to the left) to best (to the right) for all 5 sequences.

For Phase I, we showed that *careful planning of the WSN design leads to far superior solutions than random placement*. Likewise for Phase II we demonstrated the *importance of striking the optimal balance between the amount of feedback received through multiple drops, and the risk of having the UAV lost*. In both cases this underlined the large benefit of smart algorithms compared to random approaches. This also showed that *a predetermined, one-size-fits-all approach to network planning and deployment will result in a highly sub-optimal performance*. The algorithms presented in this chapter therefore justify the validity of the Phase I/Phase II approach to provide flexible, near-optimal planning for WSN deployment.

Having motivated and justified the importance of WSN deployment planning and outlined suitable algorithms for the general case, we now turn our attention to the specific scenario where the WSN is deployed to provide accurate localization to an agent navigating in a GPS-denied environment. A placement algorithm will be specifically developed for this application and will be directly applicable to Phases I, II and III.

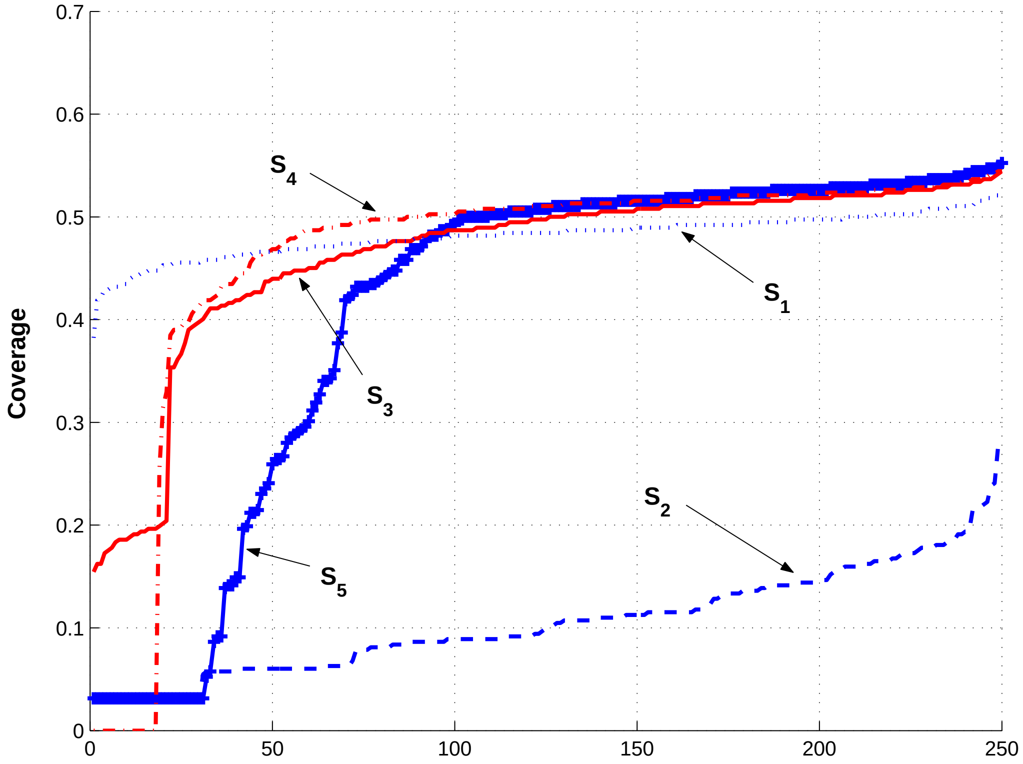

Figure 3-24: For design D_{19}, coverage values of 250 drop simulations, sorted from worst (to the left) to best (to the right) for all 5 sequences.

Chapter 4

Application to Localization in GPS-denied environments Using a Network of UWB Beacons

4.1 Introduction

The analysis developed so far has remained fairly general, so as to be applicable to any type of scenario. In the remaining chapters of this thesis we focus on a particular application of WSNs, the localization of agents in GPS-denied environments.

Since the Global Positioning System (GPS) became widely accessible [89], localization in the absolute frame (or geolocation) has found application in many different fields. In areas where there is a good line-of-sight (LOS) to GPS satellites, this technique provides a good estimate (within a few meters) of the user's location on the earth. However, in indoor and dense urban environments, geolocation has always been a more challenging problem for several reasons. Typically the GPS signal is not strong enough to penetrate through most materials. As soon as an object obscures the GPS satellite from the user's view, the signal is corrupted. This constrains the usefulness of GPS to open environments, and limits its performance in forests or in dense urban environments, as retaining a lock on the GPS signals becomes more difficult. GPS typically becomes completely useless inside buildings.

However there is an increasing need for accurate geolocation in cluttered environments, in addition to open spaces. In commercial applications for example, the tracking of inventory in warehouses or cargo ships is an emerging need. In military applications the problem of "blue force tracking," i.e., knowing where friendly forces are, is of vital importance. This is not a problem in open environments where systems can rely on GPS [1], but in dense urban or indoor environment, no satisfactory solution exists. Navigation in GPS-denied environment is also a pressing military need. For example untethered robots operating in enclosed environments such as urban canyons or inside buildings need accurate positioning to safely navigate.

Our scenario is that of an agent (such as a person or a vehicle) entering a building and *accurately* tracking its position over time. The position estimate should have a precision of under one meter (i.e. on the order of some of the building feature dimensions, such as hallway width).

4.2 Proposed Architecture

To address the problem of geolocation in cluttered environments, we consider deploying a WSN composed of n_{B} fixed beacons emitting radio signals[1]. We assume that once they are deployed, the location of these beacons is known. For example the beacons can be placed outside and rely on GPS, or agents can place them inside and determine their locations by survey or other means (e.g. an accurate map). The user agent can then extract range estimates to the beacons from the received signals (by time-of-arrival estimation), and then use the range estimates in a triangulation technique to determine its own position.

In the proposed architecture the beacons transmit ultra-wide bandwidth (UWB) signals. UWB technology potentially provides high ranging accuracy in cluttered environments [34, 42, 65, 68] owing to its inherent delay resolution and ability to penetrate obstacles [18, 98, 99, 101]. It is therefore the signal of choice for indoor ranging. Further information on the fundamentals of UWB can be found in [80, 90, 96, 97, 100] and the references therein. For

---

[1] In the configuration envisioned the "sensors" do not actually sense anything, but rather transmit a radio signal, so they are referred to as "beacons" in the remainder of this thesis.

simplicity we restrict our analysis to the two-dimensional case, but most results thereafter can be extended to three dimensions.

This architecture is illustrated on Figure 4-1. In response to an emergency in a high-rise building, a network of UWB beacons is deployed. These beacons provide a localization network to responders (or agents) moving inside the building. Note that the beacons can also provide the communication infrastructure necessary to relay information between the agents and the base (shown as a Humvee). Such a network has a similar structure to those previously studied.

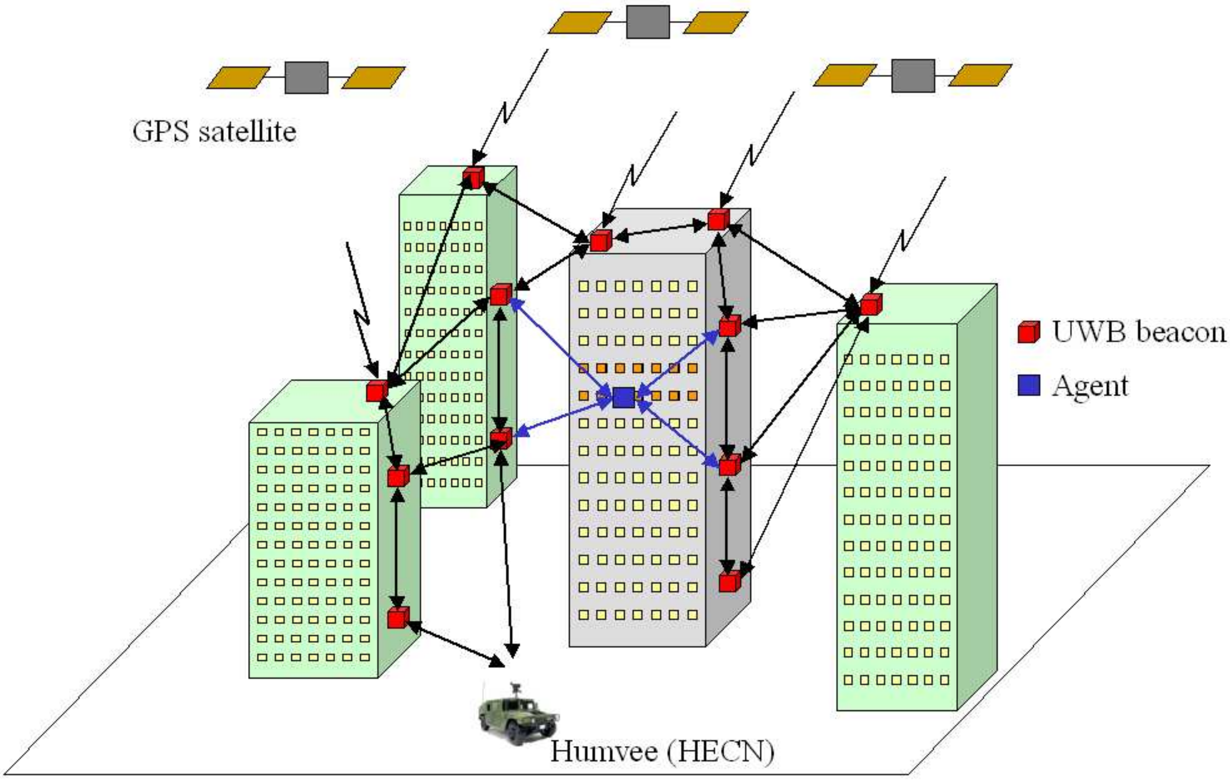

Figure 4-1: Architecture for accurate indoor localization.

In the remainder of this dissertation we will focus exclusively on the *localization* aspect of the network, so that the network architecture is less general than that described in Chapter 2. In particular *only one objective will be considered*. The potential role of the network as a communication infrastructure will not be further considered. This implies that *issues related to the communication connectivity between beacons or the HECN are not present*, as the beacons do not need to interact with one another to perform the localization task (which is performed by the agent itself). Therefore unlike the WSNs considered so far, information does not need to be relayed to other beacons. Of course if both localization and communication are of importance, then a method using MOGA can be implemented for Phase I, as discussed in the previous chapter. For example a second objective might be to maximize the total communication throughput of the network.

This WSN implementation implies that the agents is *cooperative*, i.e., it interacts with the WSN to perform its task. In this specific architecture, the role of the sensors (called beacons) is also slightly different. So far we have considered sensors that gathered information and relayed it to the rest of the network, whereas here the beacons are used to provide the *agent* with ranges. There is nothing fundamentally different with the previous general case, however. Consider the variant where the agent would broadcast a UWB signal, so that the sensors each measure the range to the agent, share the information and infer the location of the agent. In this case the WSN operates just as before (and the connectivity constraint between sensors needs to be enforced). Therefore, although the flow of information differs from the general case in this specific implementation (the ranges are measured by the agent, as opposed to the sensors/beacons), the WSN architecture considered here still fits our general framework.

The sensing range is assumed to be infinite, at least in theory. In fact the quality of range measurements will degrade with distance, so that the distance between beacons is still of concern, as discussed later on in this chapter. Note also that the placement algorithm presented in Chapter 6 can easily accommodate range constraints between beacons and agents, so that this assumption is without loss of generality.

The rest of this chapter is devoted to modeling the UWB range measurements so that in Chapter 5 an objective for the placement algorithm can be derived, which will account for the localization accuracy of beacon configurations.

4.3 Ultra-Wideband (UWB) Ranging

4.3.1 Radio-based localization

Different techniques can be used to localize an agent based on radio signals, such as the angle-of-arrival (AoA), the received signal strength (RSS), or the time delay information [17,42,76]. The AoA method measures the angle between the agent and the incoming signal using typically multiple antennae or an antenna array. By measuring the angle to 2 (non-aligned) transmitters in 2D, the location of the receiver is found at the intersection of the 2 lines. Since our system operates in a dense cluttered environment, however, the signal will often not travel in a straight line between the agent and the beacon. AoA methods are therefore not appropriate for UWB indoor positioning. The RSS method relies on a path-loss model, where the distance between receiver and transmitter is inferred by measuring the energy of the received signal. Three transmitters are needed in 2D to calculate the agent's location by using a triangulation technique. However the characteristics of the propagation channel must be known in advance in order to deduce distance from RSS, and in most cases such precise characterization will not be available.

The appropriate method for our problem is based on timing. Assuming the beacons and the agent are synchronized in time, the agent can calculate the time-of-flight of a signal by comparing its time stamp at transmission to its time-of-arrival (TOA). This can then be converted to a distance by multiplying the time-of-flight by the speed of light. Since the accuracy of TOA estimation increases with the signal-to-noise ratio (SNR) and the bandwidth [42], UWB (for a given SNR) will typically achieve great TOA accuracy compared to narrower band signals.

We note that in general the beacons and the agent will not have a common time-reference, in which case variants of this method must be used. In the round-trip method, the agent transmits a UWB signal to a certain beacon. Once it is received at the beacon, it is retransmitted and in turn received by the agent. By comparing the time of original transmission to the TOA (and accounting for the processing time required for the beacon to retransmit the signal), a time-of-flight can be determined [65] (in 2D, 3 beacons are sufficient to generate a position estimate). If the beacons are synchronized, but not the agent, then the time-of-difference-of-arrival (TDOA) method can be used. In this case the agent transmits a UWB signal, and the TDOA is computed at two beacons. The agent is then located on a hyperbola with foci at the beacons (again, 3 beacons are required in 2D). Note that these 2 variants require the agent to have the capability to transmit UWB signals. It is a realistic assumption if the agent also uses UWB for communication purposes in the mission scenario. Then the localization capability can simply be bootstrapped onto the communication system.

If the agent does not have the capability to transmit UWB signals, a similar approach to that of GPS can be used (assuming still that the beacons are synchronized). In this case, in addition to solving for the agent's position, we also solve for the clock bias of the agent. Then 4 beacons are necessary to yield a solution in 2D.

For simplicity and without loss of generality, we will assume in the following that the beacons and the agent have a common time-reference.

4.3.2 Time-of-arrival estimation

Several methods for TOA estimation of UWB signals can be found in [34,42,68]. Examples of low-complexity estimators include a maximum peak detection method and a threshold detection method. In the latter a threshold is chosen *a priori*, and the TOA is defined as the instant when the received amplitude goes above this threshold [34]. In spite of its simplicity, this method works quite well for UWB signals, especially in high SNR environments. This is illustrated on Figure 4-2.

4.3.3 Challenges to UWB ranging

Let us now define a few terms. We refer to a range measurement between a transmitter and a receiver as a direct path (DP) measurement if the range is obtained from the signal traveling along a straight line between the two points. A measurement can be non-DP if the DP signal is completely obstructed. In this case, the first signal to arrive at the receiver may come from reflected paths only. A LOS measurement is one obtained when the signal travels along an unobstructed DP, while a non-line-of-sight (NLOS) measurement can come from complete

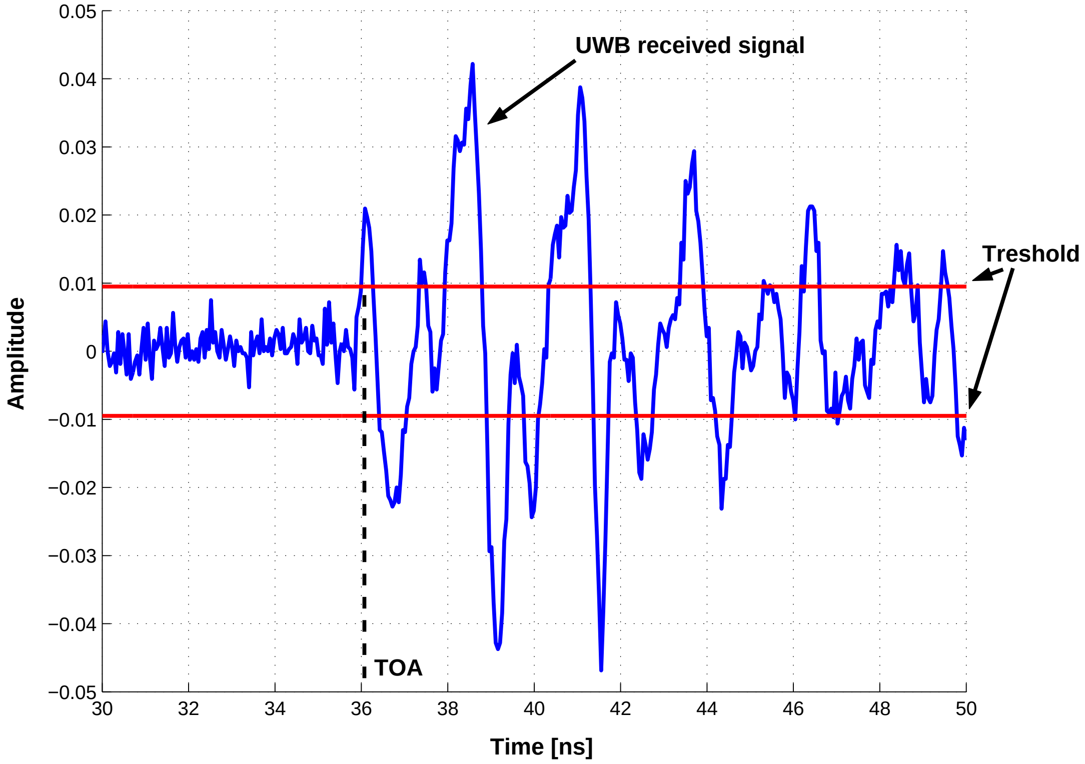

Figure 4-2: Illustration of a threshold detection method on a received UWB trace.

DP blockage or DP excess delay (in the latter case the DP is partially obstructed so that the signal has to pass through different materials, which results in additional delays).

In the case of radio-based localization, range measurements are typically corrupted by four sources: *multipath fading, thermal noise, DP blockage, and DP excess delay*. Multipath fading is due to destructive and constructive interference of signals at the receiver arriving via different propagation paths. This makes the detection of DP, if present, challenging. However UWB signals have the capability to resolve multipath components, which greatly reduces multipath fading [18,98,99,101]. The presence of thermal noise also limits the ability to accurately determine ranges as the distance increases, i.e., as the SNR decreases. We account for this effect by introducing a suitable model for the variance of the range measurement errors that increases with distance [34, 42].

The third difficulty is due to *DP blockage*. In some areas of the environment the DP to certain beacons may be completely obstructed, so that the only received signals are from reflections, resulting in measured ranges larger than the true distances. The fourth difficulty is due to *DP excess delay* incurred by propagation of the partially obstructed DP through different materials, such as walls. When such a partially obstructed DP signal is observed as first arrival, the propagation time depends not only upon the traveled distance, but also upon the materials it encountered. Because the propagation of electro-magnetic signals is slower in some materials than in the air, the signal arrives with excess delay, again yielding a range estimate larger than the true one. An important observation is that the effect of DP blockage and DP excess delay is the same: they both add a *positive bias* to the true range between agent and beacon, so that the measured range is larger than the true value (thus, from now on we will refer to such measurements as NLOS). This positive error has been identified as a limiting factor in UWB ranging performance [34, 65], so it must be accounted for.

4.3.4 Analysis of UWB Measurements

Using data collected by Win and Scholtz [101] we are able to build a probabilistic model for the beacon bias. These measurements were collected on a single floor of an office building, a map of which is provided in Figure 4-3. A bandwidth in excess of 1GHz was used, and the UWB transmitter was placed in a specific room, while measurements were taken in different rooms. For an in-depth description of the experiment, we refer the reader to [101]. We focus on two sets of data, one with measurements taken in room P, and the other with measurements taken at regular intervals along a corridor (points 1 though 33 on Fig. 4-3).

Measurements on a Grid

For the first set of data, 49 measurements were taken in a 7x7 square grid with a 6 inch spacing between measurement points. These were collected in room P, a different room from that of the transmitter (so that the signal had to propagate through several walls). From the received signal we determined the time-of-arrival of the signal by manual inspection, and inferred a range measurement. These range estimates were then compared to the true distances between transmitter and receiver (this true distance was obtained from the building

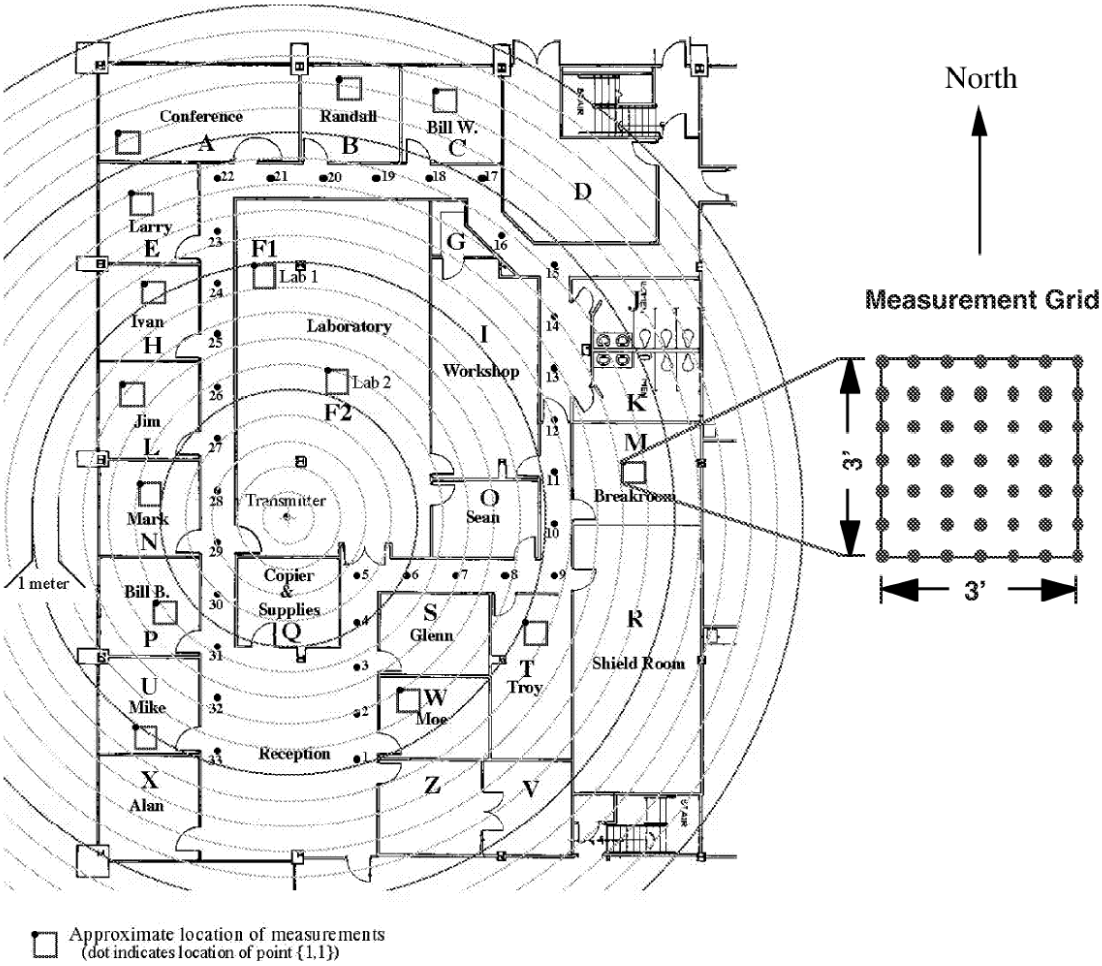

Figure 4-3: Floor map of the office building where the UWB measurements were performed [101]. The concentric circles are centered on the transmitter antenna and are spaced at $1m$ intervals. The data sets used are those from room P and the corridor points 1 through 33.

floor plan, with a precision of about $0.1m$). Figure 4-4 shows the differences between the range estimates using the measured signal and the true range for the 49 points. It can be seen that for most points there is a constant difference of about $1m$. This can be attributed to either LOS blockage or propagation delays (as mentioned before, these two phenomena produce the same effect). We also observe that for 4 points located toward one edge of the grid, this difference is higher. This shows that toward this edge either the propagation delay increases, or a multipath signal becomes the first arrival.

We draw two conclusions from this set of data. First, *the range estimates are posi-*

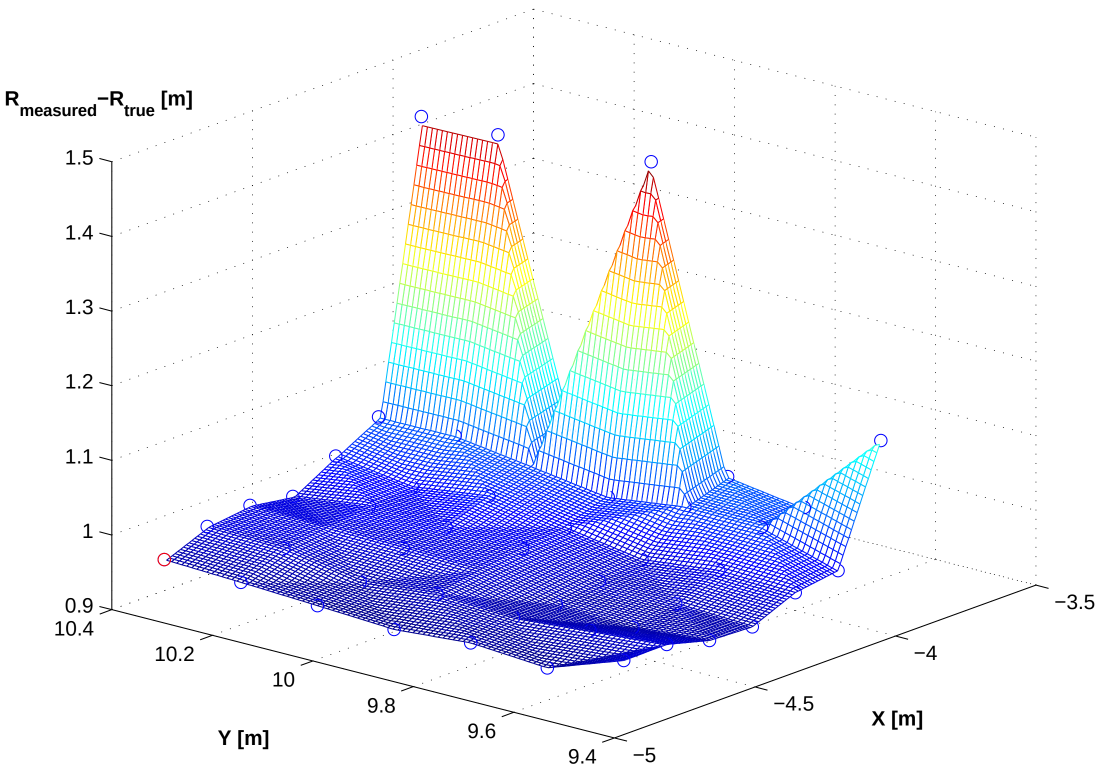

Figure 4-4: Difference in meters (on the z axis) between the measured and true ranges for the 49 points of room P.

tively biased, as expected. Second, *the value of the bias remains constant locally, and then suddenly changes by a discrete amount* as the receiver is moved through the building. For example, in Figure 4-4, notice the local plateau at $1m$ for $x < -4m$ and the sudden increase to $1.5m$ around $x = -4m$.

Measurements Along a Corridor

In this second experiment, 33 measurements were made at regular intervals as the receiver was moved around the floor along the building corridor.

Again, for each point a range estimate was extracted from the received signal and compared to the true range. This difference is plotted on Figure 4-5 for all 33 points. It is seen that the positive bias changes as the receiver moves along the corridor. In this case, the value of the bias varied between $0.15m$ and $1m$, and we will use these values in our experiments in a latter section. The distance between two measurement points was about $2m$, and we make an approximation that the bias value remains constant between two points. This assumption is consistent with our previous observation.

These observations will be used in the next section and in Chapter 7.

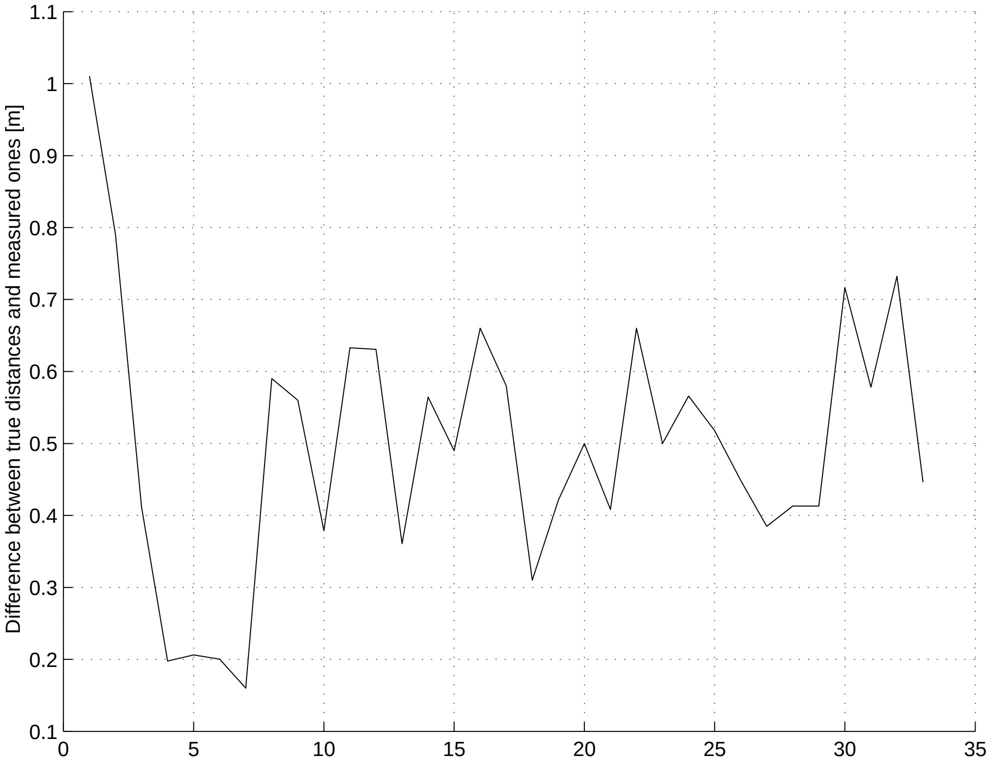

Figure 4-5: Difference in meters between the measured and true ranges for the 33 corridor locations.

4.3.5 Statistical Characterization of the UWB Range Measurements

We now derive a statistical characterization of the range measurements. From the previous section we note that the range measurements $\tilde{r_i}$ between the agent and the i^{th} beacon can be positively biased due to NLOS propagation. If we denote by d_i the true distance, the measured range $\tilde{r_i}$ can be expressed as:

$$\tilde{r_i} = d_i + b_i + \epsilon_i, \tag{4.1}$$

where b_i is the bias added to the i^{th} beacon and ϵ_i is a random Gaussian noise, independent of b_i, with zero-mean and variance σ_i^2. We model the dependence of the variance of ϵ_i on the distance d_i as $\sigma_i^2 \equiv \sigma^2(d_i) = \sigma_0^2 d_i^\alpha$, where α is the path-loss exponent and σ_0^2 is the variance at one meter [18, 42]. The probability density function (pdf) of ϵ_i is therefore:

$$f_{\epsilon_i}(\epsilon) = \frac{1}{\sqrt{2\pi}\sigma(d_i)} e^{-\frac{\epsilon^2}{2\sigma^2(d_i)}}. \tag{4.2}$$

Let us now model the statistics of the bias. If a detailed characterization of the bias is available, so that its exact value can be known at any location, then there is no need to account for biases at all: the predicted value can be subtracted from the range measurements. On the other hand, such characterization will not be possible in most cases since it assumes complete knowledge of the environment and of the agent position. This is unrealistic in military scenarios where the area of interest may be completely unmapped. We assume here that no such complete characterization of the biases is available.

Let us start by making some general observations about the biases based on the previous section 4.3.4. We first note that the bias will always be non-negative. Its actual value, however, will largely depend on the environment. We expect it to take a wider range of values in a cluttered environment with many walls, machines and furniture (such as a typical office building), than in an open space. In particular Fig. 4-4 reveals that the bias jumps from one value to another as the transmitter is moved through the building. In these measurements the bias is seen to vary between 0 and $1.5m$, depending on the room in which the measurements were taken. Note finally that the bias cannot grow infinitely large regardless of the propagation environment.

Although a detailed map of the environment may not be available, most of the time we will be able to classify the environment in broad terms, such as "concrete office building," or "wooden warehouse" (which we call an *environment class*). By performing range measurements in typical buildings of different classes beforehand, we can build a library of frequency histograms valid for different environments classes. We can then use these histograms to approximate the pdf of the biases in the building of interest. An example of such histogram is derived from Fig. 4-5, with the range measurements taken in the corridor. The corresponding frequency histogram of the bias is plotted on Figure 4-6(a). Such histograms will account for the positivity of the biases, as well as an approximate distribution of their expected value throughout the building. From Figure 4-6(a) we can see that biases around $0.5m$ are most common, while they were never larger than $1.1m$.

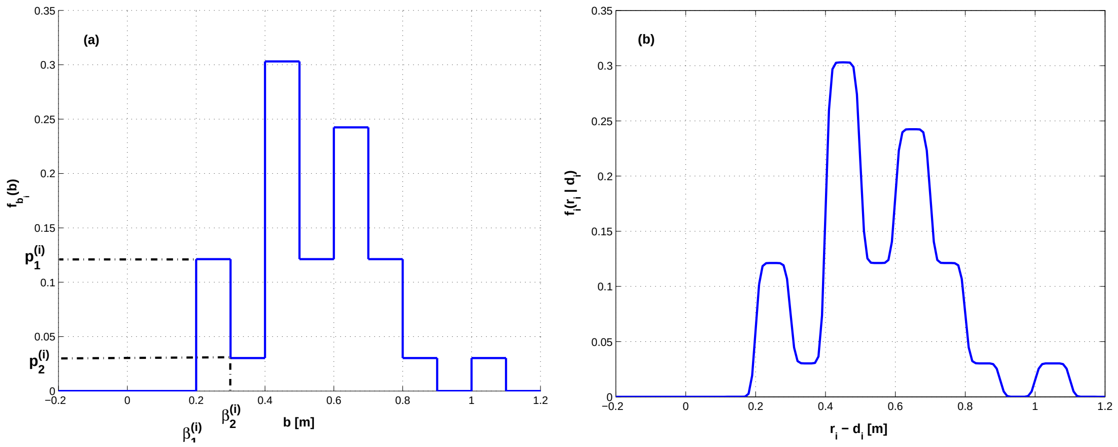

Figure 4-6: (a) Frequency histogram of the bias from range measurements performed in an office building [57]; (b) Corresponding pdf of the error in range measurements from beacon i $\tilde{r}_i - d_i$, for $d_i = 1m$ and $\beta_0^{(i)} = 10^{-3}m$

Let us assume such frequency histograms are available for each beacon. They may differ from beacon to beacon, so we index them by the beacon number i. The i^{th} histogram has $K^{(i)}$ bars, where the k^{th} bar goes from $\beta_{k-1}^{(i)}$ to $\beta_k^{(i)}$ and has frequency (height) $p_k^{(i)}$, as shown on Figure 4-6(a). We can therefore associate to the frequency histogram the pdf of b_i as:

$$f_{b_i}(b) = \sum_{k=1}^{K^{(i)}} w_k^{(i)} u_{\left\{\beta_{k-1}^{(i)},\beta_k^{(i)}\right\}}(b), \tag{4.3}$$

where $w_k^{(i)} = p_k^{(i)} \frac{1}{\beta_k^{(i)} - \beta_{k-1}^{(i)}}$, and $u_{\{a,a'\}}(b) = 1$ if $a \leq b \leq a'$, 0 otherwise, and $\beta_0^{(i)} = 0$. We note that if beacon i is LOS (i.e. it has no bias), then $f_{b_i}(b) = \delta(b)$ where $\delta(b)$ is the Dirac pseudo function.

Let us then lump the bias term with the Gaussian measurement noise $\tilde{\nu}_i = b_i + \epsilon_i$ and obtain the corresponding pdf:

$$\begin{aligned} f_{\tilde{\nu}_i}(\tilde{\nu}_i) &= \int_{-\infty}^{\infty} f_{b_i}(x) f_{\epsilon_i}(\tilde{\nu}_i - x)dx \qquad (4.4) \\ &= \sum_{k=1}^{K^{(i)}} w_k^{(i)} \int_{\beta_{k-1}^{(i)}}^{\beta_k^{(i)}} f_{\epsilon_i}(\tilde{\nu}_i - x)dx \qquad (4.5) \end{aligned}$$

$$= \sum_{k=1}^{K^{(i)}} w_k^{(i)} \left[Q\left(\frac{\tilde{\nu}_i - \beta_k^{(i)}}{\sigma(d_i)} \right) - Q\left(\frac{\tilde{\nu}_i - \beta_{k-1}^{(i)}}{\sigma(d_i)} \right) \right], \tag{4.6}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-t^2/2} dt$ is the Gaussian Q function. If the i^{th} beacon is LOS, then $f_{\tilde{\nu}_i}(\tilde{\nu}_i) = \frac{1}{\sqrt{2\pi}\sigma(d_i)} e^{-\frac{\tilde{\nu}_i^2}{2\sigma^2(d_i)}}$. The mean of $\tilde{\nu}_i$ is denoted m_i, and in order to obtain an unbiased estimator we subtract m_i from the i^{th} range measurement. This is equivalent to replacing $\tilde{\nu}_i$ by $\nu_i \triangleq \tilde{\nu}_i - m_i$. Let $\mathbf{p}_\text{A}$ be the vector of (x_A, y_A) coordinates of the agent, so that we have:

$$d_i(\mathbf{p}_\text{A}) = \sqrt{(x_\text{A} - x_i)^2 + (y_\text{A} - y_i)^2} \qquad i = 1, ..., n_\text{B} \tag{4.7}$$

where (x_i, y_i) are the coordinates of the i^{th} beacon. The unbiased range measurements are therefore modeled as:

$$r_i = d_i(\mathbf{p}_\text{A}) + \nu_i \tag{4.8}$$

with pdf given by

$$f_i(r_i|\mathbf{p}_\text{A}) = \sum_{k=1}^{K^{(i)}} w_k^{(i)} \left[Q\left(\frac{r_i - d_i(\mathbf{p}_\text{A}) + m_i - \beta_k^{(i)}}{\sigma\left(d_i(\mathbf{p}_\text{A})\right)} \right) - Q\left(\frac{r_i - d_i(\mathbf{p}_\text{A}) + m_i - \beta_{k-1}^{(i)}}{\sigma\left(d_i(\mathbf{p}_\text{A})\right)} \right) \right]. \tag{4.9}$$

The probability density function corresponding to the bias profile of Figure 4-6(a) is plotted on Figure 4-6(b) as a function of the error in range measurement $\tilde{r}_i - d_i$ for $d_i = 1m$ and $\beta_0^{(i)} = 10^{-3}m$. We observe that the pdf is a smoothed version of the bias profile, since the Gaussian error is small compared to the bias values.

This modeling will be useful in the next chapter, when deriving a lower bound on the localization accuracy.

4.4 Conclusion

In this chapter we described a WSN that can provide accurate indoor localization to an agent. This architecture relies on UWB beacons for ranging. We have also analyzed actual UWB range measurements and identified positive biases as a key limitation to the ranging accuracy. We finally built a statistical model of the range measurements, accounting for the presence of biases and the increase of the range measurement variance with distance.

This statistical modeling will be used in the next chapter to derive a lower bound on the localization accuracy of a specific beacon configuration, the so-called Position Error Bound (PEB). The PEB will serve as the objective to be minimized in the deployment planner. In particular in Chapter 6 a placement algorithm will be presented that minimizes the PEB in order to optimally place the beacons (in Phases I, II, and III). In Chapter 7 this statistical modeling will be used for the optimal operation of the network (also part of Phase III). In particular it is a critical part of the particle filter that estimates the agent position based on the range measurements and a strap-down IMU.

Chapter 5

The Position Error Bound

5.1 Motivation

In order to optimally place beacons, we need a metric that will measure the quality of beacon configurations. The accuracy of range-only localization systems depends mainly on two factors. The first is the geometric configuration of the system, i.e., how the beacons are placed relative to the agent. The second is the quality of the range measurements themselves. If the range estimates to the beacons were perfect, then three beacons, placed at any (but distinct) locations would be sufficient to determine the agent position unambiguously using any triangulation technique (since we restrict ourselves to a two-dimensional environment). In practice, however, these measurements are corrupted by several effects described in Chapter 4. Partial and complete LOS blockage lead to range estimates that are noisy and biased. Furthermore, the measurement variance increases as the received signal-to-noise ratio (SNR) decreases, which is in general related to an increase in distance between agent and beacon. All these factors will affect the localization accuracy to different degrees.

In this chapter we derive the Position Error Bound (PEB), a fundamental limit of localization accuracy using the Information Inequality [11] for the UWB localization system. The results derived thereafter are not limited to UWB ranging, however this technique is the only one able to guarantee high localization accuracy in cluttered environments. This limit can serve as the objective to be minimized in the deployment planner. In particular it can be used to design the localization network by deciding how many sensors to deploy and where to place them (Phase I), or to determine when additional sensor deployment is needed (Phases II and III). Beyond these applications it can also serve to map the area in terms of localization accuracy, so that the agent does not venture in locations with unsatisfactory coverage.

Literature Review

A measure of the localization performance used extensively in the GPS community is the geometric dilution of precision (GDOP) [89]. The GDOP provides a systematic way to compare geometric configurations of beacons. Bounds on the GDOP have also been derived in [102]. It turns out that there is a close relationship between the GDOP and the Cramér-Rao bound (CRB) from estimation theory [11, 19]. Along this line, estimation bounds for localization of sensors in a sensor network, either with or without anchor nodes, have been derived in [22]. The effect of geometric configurations and node density on the localization accuracy have been investigated in [83]. It has been shown numerically that collaborative localization, where sensors use range information not only from the anchor nodes but also from each other, is superior than localization relying solely on the anchor nodes [64].

Biases in bearing measurements are treated as additional noise with known a priori statistics in [39]. In [77, 78] the biases on range measurements are treated as additional parameters to be estimated for a range-only localization system. It was shown that in the absence of prior information about the biases, the NLOS beacons do not improve the geolocation accuracy [77], whereas if some prior information (such as their probability density function) is available, NLOS beacons can improve it [78]. In [79], the authors investigate the improvement in positioning accuracy if all multipath delays, instead of simply the first path, are processed. It is shown that using the first arrival only is sufficient for optimal localization when no prior information is known about the NLOS delays, whereas when such prior information is available, then the multipath delays can improve the accuracy (with the drawback of a more complex receiver). These papers do not take into account the dependence on distance of the variance of the range measurements. This dependence was addressed by numerical simulations in [64] and briefly discussed in [22]. Our analysis here will treat this dependence. We note that another bound was suggested in [95], and analyzed via simulations. Our contributions in this chapter are as follows [52–54]:

- We derive the PEB, a fundamental limit to the localization accuracy. This bound accounts for the dependence on distance of the range measurements and the presence of positive biases. The PEB can easily incorporate any statistical characterization of biases, e.g. one coming from measurements campaigns as described in Chapter 4.
- The structure of the PEB expression shows explicitly that the contribution in localization accuracy from each beacon is weighted by a factor reflecting the propagation environment, thereby providing insights into the importance of information coming from each beacon. In particular, it is shown that LOS beacons contribute with a larger weight than NLOS ones (at equal distance).
- We quantify the importance of information from NLOS beacons to show that NLOS beacons can significantly improve the localization accuracy (provided some minimum a priori knowledge of the biases is available), compared to the case where only LOS beacons are considered, especially in dense cluttered environments characterized by a low number of LOS beacons.
- We show that for a system of beacons placed on the vertices of a regular polygon, the lowest PEB is no longer found at the center of the polygon. This differs from the results typically found in the literature, where the sensor variances do not depend on the distance [66].
- We put forth the concept of localization accuracy outage to characterize the quality of localization throughout the area and determine whether more beacons should be deployed.

In Section 5.2 we derive the bound for our system of UWB beacons. In Section 5.3 we use this bound to conduct several numerical case studies. Finally in Section 5.4 we illustrate the achievability of the PEB on a few numerical examples to show that the PEB is not only a theoretical bound, but also a practical measure of localization accuracy that can be used in making engineering decisions, and is therefore well-suited to be the objective for a placement algorithm.

5.2 Derivation of a Lower Bound on the Localization Accuracy

In [78] the biases are treated as additional parameters to be estimated. However it can be shown that if the biases are modeled as described in Chapter 4, this approach fails to yield a bound lower than that when the NLOS beacons are ignored (the proof is given in Appendix B). In this thesis the biases are treated as additional noise terms on the time-of-arrival estimate, and we show that a better performance limit can be obtained, as the information from NLOS beacons can help improve the accuracy.

5.2.1 The Position Error Bound (PEB)

A lower bound on the covariance of any position estimator $\hat{\mathbf{p}}_{\mathrm{A}} = (\hat{x}_{\mathrm{A}}, \hat{y}_{\mathrm{A}})$ based on $\mathbf{r} = [r_1, r_2, ...r_{n_B}]$, the vector of n_{B} range measurements, is given by the Information Inequality [11][1]:

$$\mathbb{E}_{\mathbf{r}}\left\{(\mathbf{p}_{\mathrm{A}} - \hat{\mathbf{p}}_{\mathrm{A}})(\mathbf{p}_{\mathrm{A}} - \hat{\mathbf{p}}_{\mathrm{A}})^{\mathrm{T}}\right\} \geq \mathbf{J}^{-1}. \tag{5.1}$$

$\mathbf{J}$ is the Fisher information matrix (FIM) given by:

$$\mathbf{J} = \mathbb{E}_{\mathbf{r}}\left\{\left[\nabla_{\mathbf{p}_{\mathrm{A}}} \ln(f(\mathbf{r}|\mathbf{p}_{\mathrm{A}}))\right]\left[\nabla_{\mathbf{p}_{\mathrm{A}}} \ln(f(\mathbf{r}|\mathbf{p}_{\mathrm{A}}))\right]^{\mathrm{T}}\right\}, \tag{5.2}$$

where $f(\mathbf{r}|\mathbf{p}_{\mathrm{A}})$ is the pdf of the vector $\mathbf{r}$ conditioned on $\mathbf{p}_{\mathrm{A}}$, and $\nabla_{\mathbf{p}_{\mathrm{A}}}\{.\}$ denotes the gradient of a scalar with respect to $\mathbf{p}_{\mathrm{A}}$. Note that we have:

$$\sqrt{\mathbb{E}_{\mathbf{r}}\left\{(x_{\mathrm{A}} - \hat{x}_{\mathrm{A}})^2 + (y_{\mathrm{A}} - \hat{y}_{\mathrm{A}})^2\right\}} \geq \sqrt{\mathbb{T}\{\mathbf{J}^{-1}\}} \tag{5.3}$$

---

[1] The notation $\mathbb{E}_{\mathbf{r}}\{.\}$ denotes the expectation operator with respect to the random variable $\mathbf{r}$, the notation $V \geq W$ means that the matrix $V - W$ is positive semi-definite, and superscript $^{\mathrm{T}}$ denotes the transpose.

for any estimator of the position $\mathbf{p}_\mathrm{A} = (x_\mathrm{A}, y_\mathrm{A})$, where $\mathbb{T}\{.\}$ is the trace of a square matrix. In the remaining we refer to this expression as the PEB:

$$\mathsf{PEB}(x_\mathrm{A}, y_\mathrm{A}) \triangleq \sqrt{\mathbb{T}\{\mathbf{J}^{-1}\}}. \tag{5.4}$$

The PEB is a fundamental limit on the accuracy of any localization method.

5.2.2 Derivation of the FIM

We now seek to calculate the PEB for our system. Because the measurement errors are assumed to be statistically independent we have:

$$f(\mathbf{r}|\mathbf{p}_\mathrm{A}) = \prod_{i=1}^{n_\mathrm{B}} f_i(r_i|\mathbf{p}_\mathrm{A}), \tag{5.5}$$

where $f_i(r_i|\mathbf{p}_\mathrm{A})$ is given by (4.9). Hence:

$$\nabla_{\mathbf{p}_\mathrm{A}} \ln(f(\mathbf{r}|\mathbf{p}_\mathrm{A})) = \sum_{i=1}^{n_\mathrm{B}} \frac{1}{f_i(r_i|\mathbf{p}_\mathrm{A})} \begin{bmatrix} \frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}} \\ \frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial y_\mathrm{A}} \end{bmatrix}, \tag{5.6}$$

so that

$$\mathbf{J} = \mathbb{E}_\mathbf{r} \left\{ \sum_{i=1}^{n_\mathrm{B}} \sum_{j=1}^{n_\mathrm{B}} \frac{1}{f_i(r_i|\mathbf{p}_\mathrm{A})} \frac{1}{f_j(r_j|\mathbf{p}_\mathrm{A})} \begin{bmatrix} \frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}} \cdot \frac{\partial f_j(r_j|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}} & \frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}} \cdot \frac{\partial f_j(r_j|\mathbf{p}_\mathrm{A})}{\partial y_\mathrm{A}} \\ \frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial y_\mathrm{A}} \cdot \frac{\partial f_j(r_j|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}} & \frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial y_\mathrm{A}} \cdot \frac{\partial f_j(r_j|\mathbf{p}_\mathrm{A})}{\partial y_\mathrm{A}} \end{bmatrix} \right\}. \tag{5.7}$$

We show in Appendix B that all the terms in (5.7) for which $i \neq j$ are 0. We therefore have:

$$\mathbf{J} = \mathbb{E}_\mathbf{r} \left\{ \sum_{i=1}^{n_\mathrm{B}} \frac{1}{f_i(r_i|\mathbf{p}_\mathrm{A})^2} \begin{bmatrix} \left(\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}}\right)^2 & \frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}} \frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial y_\mathrm{A}} \\ \frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial y_\mathrm{A}} \frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}} & \left(\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial y_\mathrm{A}}\right)^2 \end{bmatrix} \right\}. \tag{5.8}$$

We are left with calculating partial derivatives. After a few algebraic manipulations we obtain:

$$\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}} = g_i(\nu_i)\cos\theta_i, \tag{5.9}$$

$$\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial y_\mathrm{A}} = g_i(\nu_i)\sin\theta_i, \tag{5.10}$$

where θ_i is the angle between the agent and the i^{th} beacon measured with respect to the horizontal, $\nu_i = r_i - d_i$, and $g_i(\nu_i)$ is given by:

$$\begin{aligned} g_i(\nu_i) = & \frac{1}{\sigma_0 d_i^{\alpha/2}\sqrt{2\pi}} \sum_{k=1}^{K^{(i)}} w_k^{(i)} \Bigg[\left(1 + \frac{\alpha}{2d_i}\left(\nu_i + m_i - \beta_k^{(i)}\right)\right) e^{-\frac{\left(\nu_i + m_i - \beta_k^{(i)}\right)^2}{2\sigma_0^2 d_i^\alpha}} \\ & - \left(1 + \frac{\alpha}{2d_i}\left(\nu_i + m_i - \beta_{k-1}^{(i)}\right)\right) e^{-\frac{\left(\nu_i + m_i - \beta_{k-1}^{(i)}\right)^2}{2\sigma_0^2 d_i^\alpha}} \Bigg]. \end{aligned} \tag{5.11}$$

We can therefore write:

$$\mathbf{J} = \mathbb{E}_\mathbf{r}\left\{ \sum_{i=1}^{n_\mathrm{B}} \frac{g_i(\nu_i)^2}{f_i(r_i|\mathbf{p}_\mathrm{A})^2} \begin{bmatrix} \cos^2\theta_i & \cos\theta_i\sin\theta_i \\ \cos\theta_i\sin\theta_i & \sin^2\theta_i \end{bmatrix} \right\}, \tag{5.12}$$

and by taking the expectation inside the sum we finally obtain:

$$\mathbf{J} = \sum_{i=1}^{n_\mathrm{B}} A(\boldsymbol{\beta}^{(i)}, d_i)\mathbf{M}(\theta_i) \tag{5.13}$$

with $\boldsymbol{\beta}^{(i)} = \{\beta_1^{(i)}, ..., \beta_{K^{(i)}}^{(i)}\}$,

$$\mathbf{M}(\theta) = \begin{bmatrix} \cos^2\theta & \cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta \end{bmatrix}, \tag{5.14}$$

and

$$A(\boldsymbol{\beta}^{(i)}, d_i) = \int_{-\infty}^{\infty} \frac{g_i(\nu_i)^2}{f_i(r_i|\mathbf{p}_\mathrm{A})} d\nu_i. \tag{5.15}$$

The expression (5.13) provides us with some useful insights. First, note that $\mathrm{M}(\theta_i)$ contains geometric information about the relative position of the agent with respect to the i^{th} beacon.

The FIM is therefore a weighted sum of this geometric information, where the weights $A(\boldsymbol{\beta}^{(i)}, d_i)$ depend on $\boldsymbol{\beta}^{(i)}$ and d_i. We show in Section 5.2.3 that these weights correspond to the quality of the information coming from the corresponding beacon, in other word they reflect the trustworthiness of the range information coming from this beacon.

We also note that our analysis can easily be extended to 3D. If the position vector $\mathbf{p}_\mathrm{A}$ is now a 3D vector, we define λ as its longitude (angle between the x axis and the projection of $\mathbf{p}_\mathrm{A}$ on the x-y plane) and ϕ as its latitude (angle between the projection of $\mathbf{p}_\mathrm{A}$ on the x-y plane and $\mathbf{p}_\mathrm{A}$). Then (5.9) and (5.10) become:

$$\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}} = g_i(\nu_i)\cos\lambda_i\cos\phi_i \tag{5.16}$$

$$\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial y_\mathrm{A}} = g_i(\nu_i)\sin\lambda_i\cos\phi_i \tag{5.17}$$

$$\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial z_\mathrm{A}} = g_i(\nu_i)\sin\phi_i \tag{5.18}$$

where $g_i(\nu_i)$ is given by (5.11). The only change from (5.13)-(5.15) is in the matrix $\mathrm{M}(\theta)$ which becomes:

$$\mathrm{M}(\lambda,\phi) = \begin{bmatrix} \cos^2\lambda\cos^2\phi & \cos\lambda\sin\lambda\cos^2\phi & \cos\lambda\cos\phi\sin\phi \\ \cos\lambda\sin\lambda\cos^2\phi & \sin^2\lambda\cos^2\phi & \sin\lambda\cos\phi\sin\phi \\ \cos\lambda\cos\phi\sin\phi & \sin\lambda\cos\phi\sin\phi & \sin^2\phi \end{bmatrix} \tag{5.19}$$

5.2.3 Analysis of the Weights A(β,d)

Let us investigate the behavior of $A(\boldsymbol{\beta}, d)$ as a function of $\boldsymbol{\beta}$ and d. Let us first consider the case when all the biases go to 0, with d fixed. This implies that the β_k tend to 0 for all k, and that the mean of the corresponding measurement noise m also goes to 0. We write this as $m - \boldsymbol{\beta} \to 0$. We then have the following first order approximation for small $m - \beta_k$:

$$Q\left(\frac{\nu}{\sigma} + \frac{m-\beta_k}{\sigma}\right) = Q\left(\frac{\nu}{\sigma}\right) - \frac{m-\beta_k}{\sigma\sqrt{2\pi}} e^{-\frac{\nu^2}{2\sigma^2}} + o(m-\beta_k), \tag{5.20}$$

$$e^{-\frac{(\nu+m-\beta_k)^2}{2\sigma^2}} = \left(1 - (m-\beta_k)\frac{\nu}{\sigma^2}\right) e^{\frac{-\nu^2}{2\sigma^2}} + o(m-\beta_k). \tag{5.21}$$

We use these to calculate the limit of (4.9) when $m - \boldsymbol{\beta} \to 0$:

$$\begin{aligned}
\lim_{m-\boldsymbol{\beta}\to 0} f(r|\mathbf{p}_\text{A}) &= \lim_{m-\boldsymbol{\beta}\to 0} \sum_{k=1}^{K} w_k \left[Q\left(\frac{\nu + m - \beta_k}{\sigma(d)}\right) - Q\left(\frac{\nu + m - \beta_{k-1}}{\sigma(d)}\right)\right] && (5.22)\\
&= \lim_{m-\boldsymbol{\beta}\to 0} \sum_{k=1}^{K} w_k \left[-\frac{m - \beta_k}{\sigma(d)\sqrt{2\pi}} + \frac{m - \beta_{k-1}}{\sigma(d)\sqrt{2\pi}}\right] e^{-\frac{\nu^2}{2\sigma^2(d)}} && (5.23)\\
&= \frac{1}{\sigma(d)\sqrt{2\pi}} e^{-\frac{\nu^2}{2\sigma^2(d)}} \lim_{m-\boldsymbol{\beta}\to 0} \sum_{k=1}^{K} w_k (\beta_k - \beta_{k-1}) && (5.24)\\
&= \frac{1}{\sigma(d)\sqrt{2\pi}} e^{-\frac{\nu^2}{2\sigma^2(d)}}, && (5.25)
\end{aligned}$$

where we have used the fact that $\sum_{k=1}^{K} w_k(\beta_k - \beta_{k-1}) = 1$. As expected, the pdf of the range measurements converges to a zero-mean Gaussian of variance $\sigma^2(d)$: if no biases are present, the only noise is Gaussian. We now look at the limit of $g(\nu)$. We have:

$$\lim_{m-\boldsymbol{\beta}\to 0} \left(1 + \frac{\alpha}{2d}(\nu + m - \beta)\right) e^{-\frac{(\nu+m-\beta)^2}{2\sigma^2}} = e^{-\frac{\nu^2}{2\sigma^2}} \lim_{m-\boldsymbol{\beta}\to 0} \left(1 + \frac{\alpha}{2d}\nu + (m - \beta)\left(\frac{\alpha}{2d} - \frac{\nu}{\sigma^2} - \frac{\alpha\nu^2}{2d\sigma^2}\right)\right), \tag{5.26}$$

so that:

$$\begin{aligned}
\lim_{m-\boldsymbol{\beta}\to 0} g(\nu) &= \frac{1}{\sigma(d)\sqrt{2\pi}} e^{-\frac{\nu^2}{2\sigma^2(d)}} \lim_{m-\boldsymbol{\beta}\to 0} \sum_{k=1}^{K} w_k \left[\left(\frac{\alpha}{2d} - \frac{\nu}{\sigma^2(d)} - \frac{\alpha\nu^2}{2d\sigma^2(d)}\right)(\beta_{k-1} - \beta_k)\right]\\
&= \frac{1}{\sigma(d)\sqrt{2\pi}} \left(\frac{\alpha\nu^2}{2d\sigma^2(d)} + \frac{\nu}{\sigma^2(d)} - \frac{\alpha}{2d}\right) e^{-\frac{\nu^2}{2\sigma^2(d)}}. && (5.27)
\end{aligned}$$

We can then calculate the limit of $A(\boldsymbol{\beta}, d)$ as $\boldsymbol{\beta}$ goes to 0. If we let $y = \nu / \left(\sigma(d)\sqrt{2}\right)$, we obtain:

$$\begin{aligned}
A(0, d) &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \left(\frac{\alpha}{d} y^2 + \frac{\sqrt{2}}{\sigma(d)} y - \frac{\alpha}{2d}\right)^2 e^{-y^2} dy && (5.28)\\
&= \frac{1}{\sqrt{\pi}} \left[\frac{\alpha^2}{d^2} \int_{-\infty}^{\infty} y^4 e^{-y^2} dy + \left(\frac{2}{\sigma^2(d)} - \frac{\alpha^2}{d^2}\right) \int_{-\infty}^{\infty} y^2 e^{-y^2} dy + \frac{\alpha^2}{4d^2} \int_{-\infty}^{\infty} e^{-y^2} dy\right]\\
&= \frac{1}{\sqrt{\pi}} \left[\frac{\alpha^2}{d^2} \frac{3\sqrt{\pi}}{4} + \left(\frac{2}{\sigma^2(d)} - \frac{\alpha^2}{d^2}\right) \frac{\sqrt{\pi}}{2} + \frac{\alpha^2}{4d^2}\sqrt{\pi}\right] && (5.29)
\end{aligned}$$

$$= \frac{1}{\sigma^2(d)} + \frac{\alpha^2}{2d^2}. \tag{5.30}$$

We now turn our attention to the case when the biases grow to infinity. In particular we consider the special case where $K = 1$ and $\beta_1 = \beta$ is the maximum bias, with the purpose of emphasizing in a simple way the role played by the NLOS beacons when the biases grow. This corresponds to the case where the biases are uniformly distributed between 0 and β. It can also model the case where all that is known about the environment is that the bias cannot be greater than β. The corresponding pdf (4.9) is plotted on Figure 5-1. In this case the weights $A(\beta, d)$ are equal to:

$$A(\beta, d) = \frac{1}{\beta\sigma(d)\pi\sqrt{2}} \int_{-\infty}^{\infty} h(y, \beta, d)dy, \tag{5.31}$$

where

$$h(y, \beta, d) = \frac{\left[\left(e^{-y^2} - e^{-\left(y+\frac{\beta}{\sigma(d)\sqrt{2}}\right)^2}\right)\left(1 + \frac{\alpha\sigma(d)}{d\sqrt{2}}y\right) - \frac{\alpha\beta}{2d}e^{-\left(y+\frac{\beta}{\sigma(d)\sqrt{2}}\right)^2}\right]^2}{Q\left(\sqrt{2}y\right) - Q\left(\sqrt{2}y + \frac{\beta}{\sigma(d)}\right)}. \tag{5.32}$$

When $\beta \to +\infty$ for a given d, the integral in (5.31) tends to a constant so that we have:

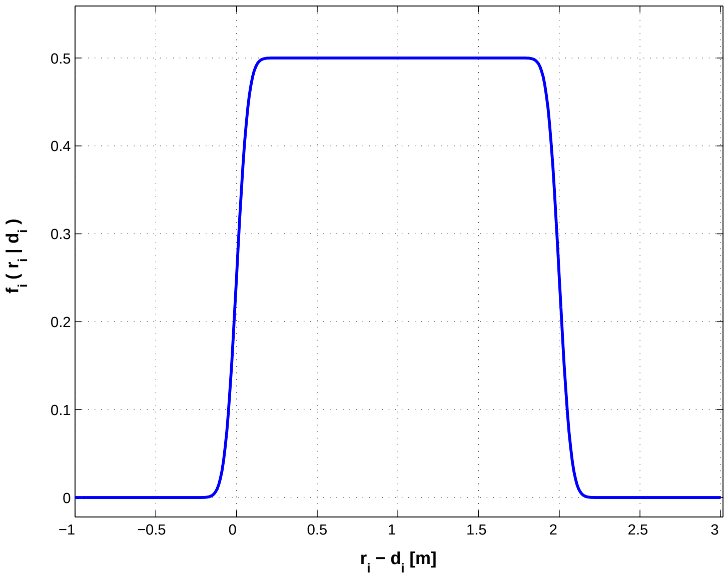

Figure 5-1: Pdf of the error in range measurements $\tilde{r}_i - d_i$ when $K^{(i)} = 1$, given the true distance $d_i = 15m$ and $\beta^{(i)} = 2m$.

$$A(\beta, d) \sim \frac{1}{\beta} \quad \text{when} \quad \beta \to +\infty \tag{5.33}$$

For a given d, $A(\beta, d)$ therefore approaches 0 as β goes to infinity. From (5.30) we see that the same is true when d goes to infinity (so that the range estimation variance goes to infinity). This is consistent with our intuition that the larger the bias or the range estimation variance, the less valuable the corresponding range information will be in determining the agent's position: the corresponding $\mathrm{M}(\theta)$ in (5.13) will receive a low weight and the contribution from the i^{th} beacon will be small. *The weights* $A(\boldsymbol{\beta}^{(i)}, d_i)$ *therefore quantify the importance of the information coming from the* i^{th} *beacon.* This implies that the information from beacons that are far away (large range measurement variance) or that are in highly cluttered areas (large bias) will not contribute much to the FIM. This behavior is illustrated on Figure 5-2, where we used (5.31) to plot $A(\beta, d)$ as a function of β, for several values of d. It can be seen that when d is small, the weights are sensitive to changes in β. Indeed at short distances the variance $\sigma^2(d)$ of the Gaussian noise is small, and therefore the dominating term in measurement inaccuracy will be the bias value. However as the distance increases, the value of β is less significant since the large $\sigma^2(d)$ tends to dominate the error. In any case, as d or β increase, the importance of the information coming from the corresponding beacon decreases.

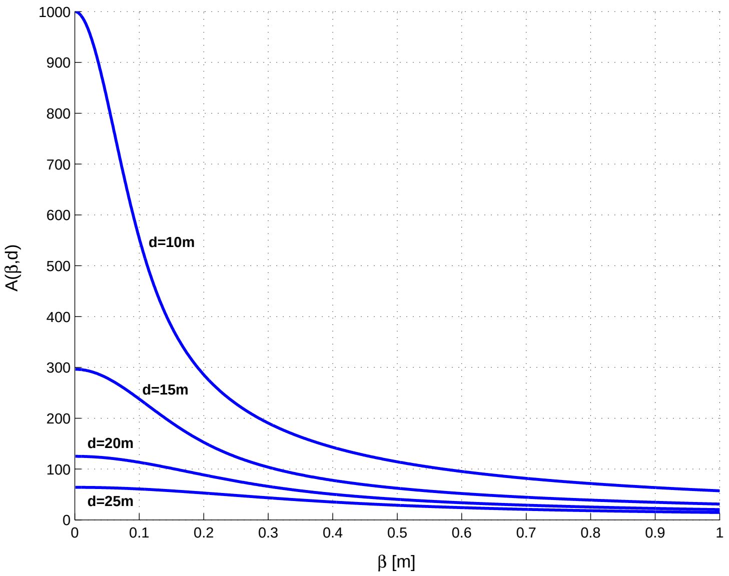

Figure 5-2: $A(\beta, d)$ for several values of d in the case where σ_0=0.001m and α=3

Consider a mix of LOS (for which $\beta_{\text{LOS}} = 0$) and NLOS beacons with no a priori knowledge of their biases (the biases can take any non-negative value so they have no upper bound, or $\beta_{\text{NLOS}} \to +\infty$). Then the information from the NLOS beacons is not used at all in the calculation of the PEB since $\lim_{\beta \to +\infty} A(\beta, d) = 0$. This observation is consistent with [77], in which it was shown that in the presence of a mix of LOS and NLOS beacons (with no *a priori* knowledge about the NLOS statistics), the performance depends only on the LOS beacons. Our result shows however that in the case where a priori knowledge of the NLOS beacon biases is available (which is always the case since biases cannot be infinitely large), the NLOS beacons should indeed be taken into account since they contribute to the PEB. In fact Section 5.3 will show that the contribution from NLOS beacons can be quite significant.

5.2.4 Analytical Expression for the PEB

We now use the analytical expression for the FIM to obtain the PEB (5.4). Recall from (5.13) that the FIM is a 2x2 matrix, so its inverse is easily obtained as:

$$\mathbf{J}^{-1} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} \sum_{i=1}^{n_{\text{B}}} A_i s_i^2 & -\sum_{i=1}^{n_{\text{B}}} A_i c_i s_i \\ -\sum_{i=1}^{n_{\text{B}}} A_i c_i s_i & \sum_{i=1}^{n_{\text{B}}} A_i c_i^2 \end{bmatrix}, \tag{5.34}$$

where $A_i = A(\boldsymbol{\beta}^{(i)}, d_i)$, $c_i = \cos\theta_i$, and $s_i = \sin\theta_i$. The PEB is then equal to:

$$\mathsf{PEB}(x_{\text{A}}, y_{\text{A}}) = \sqrt{\frac{\sum_{i=1}^{n_{\text{B}}} A_i}{(\sum_{i=1}^{n_{\text{B}}} A_i c_i^2)(\sum_{i=1}^{n_{\text{B}}} A_i s_i^2) - (\sum_{i=1}^{n_{\text{B}}} A_i c_i s_i)^2}}. \tag{5.35}$$

We can also expand the denominator to obtain the alternate expression:

$$\mathsf{PEB}(x_{\text{A}}, y_{\text{A}}) = \sqrt{\frac{\sum_{i=1}^{n_{\text{B}}} A_i}{\sum_{i<j} A_i A_j \sin^2(\theta_j - \theta_i)}}. \tag{5.36}$$

We stress that the limit on the localization accuracy given in (5.35) and (5.36) depends on the distance between the agent and the beacons, as well as on the presence of biases. If the variance were not dependent on the distance ($\alpha = 0$) and if no bias were present ($K^{(i)} = 0$

for all i), then according to (5.30) $A_i = 1/\sigma_0^2$ for all i and the PEB is equal to:

$$\mathsf{PEB}(x_{\mathrm{A}}, y_{\mathrm{A}}) = \sigma_0 \sqrt{\frac{n_{\mathrm{B}}}{(\sum_{i=1}^{n_{\mathrm{B}}} c_i^2)(\sum_{i=1}^{n_{\mathrm{B}}} s_i^2) - (\sum_{i=1}^{n_{\mathrm{B}}} c_i s_i)^2}}, \tag{5.37}$$

which is the product of the measurement standard deviation σ_0 and the GDOP. This is the case most commonly treated in the literature for range-only localization [19, 77, 83].

5.3 Numerical Case Studies

In the following case studies the biases are modeled as in Section 5.2.3, i.e., they are uniformly distributed between 0 and $\beta^{(i)}$ for all i. We adopt this model in order to study the influence of $\beta^{(i)}$ on the PEB, but a more general characterization is of course possible. We consider a set of n_{B} beacons where n_{LOS} of them are LOS, while the remaining $n_{\mathrm{NLOS}} = n_{\mathrm{B}} - n_{\mathrm{LOS}}$ are NLOS. For the sake of these case studies we assume we know whether a beacon is LOS or NLOS (if a map of the environment is available, channel modeling tools can be used [92], otherwise NLOS identification techniques exist [13, 41, 94]). We note that in practical applications it may be more realistic to assign each beacon with a probability of being LOS and NLOS.

We call ρ the fraction of LOS beacons, that is $\rho = n_{\mathrm{LOS}}/n_{\mathrm{B}}$. By definition LOS beacons have no bias so $\beta_{\mathrm{LOS}} = 0$. A LOS beacon may be one placed such that it is visible from any location in the area. On the other hand we assume that all the NLOS beacons have a common maximum bias $\beta_{\mathrm{NLOS}} = \beta$. This value will vary from building to building depending on whether the environment is highly cluttered or not. A NLOS beacon may be one placed behind a concrete wall: regardless of the agent's location in the environment, the beacon will be NLOS and the range measurement will be biased, with a maximum bias of β (which depends on the environment characteristics).

We acknowledge that assuming that beacons remain LOS or NLOS irrespective of the agent's position is not quite realistic, but our goal in the following case studies is to understand the general behavior of the PEB as we vary some key parameters. The analysis of the previous sections is perfectly amenable to cases where beacons are LOS and NLOS depending on the agent's location.

The following case studies are carried out with $\alpha = 3$ (a typical value for UWB indoor environments [18]) and $\sigma_0 = 10^{-3}m$.

5.3.1 Case Study 1: Importance of NLOS beacons

We first consider 6 beacons placed at the vertices of a polygon of radius $d = 5m$ and $20m$, and we assume that 3 beacons are LOS, while the 3 others are NLOS with a common maximum bias β (therefore $\rho = 0.5$). We plot $\mathsf{PEB}(\beta, d)$ at the center of the polygon as we vary β from 0 to $2m$ (solid curve on Figure 5-3). Also shown on the figure are PEBs if all beacons were LOS (lower dotted line) and if the NLOS beacons were ignored (upper dashed line).

It can be seen that when β goes to 0, $\mathsf{PEB}(\beta, d)$ converges to the PEB when all beacons are LOS, as expected since $\beta = 0$ corresponds to all beacons being LOS. On the other hand when β goes to infinity $\mathsf{PEB}(\beta, d)$ converges to the upper line. This is also expected since the information from the NLOS beacons is suppressed increasingly as β grows larger: for $\beta \to +\infty$ their contribution is altogether ignored (as shown in Section 5.2.3). However it is interesting to note that *in between those two extremes, the NLOS beacons can help significantly in reducing the PEB, especially for small values of β and large values of d.* Therefore the range information from NLOS beacons should not be dismissed, as it can greatly improve the localization accuracy. In particular, if there is an incentive in using as few beacons as possible to estimate the agent's position (e.g. minimization of the number of beacons deployed, energy conservation in communication, or computational complexity), these results can be used to decide which beacons to involve in the localization process.

5.3.2 Case Study 2: Mapping the PEB throughout an area

For a practical system we may be interested in the quality of localization not just at one point, but over an area. Let us map the value of the PEB throughout a square area for 6 LOS beacons placed at the vertices of a polygon of radius $d = 10m$ (Figure 5-4). This contour plot reveals that *the center of the polygon is no longer the location with minimum*

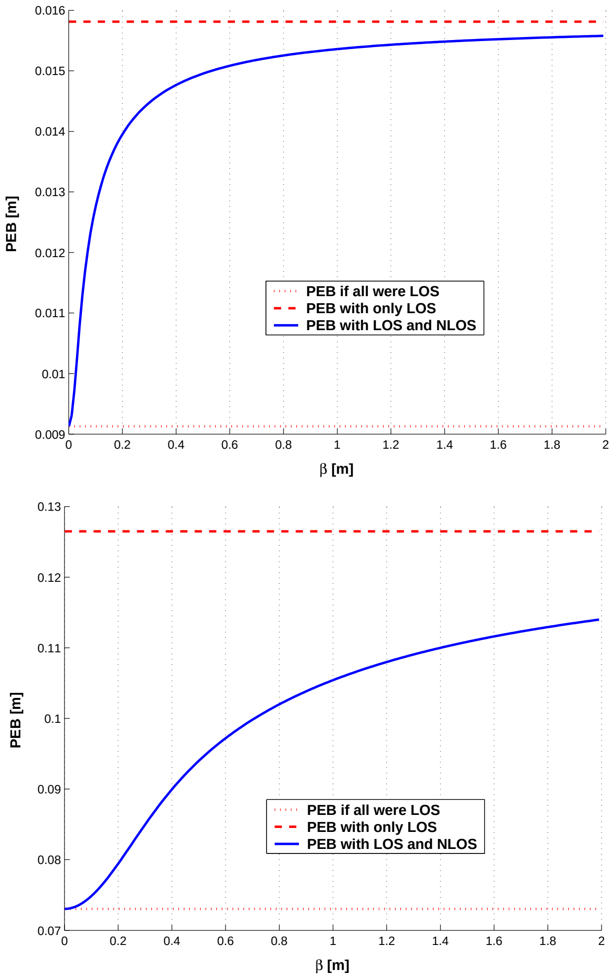

Figure 5-3: Case 2 when $d = 5m$ (top), and $d = 20m$ (bottom): PEB as a function of β (solid), bounded below by the PEB when all 6 sensors are LOS (dotted), and above by the PEB when only the 3 LOS beacons are considered (dashed). Lower values of the PEB indicate a better localization accuracy.

PEB, contrary to the common conclusions in the literature based on a model where the range measurement variance does not depend on distance between agent and corresponding beacons [66]. In other words, when the beacons are so arranged, the agent should not expect to have optimal localization accuracy in the center of the polygon. The situation becomes all the more complicated when NLOS beacons are included.

Suppose that we desire the PEB to be below a certain threshold τ for all points in the area. If at some location we have $\text{PEB} > \tau$, then whatever position estimator is used, the localization accuracy will be above the required threshold (since the PEB is a lower bound on the estimator accuracy). In this case we say that the localization system is *in outage* at

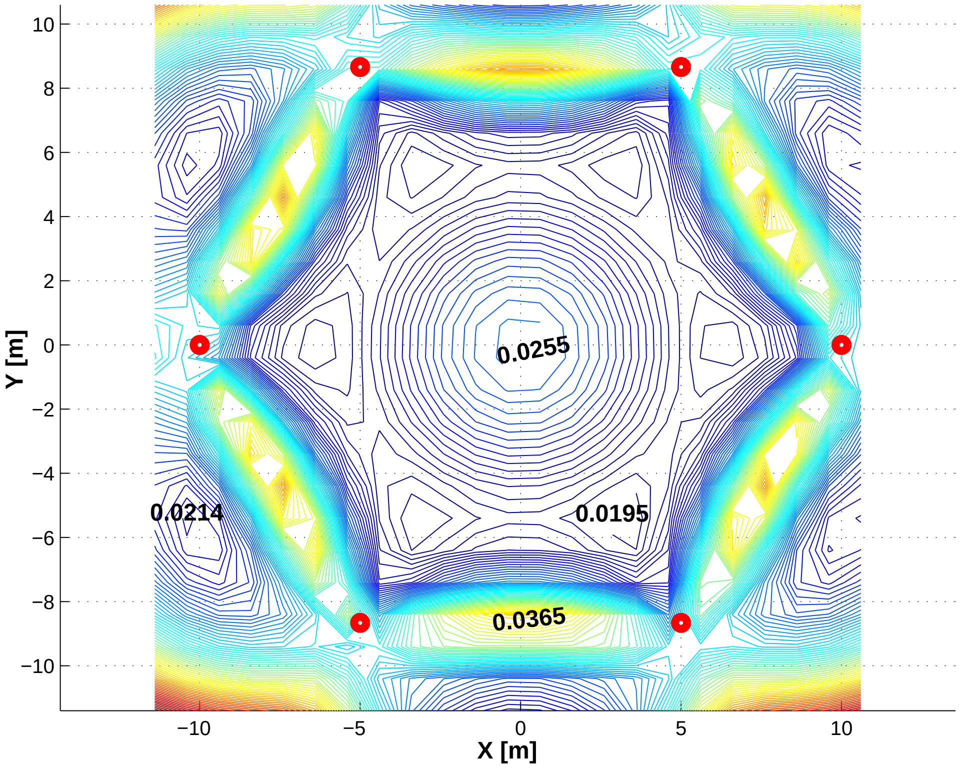

Figure 5-4: Case 1: Contour map of PEB when 6 LOS beacons are placed at the vertices of a polygon with $d = 10m$.

this location, and we define the *outage probability* for a given τ as:

$$p_{\text{out}}(\tau) \triangleq \mathbb{P}\{\mathsf{PEB} > \tau\}. \tag{5.38}$$

The outage probability tells us that as the agent moves through the area, with probability $p_{\text{out}}(\tau)$ the PEB will exceed the required threshold so that the localization accuracy will be unsatisfactory. If the threshold τ is chosen large enough, then the outage probability will approach 0, otherwise it will grow as more locations in the area will not meet the accuracy requirement. We illustrate this on Figure 5-5 where we plot the relative frequency diagram of the PEB over a $20m$ by $20m$ area where 10 beacons are placed at the vertices of a polygon, with $\beta = 2m$ and $\rho = 0.3$. The area covered for $\mathsf{PEB} > \tau$ represents the outage probability. Note that in most practical cases beacons will not remain LOS or NLOS for all possible target positions (exception made for particular cases). However, we assume here that this is the case in order to show the importance of the outage probability and the impact of NLOS beacons. Our conclusions remain valid in the more general case.

On Figure 5-6 we plot the outage probability as a function of the threshold τ for $\rho = 0.3$ and different values of β. We also show $p_{\text{out}}(\tau)$ for the extreme cases when only the LOS beacons are taken into account ($\beta \to +\infty$, rightmost curve) or when all the beacons are LOS ($\beta = 0$, leftmost curve). The curves for positive values of β lie between these two. In addition, we can observe that for low τ the sensitivity of the outage probability to β becomes larger. If a certain accuracy threshold is desired, these curves can help determine whether more beacons should be deployed. In Table 5.1 we show the outage probability for different values of the threshold τ and β.

Table 5.1: Outage probability for different values of τ[m] and β[m]

	$\beta = 0.1m$	$\beta = 1m$	$\beta = 2m$	$\beta \to +\infty$
$\tau = 0.05m$	0.025	0.4	0.57	0.9
$\tau = 0.1m$	$< 10^{-3}$	0.012	0.04	0.33
$\tau = 0.15m$	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	0.09

5.3.3 Case Study 3: Results with the ϵ-localization accuracy outage

In order to capture with a single number the quality of localization throughout the area, we define $\mathsf{PEB}_{1-\epsilon}$, the *$\epsilon$-localization accuracy outage*, as the value of the threshold τ for which the outage probability is ϵ, that is:

$$\epsilon = \mathbb{P}\{\mathsf{PEB} > \mathsf{PEB}_{1-\epsilon}\}. \tag{5.39}$$

In the remaining of this paper we consider $\epsilon = 0.1$, so that PEB_{90} gives a good indication of the performance we can expect 90% of the time as we move through the area. We plot PEB_{90} as a function of β for different values of ρ on Figure 5-7, when 10 beacons are placed randomly in a $20m$ by $20m$ area. It can be seen that the proportion of LOS and NLOS beacons has a significant impact on the PEB. If we have control over ρ, then we should try to increase it especially when β is large and the number of LOS is relatively small. In other words, *increasing the proportion of LOS beacons in cluttered environments (if it is possible) will significantly improve the PEB throughout the area.* This plot also tells us the performance loss if a beacon initially thought to be LOS turns out to be NLOS. For example notice the large increase in PEB between $\rho = 0.5$ and $\rho = 0.4$. This penalty grows for larger β and smaller ρ.

We now investigate the benefit of taking NLOS information into account, compared to the case where we neglect it. On Figure 5-8 we plot the ratio between $\mathsf{PEB}_{90}(+\infty)$ obtained by using only the LOS beacons and $\mathsf{PEB}_{90}(\beta)$, obtained with both LOS and NLOS beacons. Large values of this ratio indicate that the use of NLOS beacons yields a large reduction in the PEB compared to using only the LOS ones. The results show that *the information from NLOS beacons can lower the PEB by several factors*. This is especially true when the number of LOS beacons is relatively small. Also, lower uncertainty in the bias provides larger improvements in the PEB, indicating that the information from NLOS beacons is more useful when β is small.

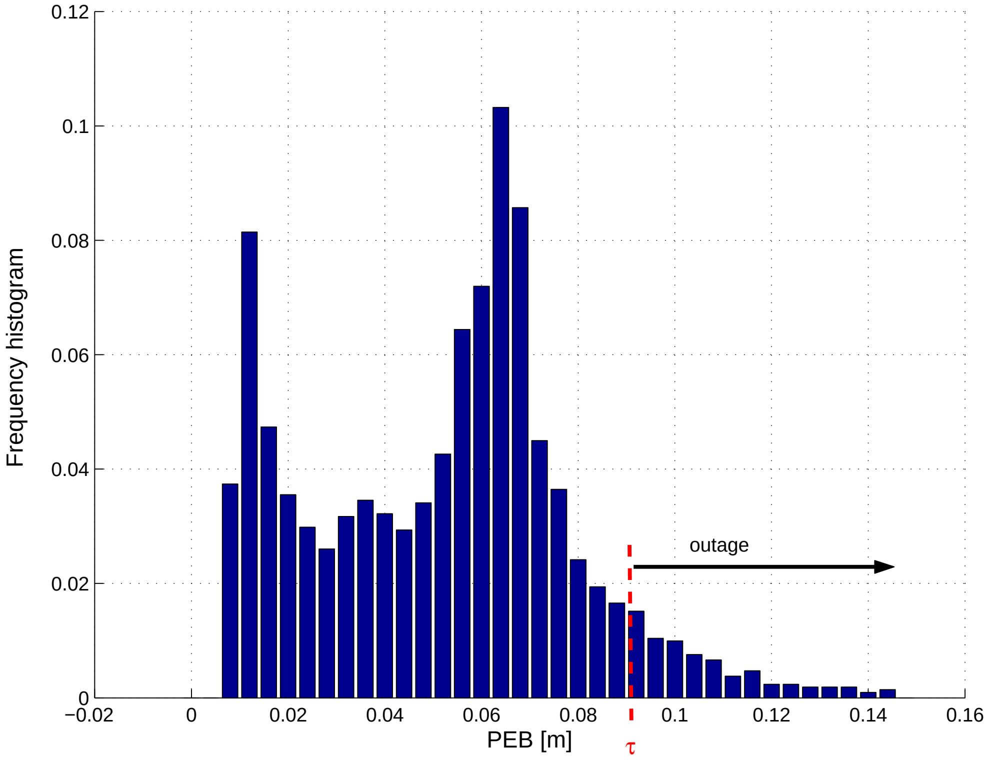

Figure 5-5: Case 3: Frequency histogram of the PEB over an area for 10 beacons placed at the vertices of a polygon, where $\beta = 2m$ and $\rho = 0.3$. The area to the right of τ represents the outage probability.

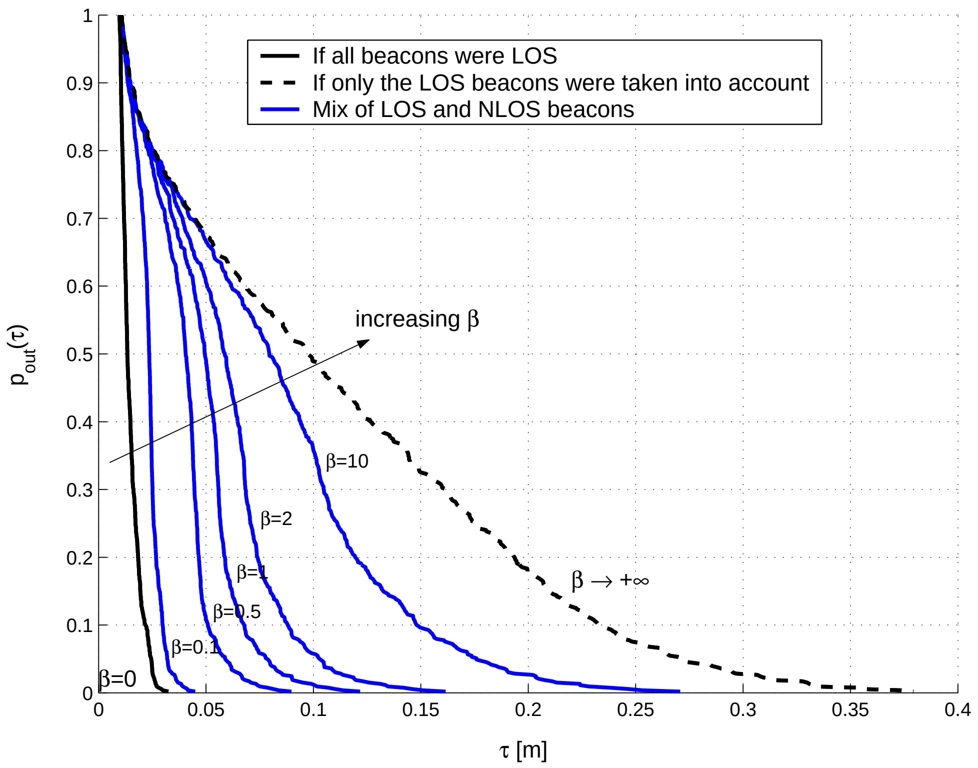

Figure 5-6: Case 3: Outage probability over the area the 10 beacons of Fig. 5-5 for $\beta = 0, 0.1, 0.5, 1, 2, 10m$, and the limit as $\beta \to +\infty$.

5.4 Achievability of the PEB

In this chapter we have used the PEB to measure a beacons configuration ability to provide accurate localization. It is a legitimate concern to wonder whether this measure has actually any relevance in reality. Indeed, although the PEB is a lower bound on the localization accuracy in theory, it may be of no practical value, for example if it is too loose. In particular two major concerns are:

- If beacon configuration A has a lower PEB than configuration B, will configuration A result in actual better localization accuracy than B?
- Also of interest is the PEB value *per se*: can it be used as a useful indication of the localization accuracy, or is it just too loose to be taken at face value?

It is known that the Maximum Likelihood (ML) estimate converges to the CRB as the SNR tends to 0 [93]. In our case this means that, when there is no bias, the PEB will

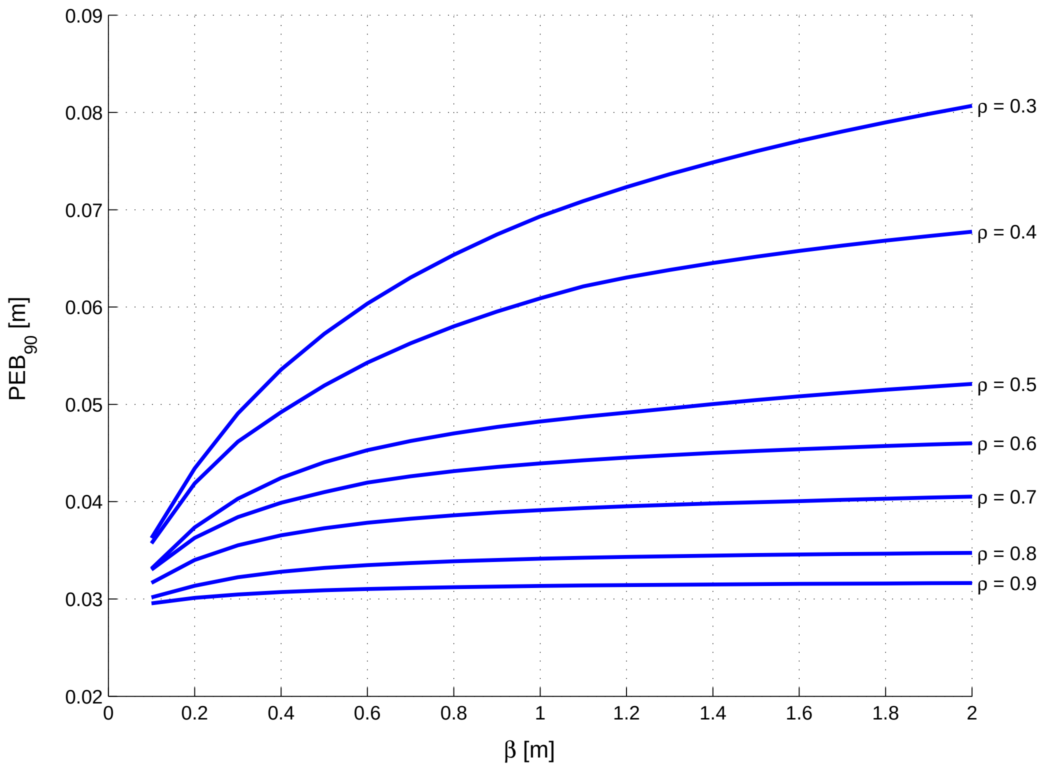

Figure 5-7: Case 3: Average $\mathsf{PEB}_{90}(\beta)$ for 10 beacons for different values of $\rho = n_{\text{LOS}}/n_{\text{B}}$.

be achievable as the variance σ^2 goes to 0. In this section we illustrate this result on a numerical example.

We consider the scenario depicted in Fig. 5-9, where $n_{\text{A}} = 50$ agent locations are divided into two subsets of the building. For each agent location, given a set of range measurements we calculate the ML estimate of the agent location by using a non-linear least-squares (NLLS) method described in Appendix C. By repeating this several times over one agent location, we can compute the Mean Square Error (MSE) of the position estimate. By repeating the process over all the agent locations, we can calculate MSE_{90} and compare it to PEB_{90}. Note that NLLS requires an initial position estimate, whereas the PEB assumes no *a priori* location information. The comparison between MSE and PEB is therefore not entirely fair, but it is still good because the initial position estimate given to NLLS is poor.

On Figure 5-10 we plot PEB_{90} (dashed) and MSE_{90} (solid) as a function of the number of sensors deployed. We do this when the beacons are placed uniformly along the boundary

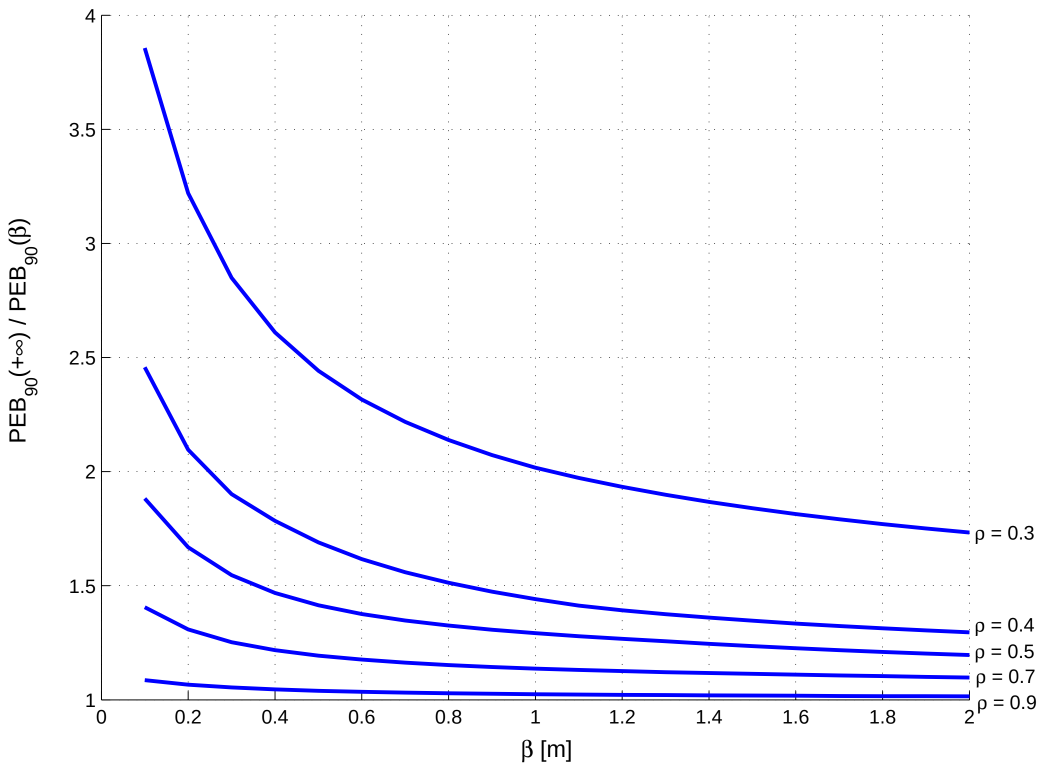

Figure 5-8: Case 3: Average $\frac{\mathsf{PEB}_{90}(+\infty)}{\mathsf{PEB}_{90}(\beta)}$. Large values indicate a large reduction in the PEB when the NLOS beacons are taken into account

(UNIFORM) and when they are clustered around the agent locations (RELOCATE). It can be seen that MSE_{90} is slightly larger than PEB_{90} (within about 5% for most test points). The same was observed for the other configurations. This indicates that, at least in some special cases when the bias is absent, the PEB will be close to achievable, so its actual value can be used as well. For example, if a certain localization accuracy is required, the PEB value can be used as an engineering tool to indicate whether more beacons should be deployed.

5.5 Conclusion

In this chapter we derived the PEB, which describes the limit on the accuracy of localization using UWB beacons. We considered the dense cluttered environment in which range measurements can be positively biased, and where their variance depends on the distance between agent and corresponding beacons. The PEB is easy to compute and accounts for

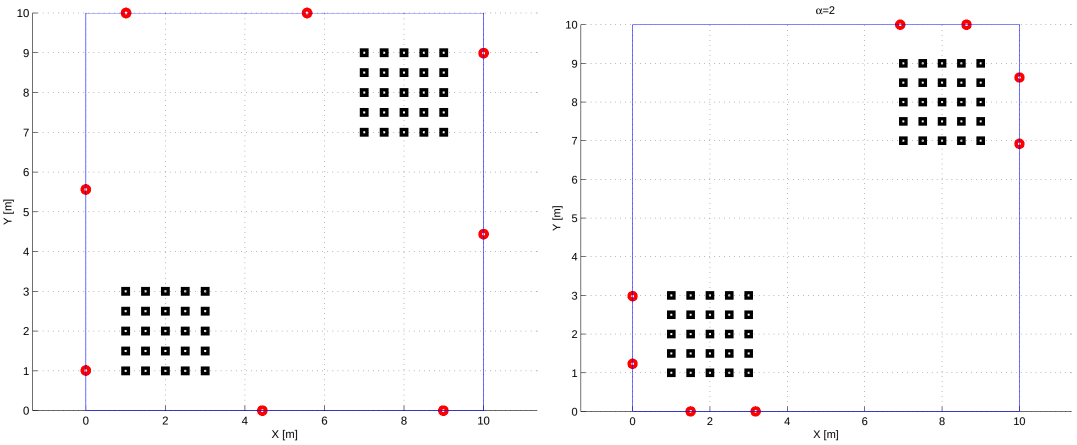

Figure 5-9: Setup for the comparison of PEB_{90} and MSE_{90}. The beacons are placed either uniformly along the boundary (UNIFORM, left), or on the boundary close to the agent locations (RELOCATE, right). The $n_{\mathrm{A}} = 50$ agent locations are shown as squares.

the geometric configuration of the system, the increase of measurement variance with distance, and the presence of positive biases with general statistical characterization.

We then investigated properties of this bound. We found that, contrary to results where the measurement variance is treated as constant [66], when the beacons are at the vertices of a regular polygon the minimum value of the PEB is not found at the center. We also found that, in the case of a mix of LOS and NLOS beacons, the information from the NLOS beacons can be very valuable: taking the NLOS beacons into account often yields a significantly lower localization bound, especially in cluttered environments. We put forth the concept of localization accuracy outage, which can guide in deciding whether to deploy additional beacons. Finally we verified that at least in certain cases where there is no bias, the PEB is achievable.

The PEB is therefore a practical measure of the localization accuracy, and it will be used in the next chapter as an objective for the optimal placement of the beacons.

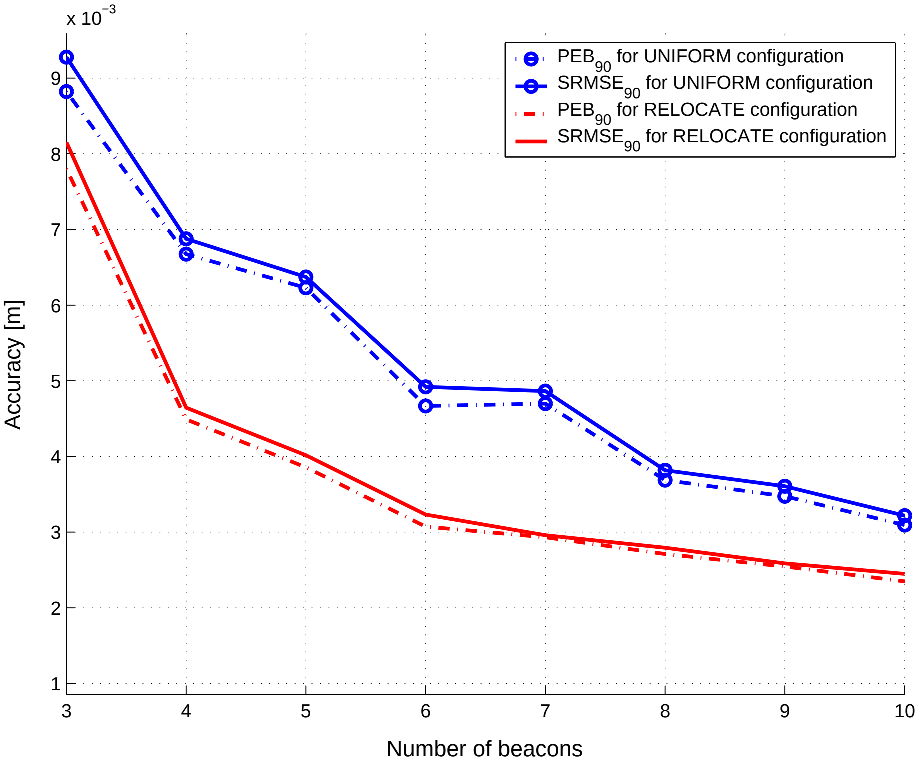

Figure 5-10: PEB_{90} (dashed) and MSE_{90} (solid) as a function of the number of beacons deployed, when the beacons are placed using UNIFORM (circles) or RELOCATE.

Chapter 6

Algorithm for Optimal Beacon Placement

6.1 Introduction

In the previous chapter we have used the Information Inequality to derive the PEB for our UWB localization system. As mentioned then, both the geometric configuration of the beacons and the quality of their range measurements are critical to accurate localization. The location of the beacons therefore directly impact the quality of localization. But although this (intuitive) dependence is well-known, there has been comparatively little work on the optimization of beacon placement. In this chapter we develop a coordinate-descent algorithm, RELOCATE, and demonstrate theoretically and through numerical simulations that this approach is well-suited for the optimal beacon placement problem [58, 59].

In Section 6.2 we start by presenting RELOCATE, the placement algorithm. In Section 6.3 we then apply RELOCATE to the *single* agent location problem where the range measurements are unbiased and have constant (but possibly different) variances. We introduce the coordinate transform that allows us to prove key results, in particular that RELOCATE converges to the global minimum efficiently. An algorithm solving this placement problem has been proposed in [103], but that method cannot be easily extended beyond that simple case. Our goal in this section will be to gain confidence in our algorithm before generalizing it to more complex cases. The coordinate transform is also critical for deriving an expected rate of convergence, so that our algorithm is shown to converge in practice with great precision in only a few steps, even for a large number of beacons.

In Section 6.4 we adapt our algorithm to deal with the more realistic case where the range measurements can be biased and where their variance depends on the beacon location, something not present to our knowledge in the literature. For example the measurement variance may increase with the distance to the agent, or when obstacles obstruct the line-of-sight between agent and beacons. In Section 6.5 we then consider the case where instead of minimizing the PEB at a single location (which does not have many realistic applications), the *average* PEB over *multiple* agent locations is to be minimized. For example the agent may be moving through the area and good localization must be provided along its trajectory. In such situations the quality of the range measurements may also vary with the beacon position, so this case is the most general. We show that our algorithm performs well on this realistic scenario by comparing its performance to Simulated Annealing (SA) [62]. SA is a natural choice for such combinatorial optimization problems, and it has been used on the beacon placement problem before [25]. We also finally show that by carefully planning the beacon placement, fewer beacons are required to achieve the same accuracy than one-size-fits-all approaches such as distributing the beacons evenly on the area boundary.

We restrict ourselves to static beacons operating in 2D. Initially we also restrict our analysis to an agent operating inside a convex building, while the beacons lie on the boundary of that building. This assumption is in accordance with our mission scenario where localization must be provided inside an area, while the beacons are deployed from outside that area. The convexity assumption will then be relaxed, and our algorithm will be applied to the case where the agent moves between different buildings, with the beacons placed on the buildings' walls.

Related Work

McKay [70] and Hegazy [46] minimize the condition number of the visibility matrix in order to minimize the impact of range measurement errors on the position estimate. In both [70] and [46], three dimensions are considered, although in [70] the beacons are constrained to lie on the ground. A Sequential Quadratic Programming method is used to solve the problem in [70], while in [46] an analytical solution is derived for 4 beacons. Sinha [87] maximizes the Fisher information gathered by a group of UAVs acting as beacons, which blends beacon geometry, survivability, and distance to the agent. A Genetic Algorithm coupled with a gradient descent algorithm is then used to search for the global minimum.

Abel [5], Martinez [69], and Zhang [103] optimize the beacon placement by minimizing a cost related to the Cramér-Rao bound (CRB), obtained from the Information Inequality. The acoustic beacons are constrained to lie on a line segment in [5], which allows for a simple analytic solution. In [69] an analytic form for the CRB in 2D and 3D is derived for the case when all the beacons have similar measurement variance. The classic result is found, namely that the configuration with minimum CRB is that with all beacons evenly spread around the agent [66]. The authors then use this result to dynamically control the beacons in order to track a moving agent. Finally in [103] Zhang considers the optimal placement of beacons in 2D, where the beacons have different measurement variances. He minimizes the determinant of the joint covariance matrix, which turns out to be equivalent to minimizing the CRB. Zhang obtains the minimum value of the CRB for this case and proposes an algorithm that converges to the optimal beacon placement in $n_{\mathrm{B}} - 3$ steps (where n_{B} is the number of beacons). Zhang's algorithm, however, does not generalize beyond the case of constant variances, which limits its applicability to more realistic scenarios.

Most of these papers are restricted to optimizing the beacon placement for the localization of a single agent location. A possible exception is [69] since the agent can move, but in this case the beacons are mobile and can adaptively rearrange their configuration. Sheng in [85] considers the placement of static beacons for the localization of an agent along its path, but the approach is more statistical in nature and assumes that many beacons can be deployed.

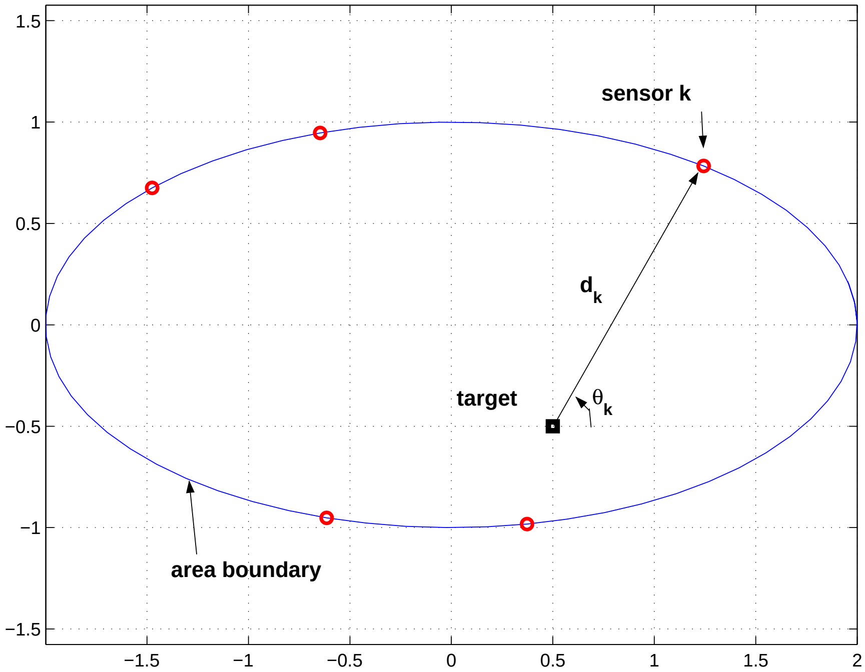

Figure 6-1: The agent (square) is inside a convex area, and the beacons (circles) are placed on its boundary. The distance between the agent and beacon k depends only on θ_k.

6.2 Preliminaries and Notations

6.2.1 The Design Variables θ

We will constrain the beacons to lie on the boundary of a set (representing for example the exterior walls of a building). Initially we assume this set to be convex, with the agent in its interior. In this case the position of the beacon is completely determined by θ_k, the angle the agent makes with the k^{th} beacons, as shown on Figure 6-1. The design variables to be optimized are the beacon locations, denoted by the vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{n_B})$. The convexity assumption of the set is for convenience, and will be relaxed later int he chapter, when for example beacons will be placed on walls belonging to different buildings. In that case the angles θ_k are not sufficient to unambiguously characterize the beacons positions and another parametrization should then be used.

6.2.2 The PEB

In the literature the range measurements are typically assumed to be unbiased, normally distributed independent variables with constant variances [5, 69, 103], but instead we will use the PEB derived in Chapter 5:

$$\mathsf{PEB}\left(\boldsymbol{\theta}\right) = \sqrt{\frac{\sum_{k=1}^{n_{\mathrm{B}}} A_k(\theta_k)}{\sum_{k=1}^{n_{\mathrm{B}}} A_k(\theta_k) c_k^2 \sum_k A_k(\theta_k) s_k^2 - \left(\sum_{k=1}^{n_{\mathrm{B}}} A_k(\theta_k) c_k s_k\right)^2}}, \tag{6.1}$$

where $c_k = \cos\theta_k$, $s_k = \sin\theta_k$, and the *importance weights* $A_k(\theta_k)$ are given by (5.15). As noted in Section 5.2.3, the importance weights indicate that the range measurements from different beacons will not be equally weighted in the PEB, depending on their accuracy. Moreover the relative positions of the beacons with respect to the agent will also influence the PEB (through the sine and cosine of (6.1)). Minimizing the PEB therefore implies striking the optimal balance between spatial diversity and range measurement quality. This non-trivial task requires using an optimization algorithm.

6.2.3 Generic Algorithm Description

We now present the RELOCATE algorithm in its generic form. We omit for the time being some additional conditions required to guarantee convergence to the *optimal* solution. This algorithm is a coordinate descent algorithm, i.e., it minimizes the PEB one coordinate at a time, until convergence. It operates as follows:

```
RELOCATE

• Randomly initialize $\boldsymbol{\theta}^1 = \{\theta_1^1, ...\theta_{n_{\mathrm{B}}}^1\}$, $p = 1$;

• Until convergence, do:

    1. Select beacon $i_p$ for relocation;

    2. Find the angle $\theta_{i_p}^*$ that minimizes the PEB along $\theta_{i_p}$;

    3. Set $\theta_{i_p}^{p+1} = \theta_{i_p}^*$ and $\theta_k^{p+1} = \theta_k^p$ for all $k \neq i_p$, so that $\boldsymbol{\theta}^{p+1} = \left(\theta_1^p, \ldots, \theta_{i_p}^*, \ldots, \theta_{n_{\mathrm{B}}}^p\right)$;

    4. $p \leftarrow p + 1$.
```

Coordinate descent algorithms are efficient as long as the minimization in step (2) is fast [9], i.e., as long as finding $\theta^*_{i_p}$ such that $\frac{\partial \mathsf{PEB}}{\partial \theta_{i_p}}(\theta^*_{i_p}) = 0$ and $\frac{\partial^2 \mathsf{PEB}}{\partial \theta^2_{i_p}}(\theta^*_{i_p}) \geq 0$ is easy. In the following section we show that step (2) can in fact be solved in *closed-form* when the importance weights are constant. This result, along with others on convergence and rate of convergence, will be made possible through the coordinate transform introduced next.

6.3 Single Agent Location and Beacons with Constant Importance Weights

Let us consider the case where the importance weights $A_k(\theta_k)$ do not depend on θ_k, that is, the weights are independent of where the beacons are located. This can be the case for example if we assume that the variance of the range measurements is constant ($\alpha = 0$) and there are no biases ($\beta = 0$), which is the typical assumption in the literature. From (6.1) we see that in this case the PEB is the same for angles modulo π, so we will only consider values of θ_k between 0 and π. We also assume without loss of generality that $A_n \geq ... \geq A_1$.

6.3.1 Coordinate Transform

Instead of working directly with the angles θ_i, we introduce a set of complex numbers (or vectors) $\mathbf{r}(\boldsymbol{\theta})$ and $\mathbf{z_i}(\boldsymbol{\theta})$ for $i = 1, \dots, n_\mathrm{B}$. This representation will be critical in allowing us to solve step (2) of RELOCATE in closed-form, to prove the optimal convergence of the algorithm, and to approximate its expected rate of convergence.

Definition 6.1 (Coordinate transform)**.**

$$\mathbf{z_i}(\boldsymbol{\theta}) = e^{-2j\theta_i} \sum_{k \neq i} A_k e^{2j\theta_k}, \quad \forall\, i = 1 \dots n_B, \tag{6.2}$$

$$\mathbf{r}(\boldsymbol{\theta}) = \sum_{k=1}^{n_B} A_k e^{2j\theta_k}, \tag{6.3}$$

$$r(\boldsymbol{\theta}) = |\mathbf{r}(\boldsymbol{\theta})|, \tag{6.4}$$

where j denotes the complex number such that $j^2 = -1$.

In particular we will show that if the PEB is minimum at $\tilde{\boldsymbol{\theta}}$, we must have $\Re\left\{\mathbf{z_i}(\tilde{\boldsymbol{\theta}})\right\} \leq 0$ and $\Im\left\{\mathbf{z_i}(\tilde{\boldsymbol{\theta}})\right\} = 0$ for all i, in other words all the $\mathbf{z_i}(\tilde{\boldsymbol{\theta}})$ must lie on the negative real axis[1].

More results and graphical interpretation of those vectors are given in Appendix D.1.

6.3.2 General Results on the PEB

The assumption of constant weights leads to two key results about the PEB. The first result, given in the following lemma, relates $r(\boldsymbol{\theta})$ to $\mathsf{PEB}\,(\boldsymbol{\theta})$.

Lemma 6.1. *When the importance weights are constant, minimizing* $\mathsf{PEB}\,(\boldsymbol{\theta})$ *is equivalent to minimizing* $r(\boldsymbol{\theta})$*, and* $\mathsf{PEB}\,(\boldsymbol{\theta})$ *can be re-written as*

$$\mathsf{PEB}\,(\boldsymbol{\theta}) = \sqrt{\frac{4\sum_{k=1}^{n_B} A_k}{(\sum_{k=1}^{n_B} A_k)^2 - r^2(\boldsymbol{\theta})}}. \tag{6.5}$$

Proof. This follows directly from (6.1) and (6.3) after a few elementary algebraic manipulations. □

$r(\boldsymbol{\theta})$ therefore provides a measure of the distance to optimality, and so it will be referred to as the *error radius*. The following lemma gives a lower bound on the error radius.

Lemma 6.2. *For any $\boldsymbol{\theta}$ we have*

$$r(\boldsymbol{\theta}) \geq r^* = \max(0, A_n - \sum_{k=1}^{n_B - 1} A_k). \tag{6.6}$$

Proof. If $A_{n_\mathrm{B}} \leq \sum_{k=1}^{n_\mathrm{B}-1} A_k$, then r^* is 0 and the relationship holds since the error radius is by definition always non-negative.

---

[1] where $\Re\{a\}$ and $\Im\{a\}$ respectively are the real and imaginary parts of a, a complex number.

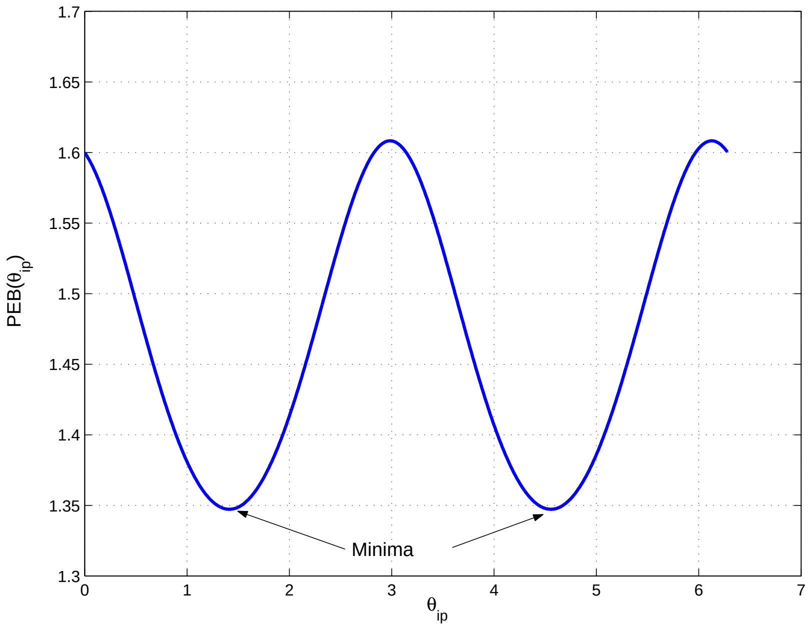

Figure 6-2: Shape of $\mathsf{PEB}(\theta_{i_p})$ when the weights are constant. Note that it has a unique minimum in $[0, \pi)$.

If $A_{n_{\mathrm{B}}} > \sum_{k=1}^{n_{\mathrm{B}}-1} A_k$, then $r(\boldsymbol{\theta})$ is minimized by having all the vectors $A_k e^{2j\theta_k}$ ($k = 1, \ldots, n_{\mathrm{B}} - 1$) aligned in the opposite direction to the vector of maximum amplitude $A_{n_{\mathrm{B}}} e^{2j\theta_{n_{\mathrm{B}}}}$. For example this can be achieved by setting $\theta_{n_{\mathrm{B}}} = 0$ and $\theta_k = \pi/2$ for $k = 1, \ldots, n_{\mathrm{B}} - 1$, so that $r(\boldsymbol{\theta}) = A_{n_{\mathrm{B}}} - \sum_{k=1}^{n_{\mathrm{B}}-1} A_k$ and (6.6) again holds. ☐

Therefore, if there exists $\tilde{\boldsymbol{\theta}}$ such that $r(\tilde{\boldsymbol{\theta}}) = r^*$, $\tilde{\boldsymbol{\theta}}$ is a global minimum of the PEB. In particular, if $A_n > \sum_{k=1}^{n-1} A_k$, the global minimum is easily found by setting (for example) $\theta_n = 0$ and $\theta_k = \pi/2$ for $k = 1, \ldots, n-1$. We will show in Section 6.3.3 that there always exists $\tilde{\boldsymbol{\theta}}$ such that $r(\tilde{\boldsymbol{\theta}}) = r^*$, and that it can be found using the proposed algorithm.

Step (2) of RELOCATE involves a 1-dimensional minimization of the PEB along θ_{i_p} (or equivalently a minimization of the error radius along θ_{i_p}). A typical shape of the PEB as a function of θ_{i_p} is plotted on Figure 6-2. As indicated on the figure, the PEB is a smooth function of θ_{i_p}, with a unique minimum in $[0, \pi)$. The second key result, stated in the following lemma, gives the closed-form expression of this unique minimum.

Lemma 6.3 (Closed-form solution to step (2) of RELOCATE)**.** *The minimization*

$$\theta_i^* = \arg \min_{\theta_i \in [0,\pi)} \{r(\theta_1, \ldots, \theta_i, \ldots, \theta_{n_B})\}$$

has a unique solution given by

$$\theta_i^* = \frac{1}{2} \arctan \left(\frac{\sum_{k \neq i} A_k \sin 2\theta_k}{\sum_{k \neq i} A_k \cos 2\theta_k} \right) + q\frac{\pi}{2}, \tag{6.7}$$

where $q \in \{0, 1\}$ *such that* $\Re \{\mathbf{z_i}(\theta_1, \ldots, \theta_i^*, \ldots, \theta_{n_B})\} \leq 0$*.*

Proof. We first write from (6.3)

$$r^2(\boldsymbol{\theta}) = |\sum_{k \neq i} A_k e^{2j\theta_k}|^2 + A_i^2 + 2A_i \sum_{k \neq i} A_k \cos(2\theta_k - 2\theta_i). \tag{6.8}$$

The minimum of $r(\theta_1, \ldots, \theta_i, \ldots, \theta_{n_\mathrm{B}})$ with respect to θ_i is the same as that of $r^2(\theta_1, \ldots, \theta_i, \ldots, \theta_{n_\mathrm{B}})$ and is found where the corresponding first partial derivative is 0 and the second derivative is non-negative. We first calculate

$$\begin{aligned}
\frac{\partial\left(r^2(\boldsymbol{\theta})\right)}{\partial \theta_i} &= 4A_i \sum_{k \neq i} A_k \sin(2\theta_k - 2\theta_i) && (6.9)\\
&= 4Ai \left[\left(\sum_{k \neq i} A_k \sin 2\theta_k \right) \cos 2\theta_i - \left(\sum_{k \neq i} A_k \cos 2\theta_k \right) \sin 2\theta_i \right] && (6.10)\\
&= 4A_i \Im \{\mathbf{z_i}(\boldsymbol{\theta})\}. && (6.11)
\end{aligned}$$

This is a sinusoidal function of θ_i, which is 0 twice in $[0, \pi)$. The two roots are given by

$$\begin{aligned}
\theta_i^0 &= \frac{1}{2} \arctan \left(\frac{\sum_{k \neq i} A_k \sin 2\theta_k}{\sum_{k \neq i} A_k \cos 2\theta_k} \right), && (6.12)\\
\theta_i^1 &= \frac{1}{2} \arctan \left(\frac{\sum_{k \neq i} A_k \sin 2\theta_k}{\sum_{k \neq i} A_k \cos 2\theta_k} \right) + \frac{\pi}{2}. && (6.13)
\end{aligned}$$

By taking the derivative of (6.10) one more time with respect to θ_i we obtain

$$\frac{\partial^2\left(r^2(\boldsymbol{\theta})\right)}{\partial\theta_i^2} = -8A_i\left[\left(\sum_{k\neq i} A_k \sin 2\theta_k\right)\sin 2\theta_i + \left(\sum_{k\neq i} A_k \cos 2\theta_k\right)\cos 2\theta_i\right] \tag{6.14}$$

$$= -8Ai\Re\left\{\mathbf{z_i}(\boldsymbol{\theta})\right\}, \tag{6.15}$$

which is a sinusoidal function of θ_i that is non-positive at either θ_i^0 or θ_i^1 (but not both), depending on which one yields $\Re\left\{\mathbf{z_i}(\boldsymbol{\theta})\right\} \leq 0$. There is therefore a *unique* value of θ_i in $[0, \pi)$ for which (6.10) is 0 and (6.15) is non-negative, and $r(\boldsymbol{\theta})$ has a unique minimum along the i^{th} coordinate, obtained at θ_i^* given by (6.7). This proof also implies that the minima of the PEB are such that all $\mathbf{z_i}$ lie one the negative real axis. ☐

The following corollary follows from this proof.

Corollary 6.4. *If at iteration p RELOCATE selects beacon i_p for relocation, the corresponding $\mathbf{z_{i_p}}$ is rotated by $-2\theta_{i_p}^*$ so as to lie on the negative real axis, i.e.,*

$$\Re\left\{\mathbf{z_{i_p}}(\theta_1^p, \ldots, \theta_{i_p}^*, \ldots, \theta_n^p)\right\} \leq 0 \textit{ and } \Im\left\{\mathbf{z_{i_p}}(\theta_1^p, \ldots, \theta_{i_p}^*, \ldots, \theta_n^p)\right\} = 0.$$

We also have the following result.

Corollary 6.5. *The stationary points of the PEB are such that all $\mathbf{z_i}$ lie one the real axis. Moreover candidates for minima are those stationary points for which all $\mathbf{z_i}$ lie on the* negative *real axis.*

Proof. At a stationary point $\tilde{\boldsymbol{\theta}}$ of the PEB, the gradient of the PEB with respect to $\boldsymbol{\theta}$ is the zero vector. In other words, all the first partial derivatives $\frac{\partial \mathsf{PEB}}{\partial\theta_i}(\tilde{\boldsymbol{\theta}}) = 0$. From (6.11) this means that $\Im\left\{\mathbf{z_i}(\tilde{\boldsymbol{\theta}})\right\} = 0$ of all i, i.e., all $\mathbf{z_i}$ lie on the real axis.

Candidates for minima will also be such that the second derivatives of the PEB will be positive. From (6.15) this implies that in addition $\Re\left\{\mathbf{z_i}(\tilde{\boldsymbol{\theta}})\right\} \leq 0$ of all i, i.e., all $\mathbf{z_i}$ lie on the *negative* real axis. ☐

The following lemma proves that RELOCATE actually converges to the stationary points that are candidates for minima.

Lemma 6.6 (Convergence of RELOCATE)**.** *RELOCATE converges to a stationary point* $\tilde{\boldsymbol{\theta}} = \left(\tilde{\theta}_1, \ldots, \tilde{\theta}_{n_B}\right)$*, such that all* $\mathbf{z_k}(\tilde{\boldsymbol{\theta}})$ *lie on the negative real axis, or*

$$\Re\left\{\mathbf{z_k}(\tilde{\boldsymbol{\theta}})\right\} \leq 0 \quad \textit{and} \quad \Im\left\{\mathbf{z_k}(\tilde{\boldsymbol{\theta}})\right\} = 0 \quad \forall k = 1, \ldots, n_B. \tag{6.16}$$

Proof. Coordinate descent algorithms are guaranteed to converge to some stationary point if the function to be minimized is continuously differentiable and if the minimum in step (2) is uniquely attained [9]. This is the case here as shown in Lemma 6.3, so RELOCATE converges to a stationary point. This stationary point will be such that all $\tilde{\theta}_k$ satisfy (6.7) and therefore all $\mathbf{z_k}(\tilde{\boldsymbol{\theta}})$ lie on the negative real axis, or $\Re\left\{\mathbf{z_k}(\tilde{\boldsymbol{\theta}})\right\} \leq 0$ and $\Im\left\{\mathbf{z_k}(\tilde{\boldsymbol{\theta}})\right\} = 0$ for all k (Corollary 6.4). □

RELOCATE therefore converges to stationary points that are candidates for minima (Corollary 6.5). Note however that such points may not be global minima. Consider for example the case with 3 beacons where $A_1 = A_2 = A_3 = 1$, so that $r^* = 0$. An optimal configuration is found at $\boldsymbol{\theta}^* = (\pi/3, \pi, -\pi/3)$ (which yields $r(\boldsymbol{\theta}^*) = 0 = r^*$). Suppose RELOCATE is started at $\boldsymbol{\theta}^1 = (0, 0, \pi/2)$. It is easy to see that $\mathbf{z_1}(\boldsymbol{\theta}^1) = \mathbf{z_2}(\boldsymbol{\theta}^1) = 0$ and $\mathbf{z_3}(\boldsymbol{\theta}^1) = -2$. All the $\mathbf{z_k}$ lie on the negative real axis, while $r(\boldsymbol{\theta}^1) = 1 > r^*$: the algorithm has converged to a suboptimal stationary point. In the next subsection an additional step will be added to RELOCATE in order to guarantee convergence to the *global* minimum.

6.3.3 RELOCATE for Constant Importance Weights

Let us now implement RELOCATE for the case of constant importance weights.

```
RELOCATE
- Define $r^* = \max(0, A_{n_B} - \sum_{k=1}^{n_B - 1} A_k)$ and the convergence threshold $\epsilon$. Set $p = 1$;
- Randomly initialize $\boldsymbol{\theta}^1 = (\theta_1^1, ..., \theta_{n_B}^1)$;
- While $(r(\boldsymbol{\theta}^p) - r^*)/r^* > \epsilon$, do:
    1. Choose $i_p$ such that $i_p = \arg\min_{k=1,...,n_B}\{|A_k - |\mathbf{z_k}(\boldsymbol{\theta}^p)||\}$;
```

2. Calculate $\theta^*_{i_p}$ according to (6.7);
3. Set $\theta^{p+1}_{i_p} = \theta^*_{i_p}$ and $\theta^{p+1}_k = \theta^p_k$ for $k \neq i_p$, so that $\boldsymbol{\theta}^{p+1} = \left(\theta^p_1, \ldots, \theta^*_{i_p}, \ldots, \theta^p_{n_\mathrm{B}}\right)$;
4. $p \leftarrow p + 1$.
5. If all $\mathbf{z_k}(\boldsymbol{\theta}^p)$ lie on the negative real axis, enter the INCREASE_RADIUS routine;

- Else, stop.

INCREASE_RADIUS routine:

1. Write $r\left(\boldsymbol{\theta}^p\right)$ as $\sum_{k\in S} A_k - \sum_{k\in\overline{S}} A_k$;
2. Let l and m be 2 distinct indices in S;
3. Choose θ^{p+1}_l such that $|A_m - |\mathbf{z_m}(\theta^p_1, \ldots, \theta^{p+1}_l, \ldots, \theta^p_{n_\mathrm{B}})|| < \sum_{k\in S} A_k - \sum_{k\in\overline{S}} A_k$;
4. $p \leftarrow p + 1$, select sensor m for relocation, and go to step (2) of RELOCATE.

Note that in step (1) of RELOCATE the beacon that yields the maximum decrease in error radius is chosen (see Corollary D.5 in Appendix D.1). Alternatively, the beacon chosen for relocation at iteration p could also be selected at random.

According to Lemma 6.6, steps (1)-(4) of RELOCATE converge to a stationary point such that all the $\mathbf{z_k}$ lie on the negative real axis. As we illustrated, however, some of these configurations are suboptimal. We show in Lemma 6.7 that step (5) enables RELOCATE to escape such suboptimal stationary points so that it converges to the *optimal* stationary point (which is guaranteed to exist), characterized by an error radius equal to r^*.

Lemma 6.7 (Optimal convergence of RELOCATE)**.** *RELOCATE converges to the global minimum:*

- *If* $A_{n_B} > \sum_{k=1}^{n_B-1} A_k$, *RELOCATE finds* $\boldsymbol{\theta}^*$ *such that* $r(\boldsymbol{\theta}^*) = A_{n_B} - \sum_{k=1}^{n_B-1} A_k$*;*
- *Otherwise, RELOCATE finds* $\boldsymbol{\theta}^*$ *such that* $r(\boldsymbol{\theta}^*) = 0$.

Proof. The proof is given in Appendix D.3. □

Let us illustrate how the algorithm operates on an example where all $A_k = 1$. We first reproduce some of the key results derived in Appendix D.1.

- The $\mathbf{z_k}\left(\boldsymbol{\theta}^p\right)$ are vectors starting at $(0,0)$ and ending on the same circle of radius $r\left(\boldsymbol{\theta}^p\right)$, centered at $(-1,0)$ (Corollary D.2);
- When beacon i_p is relocated in step (2) of RELOCATE, the corresponding $\mathbf{z_{i_p}}$ is rotated about $(0,0)$ so as to lie on the negative real axis (Corollary 6.4), while its magnitude $|\mathbf{z_{i_p}}|$ does not change (Corollary D.3). Note also that although the other θ_k (for $k \neq i_p$) are unchanged in this process, the corresponding $\mathbf{z_k}$ rotate and change magnitude;
- If beacon i_p is relocated at iteration p, the error radius is equal to $|A_k - |\mathbf{z_{i_p}}(\boldsymbol{\theta}^p)||$. Once it is relocated on the real axis, the error radius can also be found by measuring the distance between the tip of $\mathbf{z_{i_p}}(\boldsymbol{\theta}^{p+1})$ and $(-1,0)$ (Corollary D.5).

On Figure 6-3 are plotted the vectors $\mathbf{z_k}$ for 3 iterations of RELOCATE when 5 beacons are to be optimally placed. At the first iteration (Fig. 6-3(a)) beacon 3 is selected, and the corresponding $\mathbf{z_3}$ is rotated by almost π to lie on the negative real axis (Fig. 6-3(b)). Throughout the process its magnitude remains 1.62 (note the change in the axes scale). However the error radius has decreased significantly from 3.6 (in Fig. 6-3(a)) to 0.62 (Fig. 6-3(b)).

At the next iteration beacon 5 is selected for relocation and $\mathbf{z_5}$ is made to lie on the negative real axis as shown on Fig. 6-3(c). The error radius is decreased to 0.1 in the process. By repeating this procedure the error radius diminishes to zero and all the $\mathbf{z_k}$ converge to $(-1,0)$.

Note that at each iteration the beacon whose vector yields the largest decrease in error radius is selected, as specified by step (1) of RELOCATE (i.e. the one whose $|\mathbf{z_k}|$ is closest to 1). As can be seen on Fig. 6-3 beacon 3 is the first to be selected, then beacon 5 for the second iteration, and finally beacon 1 for the third iteration.

The final configuration of beacons is shown on Figure 6-4. The final PEB is equal to $0.8944m$, which is optimal since it is equal to the value of the PEB when $r = 0$ in (6.5). The beacons are evenly distributed around the agent. Note that any beacon could be moved by π without changing the PEB value, so other configurations are optimal as well.

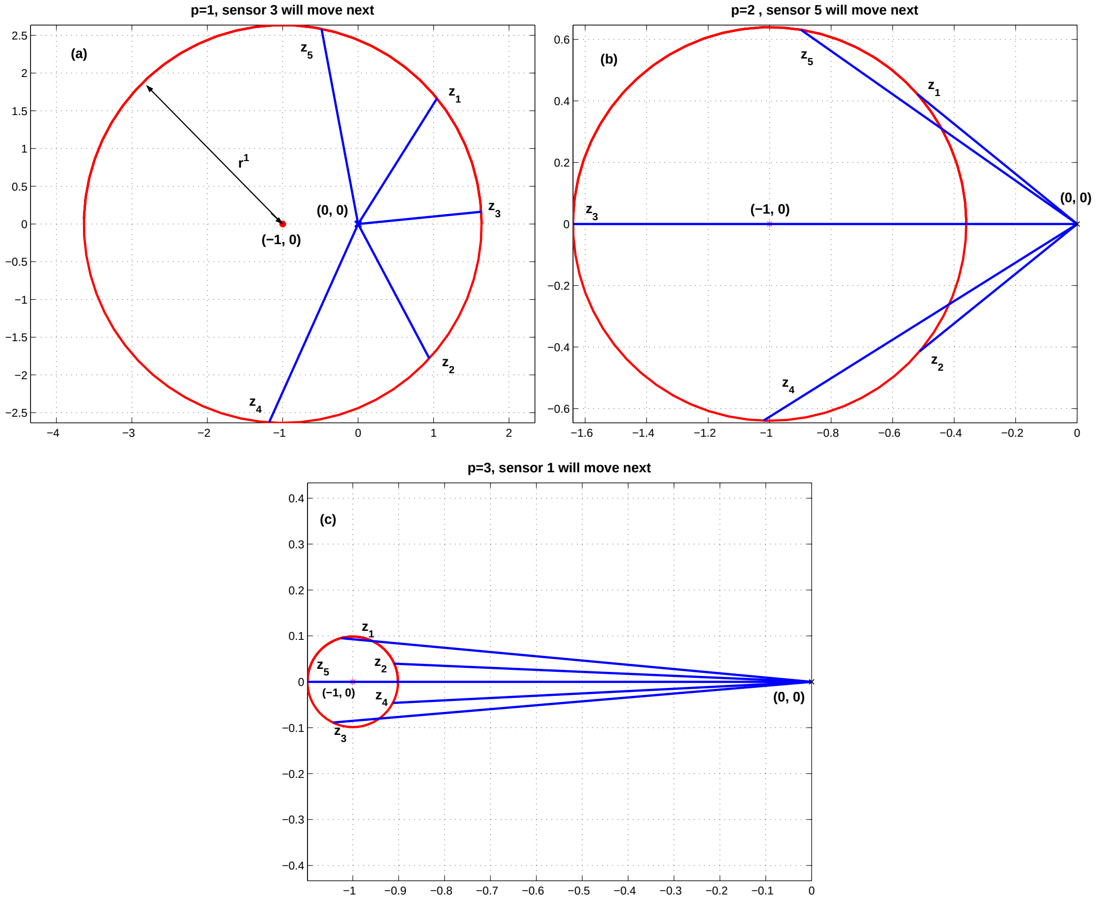

Figure 6-3: Plot of the $\mathbf{z_k}(\theta^p)$ and of the circle of radius $r(\theta^p)$ centered at -1 for $p = 1, 2, 3$ for the case where all $A_k = 1$. Note that the beacon selected for relocation is the one with $|\mathbf{z_k}|$ closest to 1, so that the decrease in error radius r is maximum. This vector is rotated so as to lie on the negative real axis.

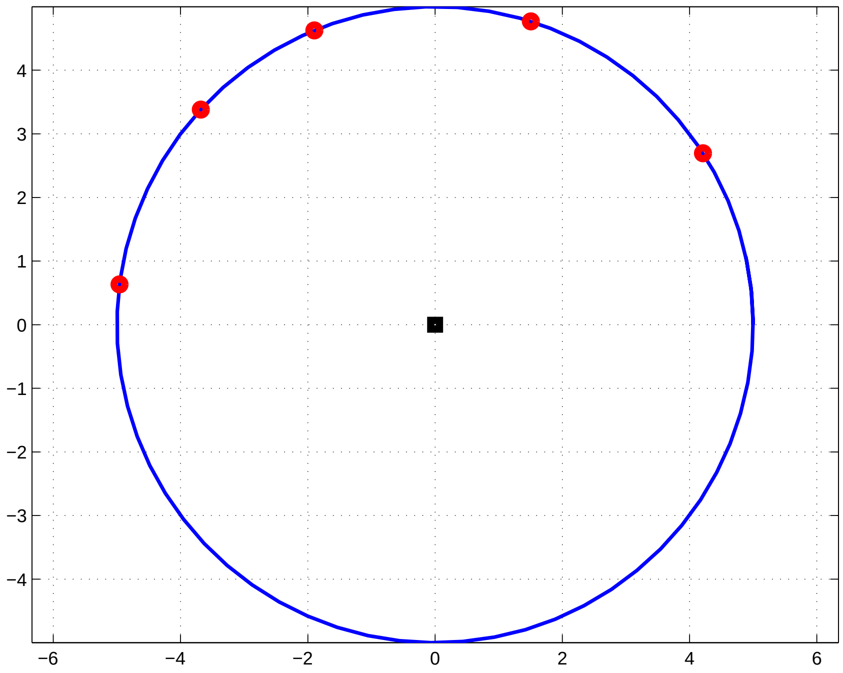

Figure 6-4: Optimal configuration found by RELOCATE when 5 beacons are to be placed and $A_k = 1$.

Another corollary is the optimal value of the PEB.

Corollary 6.8. *The optimal PEB is given by:*

$$\mathsf{PEB}^* = \sqrt{\frac{4\sum_{k=1}^{n_B} A_k}{\left(\sum_{k=1}^{n_B} A_k\right)^2 - \left(\max(0, A_{n_B} - \sum_{k=1}^{n_B-1} A_k)\right)^2}}. \tag{6.17}$$

If all the beacons have the same importance weight $A_k = 1/\sigma_0^2$ (so that $A_{n_\mathrm{B}} < \sum_{k=1}^{n_\mathrm{B}-1} A_k$), we obtain the classic result that the minimum PEB is equal to $2\sigma_0/\sqrt{n_\mathrm{B}}$ [66].

6.3.4 Rate of Convergence

In the next 2 subsections we analyze the performance of RELOCATE. We will show that not only is RELOCATE guaranteed to find the global minimum, but it is also an *efficient* algorithm. In this subsection we derive an approximation to the expected rate of convergence of RELOCATE for the special case when $A_k = 1$ for all k. Again, we are able to derive these results by relying on the coordinate transform introduced earlier.

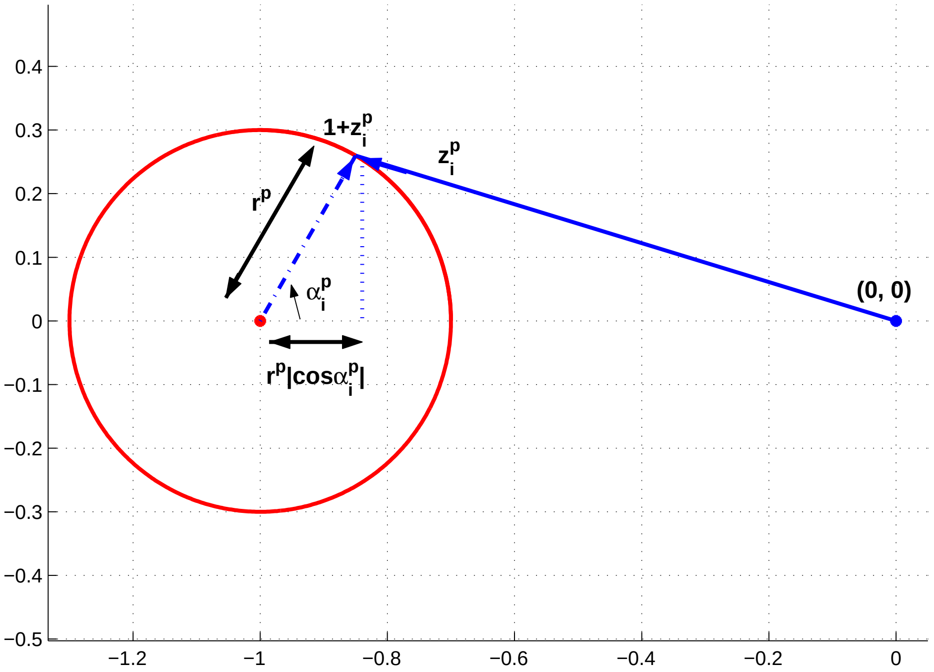

Figure 6-5: Illustration of the approximation of $r^{p+1} = |1 - |\mathbf{z_i}(\boldsymbol{\theta}^p)||$ by $r^p|\cos\alpha_i^p|$.

If beacon i is selected for relocation at iteration p, the new error radius is $r^{p+1} = |1 - |\mathbf{z_i}^p||$, which is obtained by rotating $\mathbf{z_i}^p$ until it lies on the negative real axis and then measuring the distance between the tip of the vector and -1 (Corollary D.5). Assume $r^p \ll 1$, so that $\mathbf{z_i}^p$ is close to the negative real axis as depicted in Figure 6-5. We can then approximate r^{p+1} by projecting $|\mathbf{z}_\mathbf{i}^\mathbf{P}|$ onto the real axis and measuring the distance between the tip of the projection and -1. This distance is equal to $r^p|\cos\alpha_i^p|$, where α_i^p is the argument of the vector $1 + \mathbf{z_i}^p$ (i.e. the angle it makes with the horizontal).

Because at iteration p the beacon with magnitude closest to 1 is selected for relocation, the new error radius is equal to $r^{p+1} \simeq \min_{k=1,\ldots,n_\mathrm{B}}(|\cos\alpha_k^p|)r^p$. If we define the rate of convergence τ^p as the ratio r^{p+1}/r^p, we then have

$$\tau^p \simeq \min_{k=1,\ldots,n_\mathrm{B}}(|\cos\alpha_k^p|). \tag{6.18}$$

At iteration p one beacon (beacon i) lies on the negative real axis (the one relocated at iteration $p-1$), and let us assume that the $n_\mathrm{B} - 1$ other beacons have an equal probability of lying anywhere on the circle of radius r^p centered at -1, so that the α_k^p (for $k \neq i$) are uniform random variables distributed between 0 and 2π. We also assume that these α_k^p are independent. This is a questionable assumption since the $\mathbf{z_k^p}$ are not independent, but one that will enable us to obtain a closed-form solution to the expected rate of convergence which, as we will show, agrees with numerical examples.

The rate of convergence becomes a random variable τ, and its expected value will give the expected rate of convergence of RELOCATE. As shown in Appendix D.4, at each iteration p such that $r^p \ll 1$, the expected rate of decrease of the error radius can be approximated by

$$\mathbb{E}[\tau] \simeq \int_0^1 \left(\frac{2\cos^{-1} x}{\pi}\right)^{n_\mathrm{B}-1} dx. \tag{6.19}$$

On average RELOCATE therefore converges *linearly*.

On Figure 6-6 this theoretical expected rate is plotted as a function of the number of beacons (dashed curve). It tends to 0 as n_B goes to infinity, which means that convergence is faster when more beacons are present. This is to be expected since the probability that $\mathbf{z_i}$ satisfies $|\mathbf{z_i}| = 1$ goes to 1 as the number of beacons is increased. We also performed 100 runs of RELOCATE for these values of n_B and computed the average ratio of decrease, once r^p was below 0.1. We see that as the number of beacons increases, the experimental average rate matches the theoretical value better.

6.3.5 Average Number of Iterations

Finally we use the previous result to estimate the average number of iterations required to reach a certain precision in PEB. When $A_k = 1\ \forall k$, $r^* = 0$ and the minimum value of the PEB for n_B beacons is equal to $\mathsf{PEB}^* = 2/\sqrt{n_\mathrm{B}}$ (6.5). We can then express the relative error in PEB compared to the optimum value PEB^* as

$$\frac{\mathsf{PEB}^p - \mathsf{PEB}^*}{\mathsf{PEB}^*} = \frac{1}{\sqrt{1-(r^p/n_\mathrm{B})^2}} - 1, \tag{6.20}$$

so that it is approximately equal to $(r^p/n_\mathrm{B})^2/2$ for small values of r^p. Let s be the precision required, i.e., the maximum relative error permitted.

We assume we start the algorithm with a radius of 0.1, so that $r^1 \ll 1$. On average we

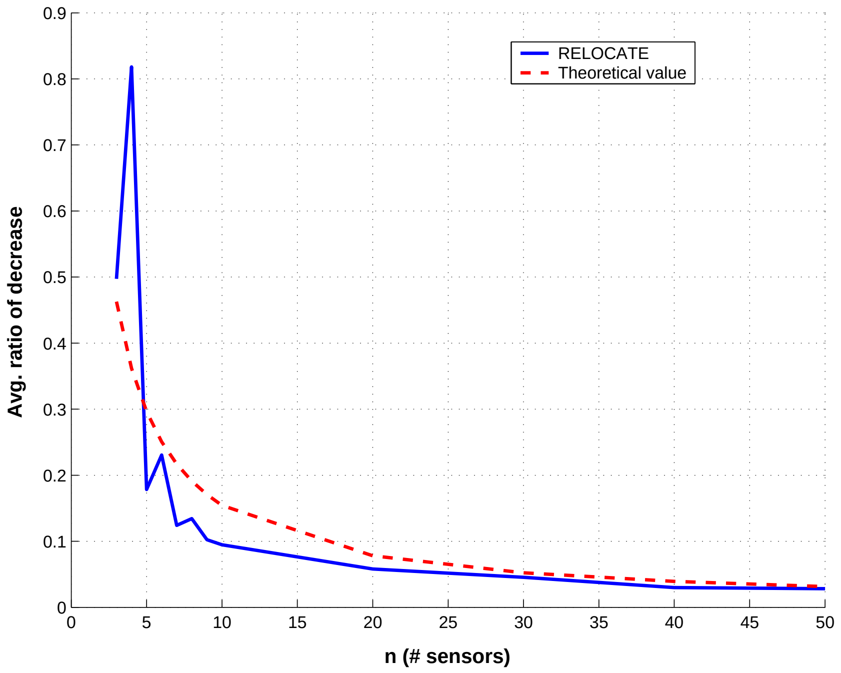

Figure 6-6: Theoretical value of the expected rate of decrease of the error radius (dashed), and experimental value of this rate (solid). The experimental value converges to the theoretical one for large values of n_{B}.

then have $r^p = \tau^{p-1} r^1$. Achieving precision s will then require $1 + \frac{\log(10 n_{\mathrm{B}} \sqrt{2s})}{log\tau}$ iterations on average (not counting the iterations required to bring the radius below 0.1). This number is plotted as a function of the number of beacons for several values of the precision on Figure 6-7. We can see that once the error radius goes below 0.1, the algorithm converges in a few iterations even for high precision requirements. This is even more so as the number of beacons increases, which tends to speed up convergence.

In this section we proved that when a single agent location is considered and when the importance weights are constant, RELOCATE converges to the global minimum. We also showed that it does so efficiently, in a few steps even for high accuracy requirement or large numbers of beacons. Fortunately RELOCATE can easily be applied to more complex cases where the importance weights depend on the beacon position, or when several agent locations are of interest. Given the theoretical guarantees in this simple case, we now proceed with confidence in applying RELOCATE to those more complex cases. In the next section we apply it to the realistic case where the importance weights depend on where the beacons are located.

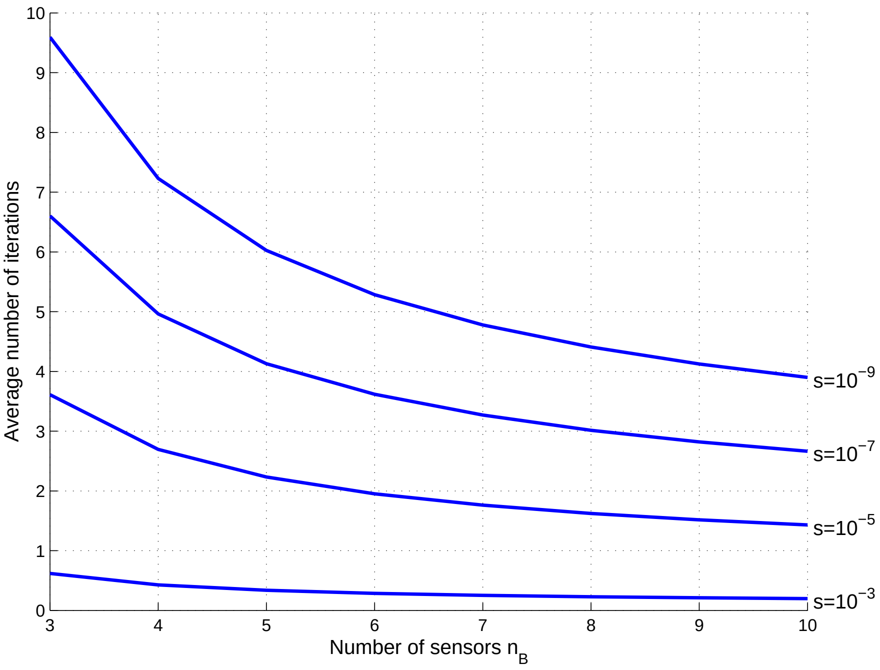

Figure 6-7: Expected number of iterations once the error radius goes below 0.1 as a function of n_{B}. We have plotted this for several values of the precision s.

6.4 Single Agent Location with Varying Importance Weights

So far the importance weights of the beacons were assumed constant, no matter where the beacons were. Although this assumption permitted us to solve the placement problem exactly, it is unlikely to be realistic in real-world scenarios. Since the signal-to-noise ratio (SNR) decreases exponentially with distance, the range measurements will be more accurate (i.e. have lower variance) if beacons and agent are close to one another. Likewise if the agent is inside a building, greater accuracy will be achieved if there is minimal obstruction between the two (as opposed to when several walls, machines, or other objects corrupt the signal). This will result in importance weights that depend on the beacons' locations with respect to the agent.

In this section we consider two cases. In the first one, the importance weights are piecewise constant functions of the angle. The second case is the most general, where the importance weights are allowed to vary arbitrarily. Note that in both cases we no longer have any guarantee of optimality, although in practice we do well.

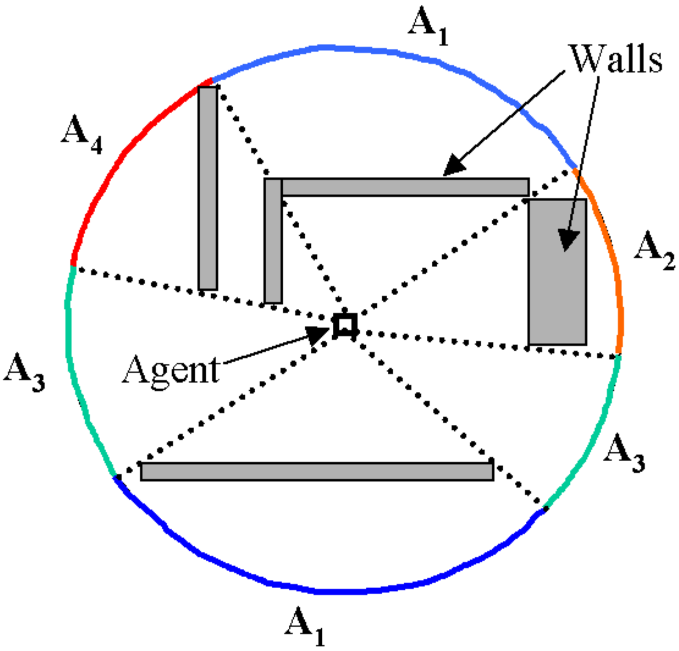

Figure 6-8: An agent is located in the middle of a circular building, which contains several walls. The importance weights take a finite number of values (4 in this example, from A_1 to A_4).

6.4.1 Importance Weights as a Piecewise Constant Function of the Angle

Consider a scenario where the range measurement variance does not depend on the distance, but where obstacles, such as walls, block the line-of-sight (LOS) between the agent and the beacons at certain angles. At these angles, a beacon will be NLOS so its range measurements will be biased, which we modeled with $\beta > 0$ [52–54]. If we have a map of the area, we can predict what value β will take depending on the location of the beacon. The corresponding importance weights will then be a piecewise constant function of the angle. This is illustrated on Figure 6-8, where the agent is in the middle of a circular building, which contains several walls inside. The value of the importance weights on the building boundary changes depending on what obstructs the LOS between agent and beacon.

Let us then divide the interval $[0, 2\pi)$ into L arcs. On arc $C_l = [\underline{c_l}, \overline{c_l})$, the importance weight is constant, equal to A_l (obtained from (5.31)). The generic RELOCATE of Section 6.2.3 can be efficiently adapted to this case. The key is to note that solving step (2) is again easy. The following lemma shows that the minimum of the PEB along one coordinate is obtained at one of $2L + 2$ points: the two extremities of each arc, at the angle specified by (6.7), and at its symmetric with respect to the agent.

Lemma 6.9. *Let* $\mathsf{PEB}(\theta_{i_p})$ *be the PEB when all the angles other than* θ_{i_p} *are kept constant. The angle* $\theta^*_{i_p}$ *minimizing* $\mathsf{PEB}(\theta_{i_p})$ *in step (2) of RELOCATE is given by*

$$\begin{aligned} \theta^*_{i_p} = \arg\min\{&\mathsf{PEB}(\tilde{\theta}_{i_p}), \mathsf{PEB}(\tilde{\theta}_{i_p} + \pi), \\ &\mathsf{PEB}(\underline{c_1}), \mathsf{PEB}(\overline{c_1}), ..., \mathsf{PEB}(\underline{c_L}), \mathsf{PEB}(\overline{c_L})\}, \end{aligned} \tag{6.21}$$

where $\tilde{\theta}_{i_p}$ *is the angle given by (6.7).*

Proof. The proof is given in Appendix D.5. □

This result makes step (2) of RELOCATE easy to solve, so that RELOCATE can again be applied to this problem efficiently. There is no longer any guarantee of global convergence however, but since the algorithm is fast it can be restarted several times from different initial conditions, to eliminate local minima.

Figure 6-9 illustrates a typical result. In this case the internal properties of the building result in 6 different importance weights at the boundary, represented by arcs of different colors. RELOCATE places 5 beacons on the boundary in order to optimally localize an agent placed at the center. Results show that RELOCATE places beacons on arcs with larger importance weight (beacons 1 through 4 are on the arc with $A_6 = 0.87$, beacon 5 is on the one with $A_2 = 0.60$), while spreading them in order to get range measurements from different viewpoints. RELOCATE tries to strike the optimal balance between spatial diversity (well-distributed measurement viewpoints) and measurement quality (arcs with large importance weights). Note that in this particular case all the beacons are located at the extremities of the arcs.

6.4.2 Importance Weights as an Arbitrary Function of the Angle

Consider the same scenario as before, except that now the range measurement variance increases with the distance d to the agent as $\sigma^2(d) = \sigma_0^2 d^\alpha$ (see Chapter 4). To be general we

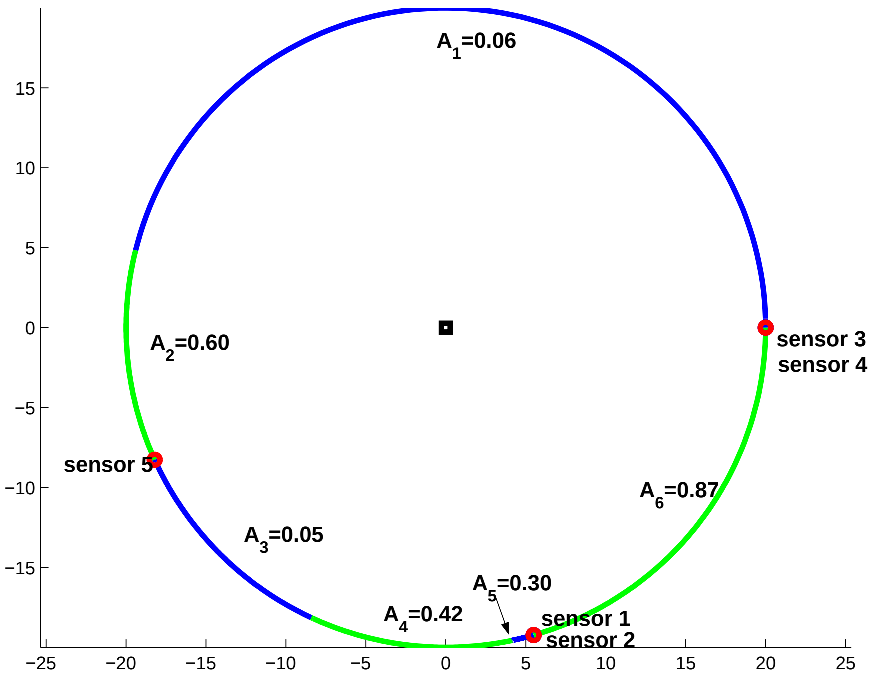

Figure 6-9: Example of RELOCATE when the importance weights are a piecewise constant function of the angle. Beacons tend to be placed on arcs with larger weight, while maintaining some spatial diversity in their placement.

assume that α and β can also be arbitrary functions of the beacon location. The importance weights given by (5.31) can then be any function of the angle. This is the most general case for a single agent location, where the range measurement variance increases with the distance (possibly with different path-loss exponents), and where the beacons become NLOS at certain locations so that $\beta > 0$. Unfortunately this also implies that there is no longer any analytical solution to the minimization of step (2) of RELOCATE, so it must be solved numerically.

Let us consider a square area as shown on Figure 6-10, characterized by $\beta = 0$ and $\alpha = 0$, $\alpha = 0.2$ and $\alpha = 2$. The configurations for 6 beacons obtained through RELOCATE are shown for the cases where the agent is at the center (a)-(c) and at the lower left (d)-(f) of the area. When $\alpha = 0$ the beacons are scattered all around the agent. Note that by symmetry there are many beacon configurations that minimize the PEB in this case. However as α increases they tend to bunch together, so that when $\alpha = 2$ the beacons are evenly split into 2 clusters. We can see here again that RELOCATE strikes the optimal balance between spatial diversity and range measurement quality in order to minimize the

PEB. The beacons are placed close to the agent so that they get range measurements of good quality, while also taking those measurements from different viewpoints (a minimum of two distinct measurement locations are necessary to localize the agent in 2D).

6.5 Multiple Agent Locations

6.5.1 Results with Average PEB

So far we have considered placing beacons in order to minimize the PEB at a *single* location. However in real scenarios we will often want to ensure good localization everywhere in the area, or along a pre-planned path. There are several possible choices of metrics to capture this accuracy, but a natural choice adopted here is to minimize the *average* PEB over the area or the path. RELOCATE can be applied as before, except that in step (2) the average PEB is minimized.

To illustrate this we consider the same square area as before, except that now several agent locations are specified (denoted by squares on Figure 6-11). The beacon configurations given by RELOCATE for different agent locations are also shown. In Fig. 6-11(a) the agent locations are evenly distributed throughout the area, which models the scenario where we want to ensure good localization everywhere (e.g. there is no pre-planned path). In this case RELOCATE places the beacons at regular intervals on the boundary, as intuition would suggest. Interestingly, results do not depend on the value of α.

Fig. 6-11(b)-(c) illustrate a scenario where we only want to ensure good localization in 2 parts of the building. For example the agent may know beforehand that it will only need to inspect 2 rooms inside a building, so good localization accuracy has to be provided there only. The configurations given by RELOCATE differ widely depending on α. If $\alpha = 2$, the beacons are evenly split between the two clusters of agent locations, and for each cluster they again strike the optimal balance between spatial diversity and measurement quality. For $\alpha = 0$ however, the measurement quality is uniform everywhere, so the beacons are more spread out.

Finally in Fig. 6-11(d)-(e) we consider a path inside the area. The agent already knows

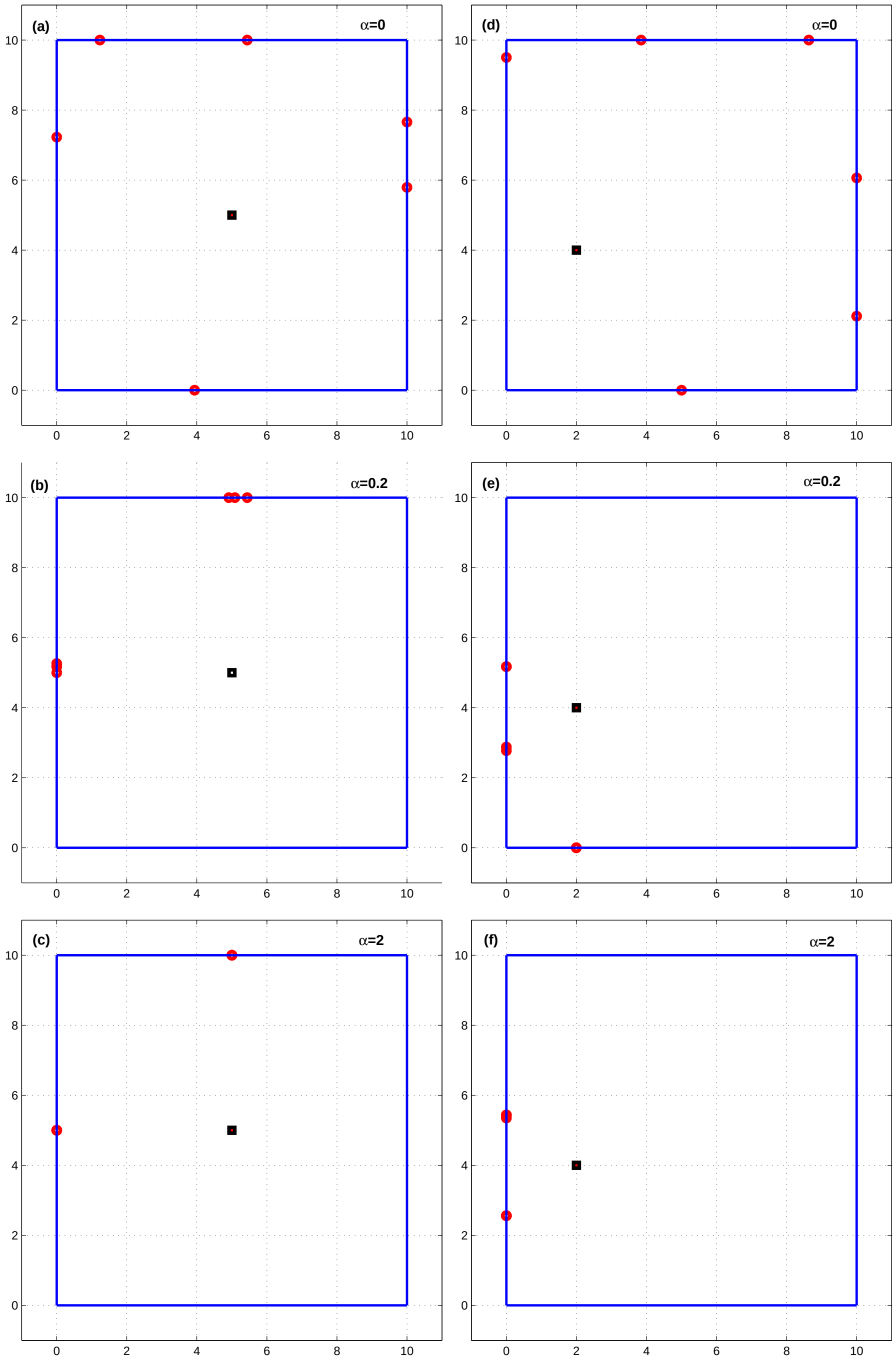

Figure 6-10: Configuration of beacons (denoted by circles on the perimeter) given by RELOCATE. The agent (denoted by a square) is placed at the center (figures (a)-(c)) and at the lower left (figures (d)-(f)). $\beta = 0$ and α takes 3 values in each case: 0, 0.2, and 2.

where it will travel, so it desires to place beacons so as to optimize the localization accuracy along that path. RELOCATE then concentrates the beacons on the wall close to the path when $\alpha = 2$, and spaces them evenly.

Results on the Fort McKenna MOUT Scenario

Let us consider an even more general scenario, where the agent can travel outside the building boundary. In particular we use a map of the Military Operations on Urbanized Terrain (MOUT) site at Fort McKenna to simulate a mission where an agent traveling through the area has to be accurately localized at all times, while the beacons are placed on the exterior walls of different buildings.

In the simulation shown on Figure 6-12 we assume that range measurements can only be made by beacons with LOS to the agent. This simulation can easily accommodate the case where range measurements can be made through buildings, for example by penalizing NLOS measurements by $\beta > 0$.

The path of the agent is shown as black squares, and RELOCATE has to place 8 beacons accordingly. The resulting beacon placement shown on Fig. 6-12 indicates that RELOCATE performed its task well. In particular we note that every agent location is in view of at least 2 beacons, so that localization can be ensured at all times. These good results further indicate that RELOCATE is very flexible to more complex scenarios, where the *average* PEB is minimized and where the agent is not restricted to the interior of the buildings' boundary.

We also note that RELOCATE can easily deal with a *probabilistic* map of agent locations. In many scenarios the agent may not know beforehand where exactly it will go, but it may have an *a priori* density map of its future locations. The area can then be divided into a grid of agent locations, each assigned with a probability given by the density map. The *expected* PEB is then minimized in step (2) of RELOCATE.

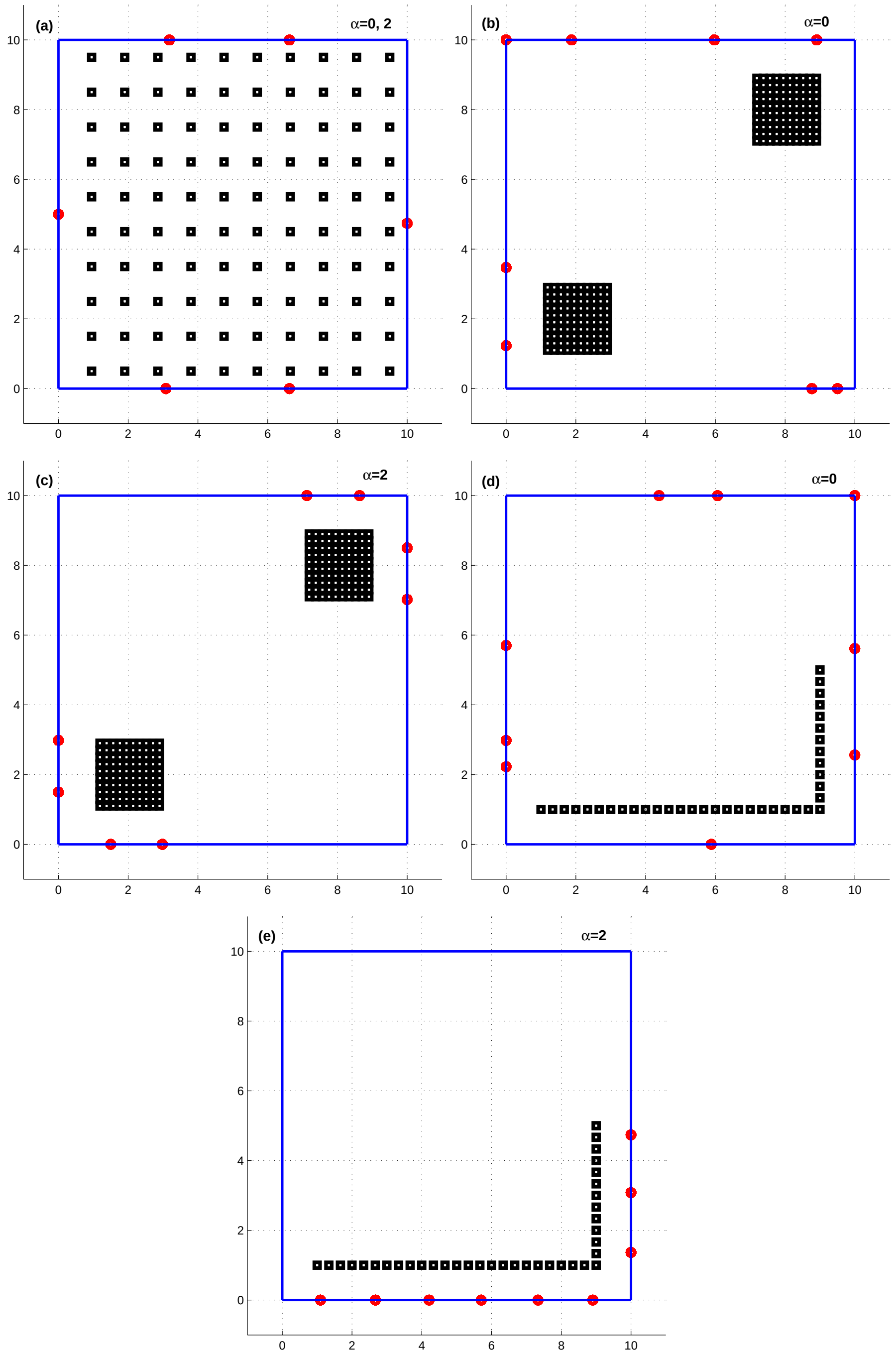

Figure 6-11: Optimal configuration of beacons (denoted by red circles on the perimeter) given by RELOCATE for several agent locations (black squares). When the agent can be anywhere in the building (a), the beacons are evenly distributed on the building's boundary, whether $\alpha = 0$ or 2. The placement varies with α in the other two cases, when only portions of the building (b)-(c) or a path (d)-(e) must be covered.

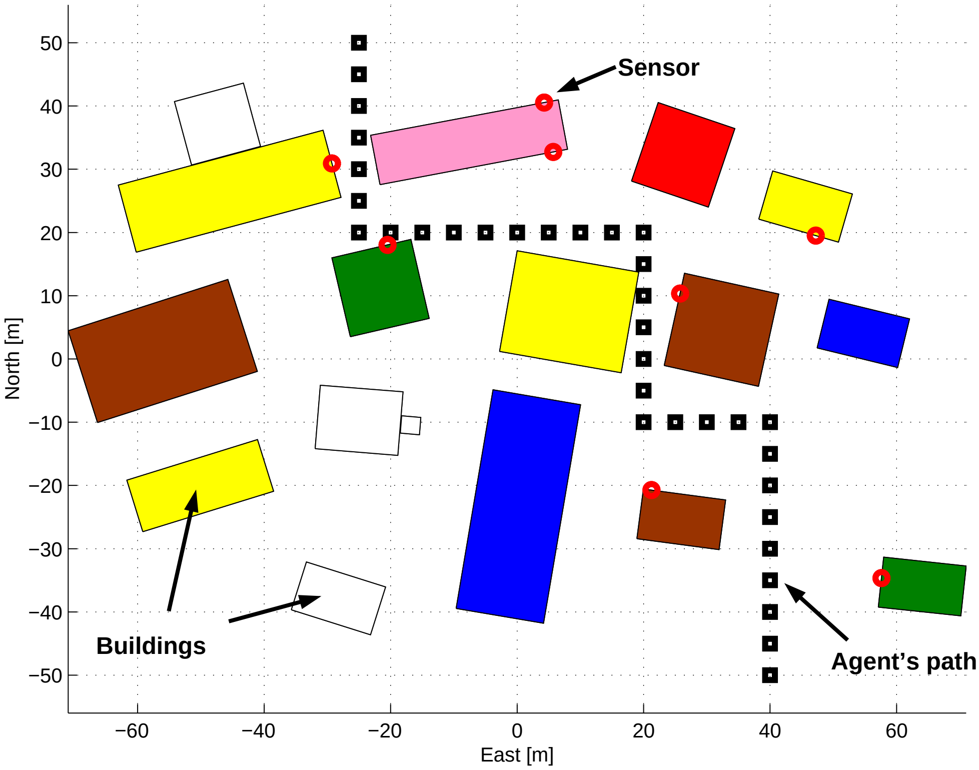

Figure 6-12: Results of RELOCATE for the MOUT site of Fort McKenna. 8 beacons are placed on the boundary of buildings and can make range measurements to the agent when it is LOS. Note that every agent location is in view of at least 2 beacons, ensuring localization at all times.

6.5.2 Benchmarking RELOCATE with Simulated Annealing (SA)

Although there is no longer any guarantee of optimality or efficiency in the case of multiple agent locations with varying importance weights, we show in this section that RELOCATE is still efficient and gives results that are near-optimal. In particular we compare the performance of RELOCATE to that of Simulated Annealing (SA) [62]. SA is a stochastic algorithm, so we expect it to avoid local minima and approach the global minimum. It is also an efficient heuristic algorithm, and it is particularly well-suited to such combinatorial optimization problems [25], so we use it to benchmark RELOCATE. We use the scenario of Fig. 6-11(d) to compare the two methods with 9 beacons to be placed. The average PEB obtained through RELOCATE ($\mathsf{PEB}_{\mathsf{RELOCATE}}$) and SA ($\mathsf{PEB}_{\mathsf{SA}}$) are compared over 100 simulation runs. In Figure 6-13, we plot the frequency histograms of the ratio $(\mathsf{PEB}_{\mathsf{SA}} - \mathsf{PEB}_{\mathsf{RELOCATE}})/\mathsf{PEB}_{\mathsf{RELOCATE}}$ for 3 sets of parameters of the SA that result in 3 different running times. The 3 SA parameterizations respectively took 0.22, 0.94, and 6.74 of the time it took for RELOCATE to complete. Positive values of the ratio indicate that the SA solution is worse than that of RELOCATE.

We see that although RELOCATE is a deterministic algorithm, it yields better results than SA most of the time. For the first two SA parameterizations, RELOCATE produces solutions that are always better than those of SA (the computational cost of SA and RELOCATE in Fig. 6-13(b) are almost similar). Only for longer runs does SA sometimes find better solutions than RELOCATE (Fig. 6-13(c)), but this happens rarely (8% of the time), while the improvement in average PEB is small (2% at most) and the time to completion is much larger than RELOCATE (6.74 more expensive computationally).

We proved before that RELOCATE finds the global minimum efficiently for a single agent and constant importance weights, and this study shows that even in more complex cases (multiple agent locations, varying weights) RELOCATE finds solutions very close to the global minimum. In addition, RELOCATE finds better solutions (indeed solutions within 2% of the minimum given by SA) in less time than SA. We conclude that RELOCATE remains an efficient, near-optimal algorithm even for complex, realistic cases.

6.5.3 Benefit of Using a Placement Algorithm

Figures 6-11(d)-(e) illustrated how optimal beacon configurations vary with the value of α. In this case as α increases, beacons tend to gather closer to the path of the agent. A one-size-fits-all approach which would distribute the beacons evenly on the boundary (which we call UNIFORM) may therefore not be a good idea, at least in certain situations. Let us for example consider the agent path depicted in Fig. 6-11(e) with $\alpha = 2$. We compare three types of placement strategies along the boundary:

- Placement using RELOCATE

- Uniform placement (UNIFORM)

- Random placement (RANDOM)

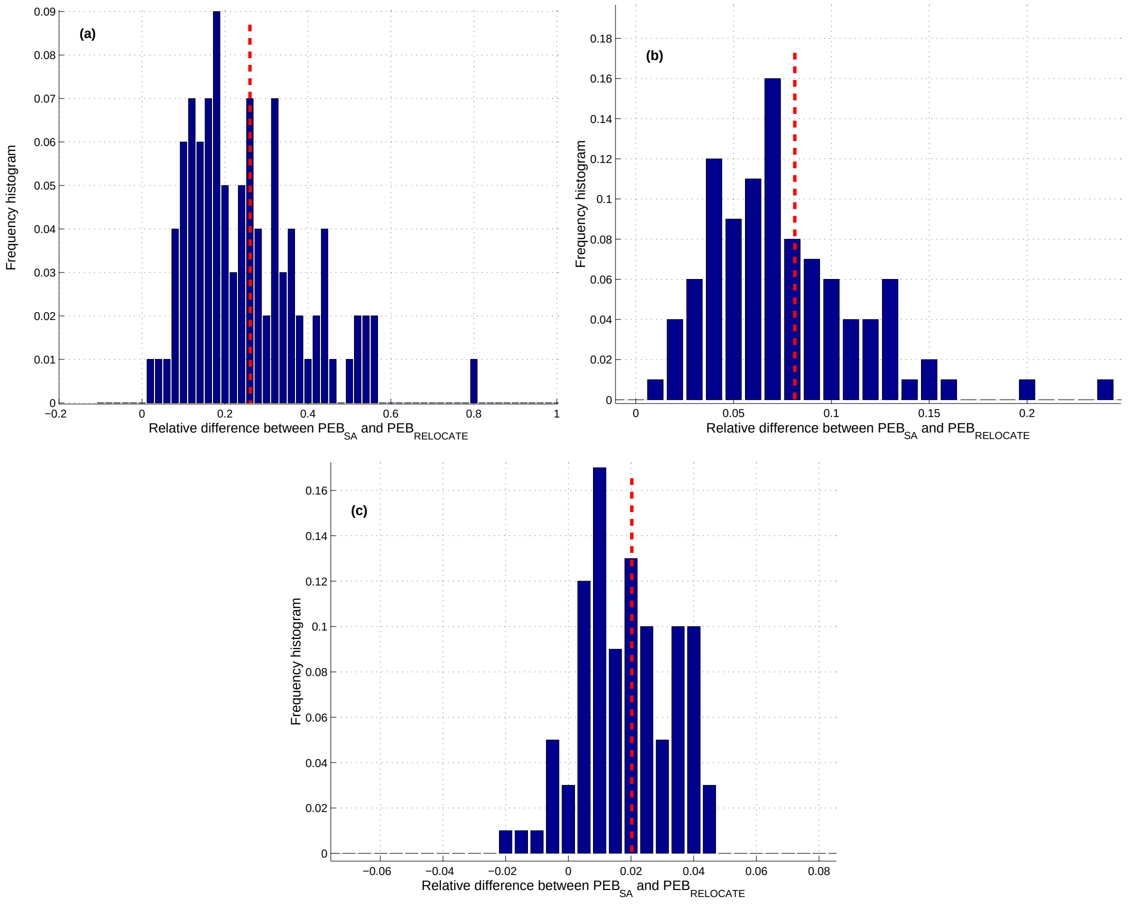

Figure 6-13: Frequency histograms of the relative difference in PEB between the solution given by SA and RELOCATE, with the mean indicated by a dashed line. The SA respectively took a fraction of 0.22 (a), 0.94 (b), and 6.74 (c) of the time it took for RELOCATE to complete.

For the last two strategies the results are averaged over 100 trials. We plot the average PEB resulting from these three methods in Figure 6-14 for different values of the number of beacons. We see that for a given number of beacons, RELOCATE yields an average PEB that is at least twice lower than that obtained by simply distributing the beacons evenly on the boundary. This is important in terms of the number of beacons needed to achieve a certain PEB. For example, to obtain an average PEB below $2mm$, 7 beacons are necessary using RELOCATE, whereas we need 15 with a uniform distribution, and 20 with random placement. Results when beacons are randomly placed are the worst, although not much worse than UNIFORM. The RELOCATE algorithm will therefore use significantly fewer beacons to achieve the same accuracy than a simple, one-size-fits-all approach. This demonstrates

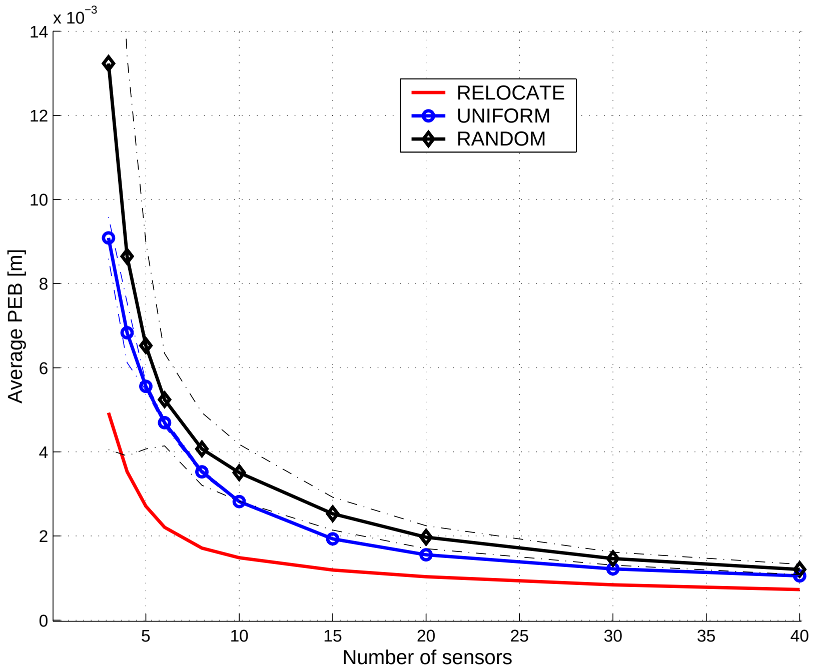

Figure 6-14: Average PEB as a function of the number of beacons for the agent path depicted in the bottom plot of Figure 6-11 with $\alpha = 2$. The average PEB is obtained for 3 placement strategies: RELOCATE, UNIFORM, and RANDOM. The 1-σ envelope is indicated in dashed for the last two.

the importance of planning the beacons configuration optimally.

This is even more dramatically illustrated by considering the Fort McKenna scenario. For different number of beacons, we calculate the average PEB obtained by randomly placing the beacons on the perimeter of the buildings, versus placing them according to RELOCATE. The results are shown on Figure 6-15, and it is clear that the random placement is much worse than RELOCATE, especially when the number of beacons is small. To better visualize this, on Figure 6-16 we plot the ratio between the average PEB obtained by random placement and that by RELOCATE. We can see that RELOCATE typically beats random placement by several orders of magnitude.

6.6 RELOCATE in Phases I, II, and III

The flexibility of RELOCATE makes it readily applicable to all 3 phases of the network planning.

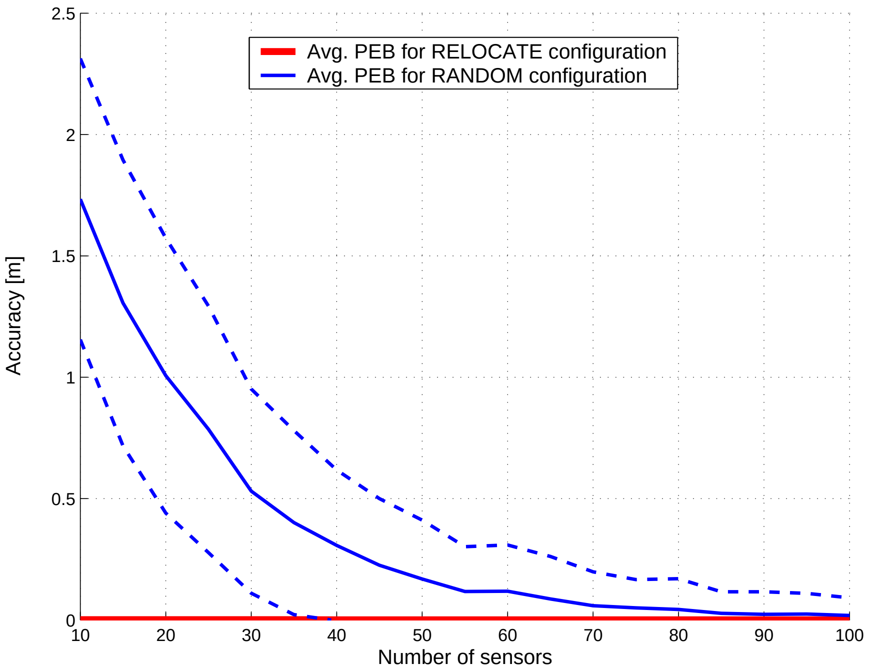

Figure 6-15: Average PEB as a function of the number of beacons for the agent navigating in Fort McKenna as depicted in Figure 6-12. $\alpha = 0$ and agent and beacons must be LOS in order to measure a range. The average PEB for the RELOCATE configurations is in thick red (it is too small to be distinguished from 0 on this scale), while the average PEB for random placement is in solid blue. The 1-σ envelope is indicated in dashed.

Most of the results used to illustrate RELOCATE in this chapter fall into Phase I.

- A network design is obtained from scratch, without any preexisting solution.
- The PEB accounts for any degree of knowledge the user may have about the environment. If nothing is known, the PEB can simply have constant importance weights. But if a statistical characterization of the building is available, or if the detailed propagation characteristics of the UWB signal can be predicted, then the PEB can easily incorporate it.
- Although the number of beacons is fixed, RELOCATE can be run with different numbers of beacons present, so that the user can decide how many beacons are worth deploying.
- RELOCATE can account for the airdrop inaccuracy. In this case given a nominal drop location and the drop vehicle accuracy characterization, the *expected* PEB is

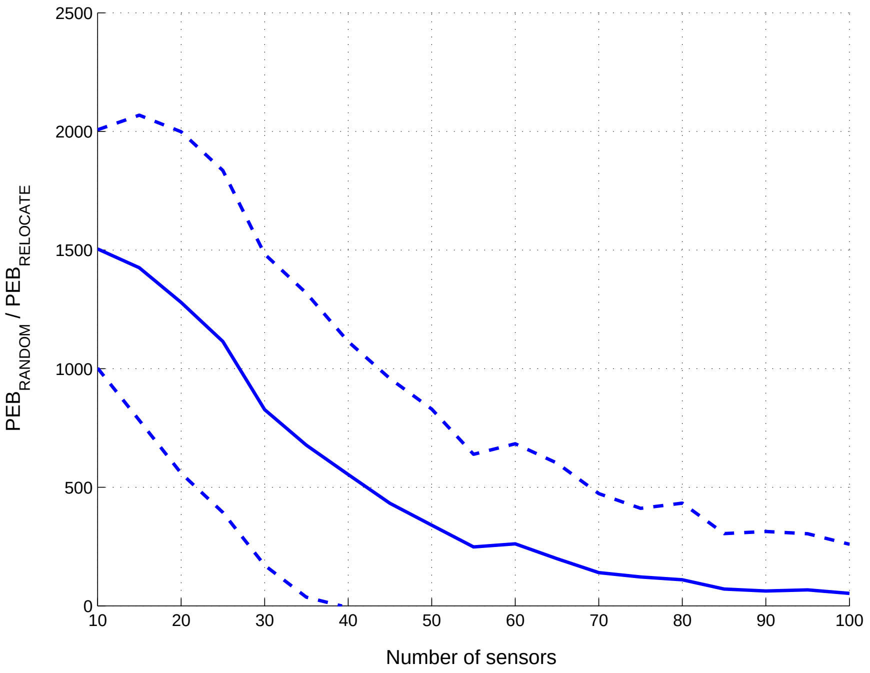

Figure 6-16: Ratio of the average PEB obtained through random placement and that obtained through RELOCATE, from Fig. 6-15. RELOCATE beats random placement by several orders of magnitude, especially for small numbers of beacons.

minimized.

- RELOCATE can even account for modeling uncertainty in Phase I. In particular if the values of α_k or β_k are random variables with known statistics, the expected PEB can again be used.

Not only can RELOCATE produce network designs from scratch, but it can also easily work with some preexisting beacons. Given some beacons already deployed on the ground, it can easily add on to this configuration to minimize the PEB. It is therefore well-suited to Phase II and Phase III, all the more so since, as we have already stressed, it can account for airdrop or modeling uncertainty. If the mission objective change, i.e. if the agent path varies as the mission unfolds, RELOCATE can be used as is to deploy more beacons and so adapt the network to the new requirements (Phase III).

RELOCATE therefore fits perfectly into the Phase I-III framework. The only planning aspects it does not treat are:

- The selection of the optimal drop sequence in Phase II. The suboptimal technique of

Chapter 3 can be used here.

- The optimal operation of the network, i.e. how to actually optimally combine the range measurements to produce a navigation solution. A NLLS method was used for this in Chapter 5, but it did not account for the presence of biases. The next chapter will cover this last piece of the puzzle.

6.7 Conclusion

Although metrics based on the Information Inequality are widely used in the literature to measure the quality of beacon configurations for localization, there have only been a few papers on how to optimally place the beacons to guarantee good localization. In this chapter we have proposed RELOCATE, an iterative algorithm that minimizes the Position Error Bound (PEB). We proved that it converges to the global minimum when the range measurements are unbiased and have constant variances, and we have derived its expected rate of convergence.

We have also shown that it can easily be extended to more realistic cases, where the quality of range measurements depends on the beacon locations. We have also applied RELOCATE to the optimal placement of beacons in order to minimize the *average* PEB over multiple agent locations. In all these cases, RELOCATE attempts to strike the optimal balance between range measurement quality and spatial diversity. Those results have shown that the optimal configuration of the beacons strongly depends on the influence of the environment on the quality of range measurements. A one-size-fits-all placement strategy is therefore inappropriate, a point we illustrated by showing that using RELOCATE can significantly reduce the number of beacons needed to achieve a given accuracy requirement.

We have also shown that RELOCATE converges to solutions that are very close to the global minimum, and that it achieves these results efficiently when compared to Simulated Annealing.

RELOCATE can be used in Phase I, when planning the network from scratch, but it can also work with partial design and it can deal with uncertainty, so it is well-suited for Phases II and III as well. This algorithm can therefore be used in all 3 phases of the deployment strategy in order to robustly deploy the network.

Chapter 7

Monte Carlo Localization Using an IMU and UWB Ranging

7.1 Introduction

In Chapter 6 RELOCATE was presented as an efficient algorithm for placing beacons in order to provide optimal localization to an agent. As was shown, the task of beacon placement in all phases of the deployment can be optimized using RELOCATE. In this chapter we consider another aspect of Phase III, namely the optimal operation of the network, that is how to optimally combine the range information coming from the different beacons in order to obtain an accurate position estimate. We assume that the locations of the beacons are known, and that the agent carries a strap-down Inertial Measurement Unit (IMU) that provides attitude rates and instant accelerations. These ranges, coupled with the IMU information, are then used to update the agent position.

If the ranges to the beacons were accurate, then three beacons would be sufficient to determine the agent position with accuracy using any triangulation technique. But we have shown in Chapter 4 that although UWB ranging performs better than other signals indoor, the range measurements are still corrupted by the presence of *positive biases* resulting from NLOS propagation. This difficulty cannot be resolved at the hardware level, and it has been referred to as a fundamental limiting factor in UWB ranging performance [65].

If accurate UWB channel modeling was available, then it may be possible to predict these biases throughout the environment. In that case the predicted biases can be subtracted from the range measurements, thus overcoming this difficulty. However in many applications such modeling will not be accurate, and often not available at all (e.g. military scenarios where the area to explore is unmapped). Another way around this difficulty can be found in GPS receivers, where a voting system identifies corrupted range measurements and discards them [60]. This method works because of the redundancy in range measurements (typically more than 4 satellites are in view at all time). However in indoor environments we will not usually have this luxury since most (if not all) range measurements are likely to be biased. If biased or otherwise corrupted measurements are discarded, we could potentially be left without any measurements at all. This is illustrated on Figure 7-1, where an Extended Kalman Filter (EKF) [7, 8] with outlier rejection is used to estimate the position of the agent along a path. The range measurements from 2 beacons are positively biased at all times, but the outlier rejection rejects measurements that differ too much from their expected value (assuming no bias is present). Very quickly the outlier rejection rejects measurements from one beacon, and the position estimate of the agent diverges. We therefore cannot afford rejecting all biased measurements. An indoor localization method oblivious to this difficulty is unlikely to perform well.

We show in this chapter that these biases can instead be estimated jointly with the position of the agent, allowing for the range measurements to be corrected and the agent accurately localized. We use Monte Carlo Localization (MCL) specifically to capture the non-linearity of bias transitions [57].

Related work

We focus on localization using a set of beacons with *a priori* known positions; we will not deal with the problem of Simultaneous Localization And Mapping (SLAM) of the beacon position: the goal is to have a system that provides the same capabilities as GPS, but in a GPS-denied environment. We do, however, assume that no map of the physical layout building is available to infer the signal bias.

Ours is not the first approach to using Monte Carlo techniques for inference in sensor

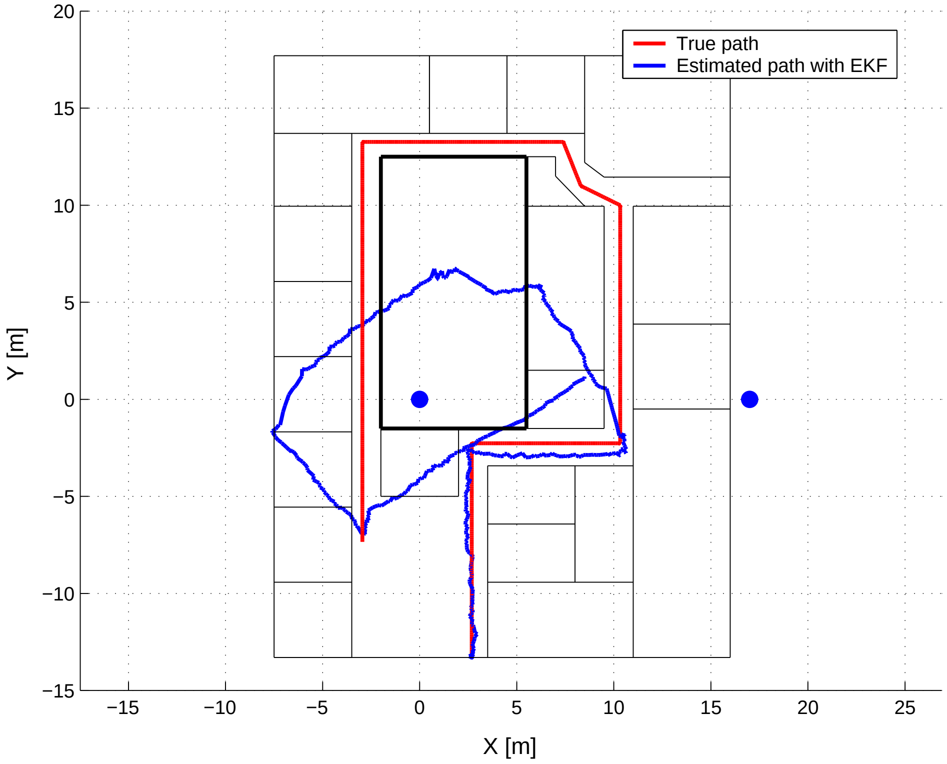

Figure 7-1: Illustration of the failure of classical methods such as this EKF with outlier rejection. Since all measurements are biased, this method based on rejecting "biased" measurements leads to divergence.

networks, however, we believe that ours is the first range-based localization to demonstrate robustness to hidden biases. In contrast, Ladd *et al.* [63] build an explicit model of the spatial distribution of biases which they then use to build an HMM and solve the global localization problem. They are able to use 802.11 signals to localize a laptop based only on beacon measurements, however, they do require the substantial initial training phase.

Smith *et al.* [88] avoid many of the bias issues by explicitly using two different range sensors in the Cricket system. Their results indicate that the EKF approach with outlier rejection provides accurate localization in the face of bias given a sufficient number of beacons (several per room). However, they do encounter periodic EKF failures; their assumption is that having the moving agent transmit additional signals to the beacons is sufficient to recover from the EKF failure.

Biswas *et al.* [12] take a similar approach to ours in factoring the likelihood model, however, they are attempting to solve a fundamentally different problem in assessing the presence of enemy agents in the sensor network.

This chapter is organized as follows. In Section 7.2 we describe MCL. In Section 7.3 we use the results of Chapter 4 to model the biases in the particle filter. Section 7.4 contains the results from experiments utilizing real-world data from Chapter 4.

7.2 Monte Carlo Localization (MCL)

Our goal is to estimate the position of an agent moving indoor based on IMU instantaneous rate measurements and range measurements from a set of UWB beacon. As before we restrict ourselves without loss of generality to two dimensions. We also assume for simplicity that the translational velocity of the agent is known at all times. The goal of MCL will be to optimally estimate the state of the agent as it moves through the area. In particular the state vector $\mathbf{x}(t)$ of the agent at time t contains the following variables:

- $x(t)$, $y(t)$ are the coordinates of the agent;
- $\theta(t)$ and $\dot{\theta}(t)$ are its heading angle and heading angle rate

Since we are in a 2D environment with known velocity, the onboard IMU needs only be a rate gyro. The measurements received at time t are then:

- $z_{\dot{\theta}}(t)$, the heading angle rate of the agent from the IMU;
- $\mathbf{r}(t) = \{r_1(t)\ \ldots, r_{n_\mathrm{B}}(t)\}$, the ranges from the n_B beacons.

We acknowledge the degree of simplification of our modeling compared to a 6 degrees of freedom (6-DOF) modeling, but our goal here is to provide a “proof-of-concept” that can then be generalized to more faithful 6-DOF models [67].

Our goal is to maintain the probability distribution of the state $\mathbf{x}(t)$ conditioned on all the previous measurements. This distribution is called the posterior distribution, and is denoted at time t as $p(\mathbf{x}(t)|z_{\dot{\theta}}(t), \mathbf{r}(t), z_{\dot{\theta}}(t{-}1), \mathbf{r}(t{-}1), \ldots)$. We first seek to obtain a recursive equation linking the posterior of $\mathbf{x}(t)$ to that of $\mathbf{x}(t-1)$. We write:

$$
\begin{aligned}
p\left(\mathbf{x}(t)|z_{\dot\theta}(t),\mathbf{r}(t),z_{\dot\theta}(t-1),\mathbf{r}(t-1),\ldots\right) &= \frac{p(\mathbf{x}(t),z_{\dot\theta}(t),\mathbf{r}(t)|z_{\dot\theta}(t-1),\mathbf{r}(t-1),\ldots)}{p(z_{\dot\theta}(t),\mathbf{r}(t)|z_{\dot\theta}(t-1),\mathbf{r}(t-1),\ldots)} \qquad (7.1)\\
&= \alpha\cdot p(z_{\dot\theta}(t),\mathbf{r}(t)|\mathbf{x}(t))\cdot p(\mathbf{x}(t)|z_{\dot\theta}(t-1),\mathbf{r}(t-1),\ldots) \qquad (7.2)\\
&= \alpha\cdot p(z_{\dot\theta}(t)|\mathbf{x}(t))\cdot p(\mathbf{r}(t)|\mathbf{x}(t))\cdot p(\mathbf{x}(t)|z_{\dot\theta}(t-1),\mathbf{r}(t-1),\ldots), \qquad (7.3)
\end{aligned}
$$

where α is a normalization term. In (7.1) we have used the definition of conditional probabilities. In the middle term of the right-hand side of (7.2) we have assumed that the measurements at time t only depend on the current state $\mathbf{x}(t)$ (by the Markov assumption). Finally in (7.3) we assumed that, given $\mathbf{x}(t)$, the measurements $z_{\dot\theta}(t)$ and $\mathbf{r}(t)$ are independent. We can express the right-most term of (7.3) as a function of the probability distribution of the previous state $\mathbf{x}(t-1)$:

$$
p(\mathbf{x}(t)|z_{\dot\theta}(t-1),\mathbf{r}(t-1),\ldots) = \int p(\mathbf{x}(t)|\mathbf{x}(t-1))\cdot p(\mathbf{x}(t-1)|z_{\dot\theta}(t-1),\mathbf{r}(t-1),\ldots)d\mathbf{x}(t-1), \tag{7.4}
$$

where again we used the Markov assumption to write $p(\mathbf{x}(t)|\mathbf{x}(t-1),z_{\dot\theta}(t-1),\mathbf{r}(t-1),\ldots) = p(\mathbf{x}(t)|\mathbf{x}(t-1))$. We therefore obtain the following recursive estimator, known as *Bayes filter* [31,38]:

$$
\begin{aligned}
p(\mathbf{x}(t)|z_{\dot\theta}(t),\mathbf{r}(t),z_{\dot\theta}(t-1),\mathbf{r}(t-1),\ldots) = \alpha\cdot p(z_{\dot\theta}(t)|\mathbf{x}(t))\cdot p(\mathbf{r}(t)|\mathbf{x}(t))\times\\
\int p(\mathbf{x}(t)|\mathbf{x}(t-1))\cdot p(\mathbf{x}(t-1)|z_{\dot\theta}(t-1),\mathbf{r}(t-1),\ldots)d\mathbf{x}(t-1)
\end{aligned} \tag{7.5}
$$

Given the posterior at time $t-1$ and a set of measurements, we can use Bayes' filter to obtain the posterior at time t.

A conventional EKF is often used to maintain the distribution $p(\mathbf{x}(t))$ by linearizing the prediction and measurement functions and modeling the noise terms as Gaussian [7,8]. These two constraints allow the distribution over $\mathbf{x}(t)$ to be approximated as a Gaussian. The advantage to such a representation is that the distribution can be represented using only a small number of parameters (a mean vector and covariance), and updated very efficiently. If, however, the prediction and measurement models are not easily linearized and the noise terms are not Gaussian, then the EKF typically does an increasingly poor job of approximating the true distribution over $\mathbf{x}$, often leading to filter divergence [29, 74].

An alternate technique for representing $p(\mathbf{x}(t))$ is to maintain a set of sample states drawn from the distribution [31]. Good techniques exist to sample from distributions even when the distribution itself cannot be represented, and statistics such as the mean, variance and higher order moments of the distribution can be computed directly from the samples instead of from the distribution parameters.

MCL [31, 37] is a form of robot localization using Importance Sampling [40], in which samples from a target distribution $p(\mathbf{x})$ are desired but cannot be drawn directly. Instead, samples are drawn from some known proposal distribution $q(\mathbf{x}(t))$ that does permit direct sampling. Each sample is assigned an importance weight $p(\mathbf{x}(t))/q(\mathbf{x}(t))$, and the set of weighted samples can be used in place of the distribution $p(\mathbf{x}(t))$. In sampling problems where the target distribution changes over time, the sample weights can be updated directly to reflect the new distribution, although finite numerical precision can cause the sample weights to converge eventually to 0. To avoid this problem, in Importance Sampling Resampling [44], the weighted samples are periodically resampled according to their weights to generate a new set of uniformly weighted samples.

In the localization problem, our target distribution $p(\mathbf{x}(t))$ is the posterior $p(\mathbf{x}(t)|z_{\dot{\theta}}(t), \mathbf{r}(t), z_{\dot{\theta}}(t\text{–}1), \mathbf{r}(t\text{–}1), \ldots)$. Under the assumption that we do not have a parametric representation of this distribution (i.e. we do not use an EKF), we maintain a set of particles where the i^{th} particle $\mathbf{x}^{[i]}(t)$ is written:

$$\mathbf{x}^{[i]}(t) = [x^{[i]}(t), y^{[i]}(t), \theta^{[i]}(t), \dot{\theta}^{[i]}(t)]. \tag{7.6}$$

We assume that we have a motion simulator $p(\mathbf{x}(t)|\mathbf{x}(t-1))$ that models the dynamics of our agent, just as we do in Kalman filtering (prediction model). We can therefore *sample from the prediction model*, that is given a state at $t-1$, the simulator can be used to propagate the state to instant t. We also assume that we have a model for the measurements (again, just as in Kalman filtering) in the form of likelihood functions for the IMU $p(z_{\dot{\theta}}(t)|\mathbf{x}(t))$ and for the range measurements $p(\mathbf{r}(t)|\mathbf{x}(t))$. The measurement likelihood is simply the product of those two functions.

Given the posterior at time $t-1$, let us use as proposal distribution $q(\mathbf{x}(t))$ the distribution of predicted states:

$$q(\mathbf{x}(t)) = \int p(\mathbf{x}(t)|\mathbf{x}(t-1))p(\mathbf{x}(t-1)|z_{\dot{\theta}}(t-1), \mathbf{r}(t-1), \ldots)d\mathbf{x}(t-1). \tag{7.7}$$

The importance weights are then given by:

$$\begin{aligned} p(\mathbf{x}(t))/q(\mathbf{x}(t)) &= \frac{p(\mathbf{x}(t)|z_{\dot{\theta}}(t), \mathbf{r}(t), \ldots)}{\int p(\mathbf{x}(t)|\mathbf{x}(t-1))p(\mathbf{x}(t-1)|z_{\dot{\theta}}(t-1), \mathbf{r}(t-1), \ldots)d\mathbf{x}(t-1)} && (7.8) \\ &= \alpha \cdot p(z_{\dot{\theta}}(t)|\mathbf{x}(t)) \cdot p(\mathbf{r}(t)|\mathbf{x}(t)), && (7.9) \end{aligned}$$

where we have used Bayes filter equation (7.5) to simplify the second equation. We therefore conclude that *the importance weights* $p(\mathbf{x}(t))/q(\mathbf{x}(t))$ *are simply proportional to the measurement likelihood.*

We summarize the MCL algorithm below.

0) Initialize the set of n_S sample $\left\{\mathbf{x}^{[i]}(0)\right\}_{i=1\ldots n_\mathrm{S}}$ by drawing them according to some initial distribution, $t = 1$.

1) For $i = 1 \ldots n_\mathrm{S}$ do:

   (j) Propagate $\mathbf{x}^{[i]}(t-1)$ one step in time to obtain $\hat{\mathbf{x}}^{[i]}(t)$ using the prediction model $p(\mathbf{x}(t)|\mathbf{x}(t-1))$.

   (jj) Assign importance weight to $\hat{\mathbf{x}}^{[i]}(t)$ given the measurements $z_{\dot{\theta}}(t)$ and $\mathbf{r}(t)$ using the likelihood function $p(z_{\dot{\theta}}(t)|\hat{\mathbf{x}}(t)) \cdot p(\mathbf{r}(t)|\hat{\mathbf{x}}(t))$.

2) Resample n_S new samples $\left\{\mathbf{x}^{[i]}(t)\right\}_{i=1\ldots n_\mathrm{S}}$ from $\left\{\hat{\mathbf{x}}^{[i]}(t)\right\}_{i=1\ldots n_\mathrm{S}}$ according to their importance weights.

3) $t \leftarrow t+1$, go to 1).

If the measured ranges are unbiased, the measurement models $p(\mathbf{r}(t)|\mathbf{x}(t))$ and $p(z_{\dot{\theta}}(t)|\mathbf{x}(t))$ will be sufficient to unambiguously determine the agent position given measurement of three beacons.

7.3 Modeling of the Biases, the Agent, and the Measurements

As we have previously argued, the ranges are positively biased (Chapter 4). In order to accurately estimate the position of the agent, it is necessary to *jointly* estimate the agent position and the set of beacon biases $\mathbf{b}(t) = \{b_1(t), \ldots, b_{n_\mathrm{B}}(t)\}$. The gyro, in addition to giving noisy measurements, also has a bias that evolves over time. We therefore also included the gyro bias $g(t)$ as a state variable to be estimated. The i^{th} particle $\mathbf{x}^{[i]}(t)$ is then written:

$$\mathbf{x}^{[i]}(t) = \left[x^{[i]}(t), y^{[i]}(t), \theta^{[i]}(t), \dot{\theta}^{[i]}(t), g^{[i]}(t), b_1^{[i]}(t), \ldots, b_{n_\mathrm{B}}^{[i]}(t)\right], \tag{7.10}$$

where $b_j^{[i]}(t)$ is the bias estimate of the j^{th} beacon for particle *i*. In order to estimate the biases $\mathbf{b}(t)$ with the particle filter, we need a proposal distribution (cf. equation 7.7) and a likelihood model (cf. equation 7.9). Our likelihood model does not change as a result of estimating biases, but we need to modify the proposal distribution to model how these biases change over time.

7.3.1 Beacon bias dynamics modeling

We construct the probabilistic model of beacon bias transition by looking at actual UWB measurements. Recall from Chapter 4 that the biases are positive, and that they are locally constant and tend to change in discrete increments. As the agent moves, we can therefore expect the biases to either remain constant or change to a significantly different value. Let us then first define the rate of beacon bias change r_{change}, which is the expected number of times the bias changes per second and per beacon. This rate typically depends on the environment, but we found that the particle filter performed well even for inaccurate values of r_{change}. In our experimental results, we set $r_{change} = 1$ Hz. At each time step dt, the probability that the bias of the j^{th} beacon changes for a given particle is $r_{change}dt$. When this is the case, the particle's j^{th} bias value is assigned a uniformly distributed random number between 0 and $b_j^{[i]} - \epsilon$ or between $b_j^{[i]} + \epsilon$ and β, where β is the maximum value the bias can take, and ϵ is a positive number smaller than β. The role of ϵ is to ensure that the bias change is sufficiently large, in order to model the fact that when the bias changes, it is likely to change to a value significantly different from the previous one (e.g. in Figure 4-4, it jumps from 1m to 1.5m). Our bias motion model is therefore a uniform distribution notched about the current bias, where the notch has a width of 2ϵ. In practice, we set ϵ to be equal to 3 standard deviations of the beacon measurement noise, in order to ensure we do not mistake measurement noise for bias change.

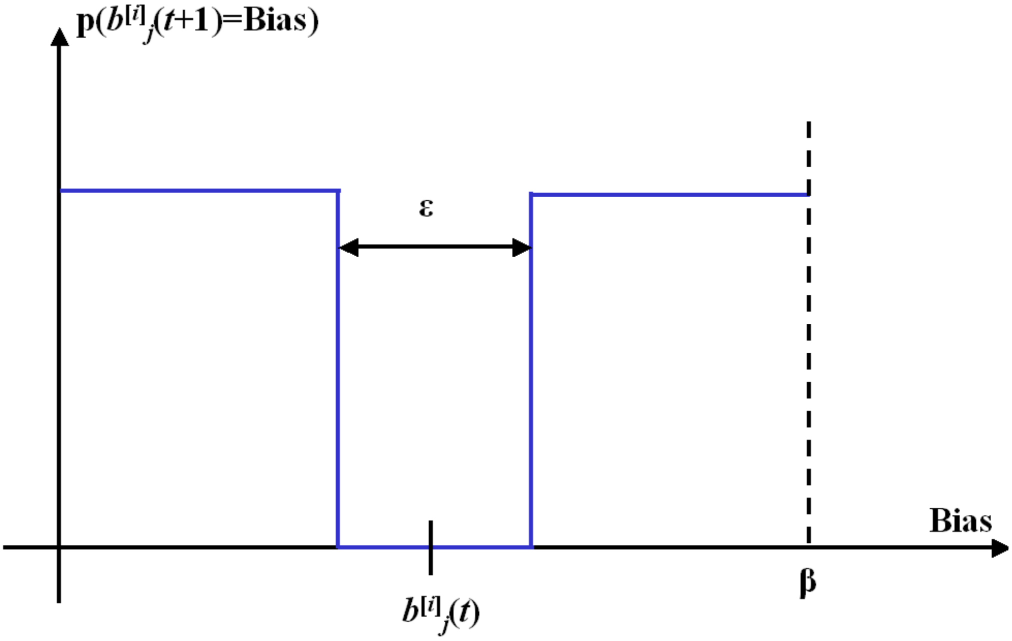

Figure 7-2: At time $t+1$, with probability $r_{change}dt$ a new bias value is sampled for $b_j^{[i]}$ according to the pictured distribution. Otherwise $b_j^{[i]}$ retains its current value.

We may have been tempted to simply model the bias transition as a Gaussian centered at the current bias value. This would have been advantageous since it would have enabled us to use an EKF, which is more efficient than MCL. However such modeling does not agree with the measurements presented earlier, especially with the discrete nature of bias transitions. Simulations showed in fact that the resulting estimation performs poorly. The non-linearities of bias transition must be kept in order to achieve accurate localization, and therefore makes our problem non-amenable to the EKF format.

7.3.2 Agent dynamics modeling

The probabilistic model of the agent's dynamics is as follows:

$$g^{[i]}(t) = g^{[i]}(t\text{–}1) + N(0, \sigma_g^2) \tag{7.11}$$

$$\dot{\theta}^{[i]}(t) = \dot{\theta}^{[i]}(t\text{–}1) + N(0, \sigma_{\dot{\theta}}^2) \tag{7.12}$$

$$\theta^{[i]}(t) = \theta^{[i]}(t\text{–}1) + \dot{\theta}^{[i]}(t\text{–}1)dt \tag{7.13}$$

$$x^{[i]}(t) = x^{[i]}(t\text{–}1) + V\cos(\theta^{[i]}(t))dt \tag{7.14}$$

$$y^{[i]}(t) = y^{[i]}(t\text{–}1) + V\sin(\theta^{[i]}(t))dt \tag{7.15}$$

where V is the (known) translational velocity, and σ_g and $\sigma_{\dot{\theta}}$ are the standard deviations of the noise for the gyro and the gyro bias, respectively. $N(0, \sigma^2)$ is a normally distributed random number with mean 0 and variance σ^2. In our experimental results, $\sigma_g = 0.001 rad.s^{-1}$ and $\sigma_{\dot{\theta}} = 3 rad.s^{-1}$.

7.3.3 Measurement model

The measurement model assumes not only that the gyro measurements are independent of the beacon measurements, but also that the beacon measurements are independent of each other. Therefore, for a given particle $\mathbf{x}^{[i]}(t)$, the likelihood is a product of $n_{\mathrm{B}} + 1$ factors. The first factor is the likelihood of the current measurement $z_{\dot{\theta}}(t)$, which we model as a normal distribution:

$$p(z_{\dot{\theta}}(t)|\mathbf{x}(t)) = N(\dot{\theta}, \sigma_{z_{\dot{\theta}}}^2), \tag{7.16}$$

where $\sigma_{z_{\dot{\theta}}} = 0.1 rad.s^{-1}$.

The n_{B} remaining factors are the likelihoods of the current n_{B} range measurements given each particle's position $\mathbf{x}(t)$ and its n_{B} beacon biases $\{b_1^{[i]}(t), b_2^{[i]}(t), \ldots, b_{n_{\mathrm{B}}}^{[i]}(t)\}$,

$$p(r_i(t)|\mathbf{x}(t), b_i(t)) = N(||\mathbf{x}(t) - \mathbf{y}_i(t)|| - b_i(t), \sigma_{rf}^2), \tag{7.17}$$

where $\mathbf{y}_i(t)$ is the location of the i^{th} beacon, and the σ_{rf} is the error in the range sensor taken from the sensor specification. This was specified as $\sigma_{rf} = 0.025m$ for our experiments. This value in fact was a slight underestimate; using the range measurements from the equal-bias points on the plateau $(x < -4m)$ in the room experiment (Figure 4-4) showed the variance to be $\sigma_{rf} = 0.03m$.

7.4 Experimental Results

In this section we describe our simulation and then compare the results for different cases. We only had access to actual UWB signal data for a single transmitter, so the second signal is simulated. Although in theory three beacons are necessary to unambiguously localize an agent in 2D, two beacons are sufficient in our scenario because we use an IMU and the agent's initial position is known, so that the ambiguity is removed.

7.4.1 Experimental Setup

An agent travels at a constant speed of $1m.s^{-1}$ along a corridor in an office building. It carries a rate gyro providing its instantaneous heading angle rate, and receives range measurements from two beacons at a rate of 10Hz. The range measurements from those two beacons are positively biased by $\mathbf{b}(t)$. For the bias of one beacon, we use the actual range measurements from a physical transmitter (as described in Chapter 4). For the second beacon, we simulated range measurements and a bias profile. The map of the environment is shown in Figure 7-3. We assume that the agent knows its initial state since prior to entering the building it can use GPS to determine its exact location. We show the results of the estimation using the particle filter for the following cases:

- Case 1: the agent uses only its IMU to estimate $\mathbf{x}(t)$;
- Case 2: the agent uses its IMU and beacon range to estimate $\mathbf{x}(t)$, but does not estimate $\mathbf{b}(t)$;
- Case 3: the agent uses its IMU and beacon range, estimating the joint distribution over $\mathbf{x}(t)$ and $\mathbf{b}(t)$.

We also compare our results against an EKF with outlier rejection.

Case 1 Results

The accuracy of the IMU used in the simulation is poor, and the resulting estimated path (Figure 7-3(a)) does not track the true trajectory, as the gyro noise quickly dominates.

Case 2 Results

The results shown in Figure 7-3(b) are better than in Case 1, but the path estimate oscillates about the true trajectory, yielding a position error of more than $1m$ on average. This is due to the fact that the ranges have unmodeled bias. The position estimate oscillates as the particle filter tries to best adjust to the changing biased measurements.

Case 3 Results

In this case the beacon biases are estimated and the results of Figure 7-3(c) show a very close tracking of the true path. The reason for such good performance comes from the fact that the beacon biases are being estimated: the particle likelihoods incorporate the biases, so the measurements can still be used with confidence. Figure 7-4 shows the bias estimates for beacon 1 and beacon 2.

We also show an EKF with outlier rejection. Since only two beacons are present, there is not enough redundancy to ensure sufficient measurement updates, so the results are poor, as shown in Figure 7-3(d).

7.4.2 Systematic Comparison

Cases 1, 2 and 3 and the EKF with outlier were performed over 100 simulation runs. The mean and variance of the average position error over the path are shown on Figure 7-5. The mean error of Case 1 is not surprisingly much worse than the other two mean errors, and Case 3 provides localization with twice the accuracy of Case 2. Estimating the beacon biases doubles the average accuracy in this example. The level of accuracy achieved (a few tenths of centimeters) is satisfactory for indoor navigation.

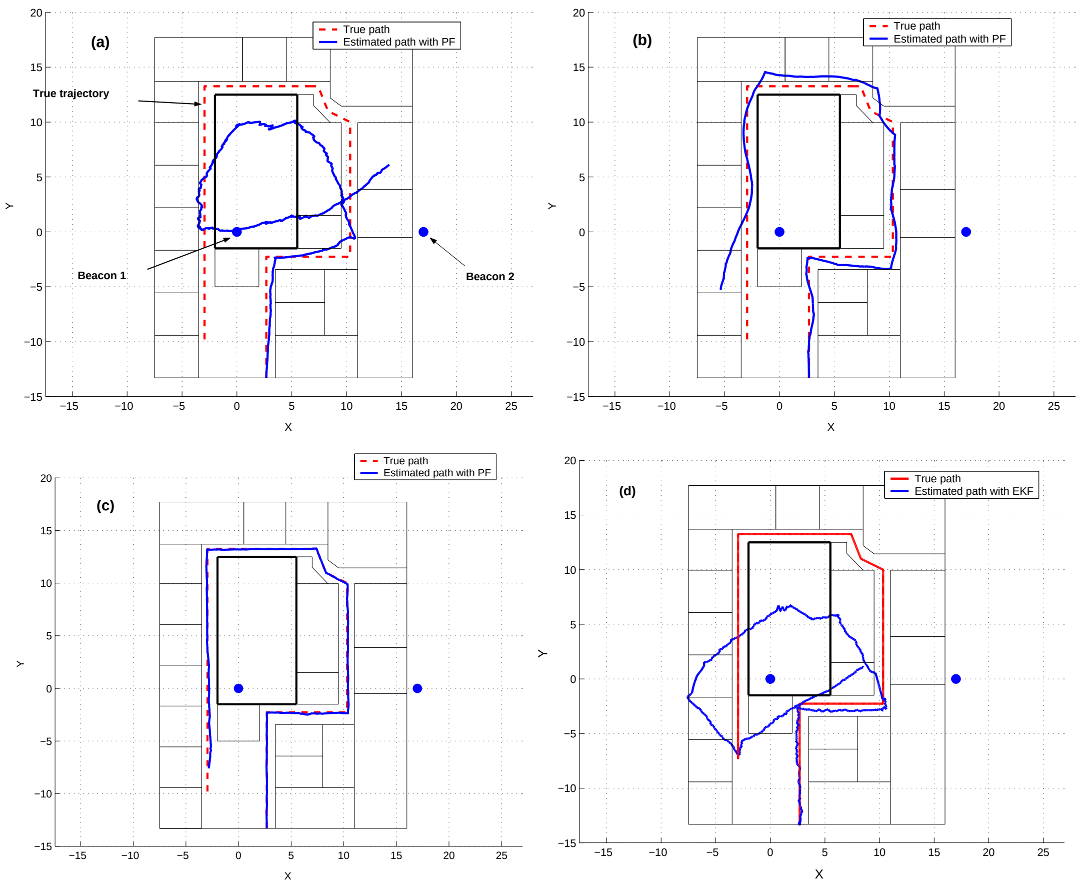

Figure 7-3: Path estimate for (a) Case 1 (IMU only), (b) Case 2 (IMU and beacons, but no bias estimation), (c) Case 3 (full pose and bias estimation) (d) EKF with outlier rejection.

It should be noted that in this example the beacon biases were limited to a maximum of 1m, so that the errors are fairly limited. If the biases increase (in Figure 4-4 some equal 1.5m), then we can expect Case 2 to perform increasingly worse as the estimator has to reconcile measurements increasingly inconsistent. However we expect the PF with bias estimation (case 3) to remain accurate even for large biases, since they are estimated and removed from the measurements.

7.5 Conclusions

Once the beacons are on the ground, we have shown that localization can be improved by coupling measurements from an onboard IMU and ranges from UWB beacons. This method has been shown to overcome the presence of biases in the range measurements, which is the main factor currently limiting the accuracy of indoor localization systems. We showed that a particle filter can be used to simultaneously estimate the state of the agent and the beacon measurement biases. Experimental results incorporating real and simulated UWB measurements demonstrated the efficacy of the particle filter approach, which enabled us to localize the agent within a few tenths of a meter. Although these results are based on a limited set of data, we believe that this example shows the validity of our concept to provide a realistic solution to the challenge of accurate indoor geolocation.

The results are indeed encouraging, but more work is needed. In particular more UWB range measurements should be performed in order to further validate the beacon bias model. Then a 6-DOF model should be implemented with a 3-axis IMU. Since this will introduce many more states, we recommend implementing a Rao-Blackwellized particle filter (or Mixture Kalman filter) [23]. This variant factors some states in a KF format, while retaining the "particle" format for some. In our case the agent's states can be propagated and updated using an EKF, while the beacon biases can remain in the particle format. This should yield significant computational savings, while keeping the crucial non-linear modeling of the bias transitions.

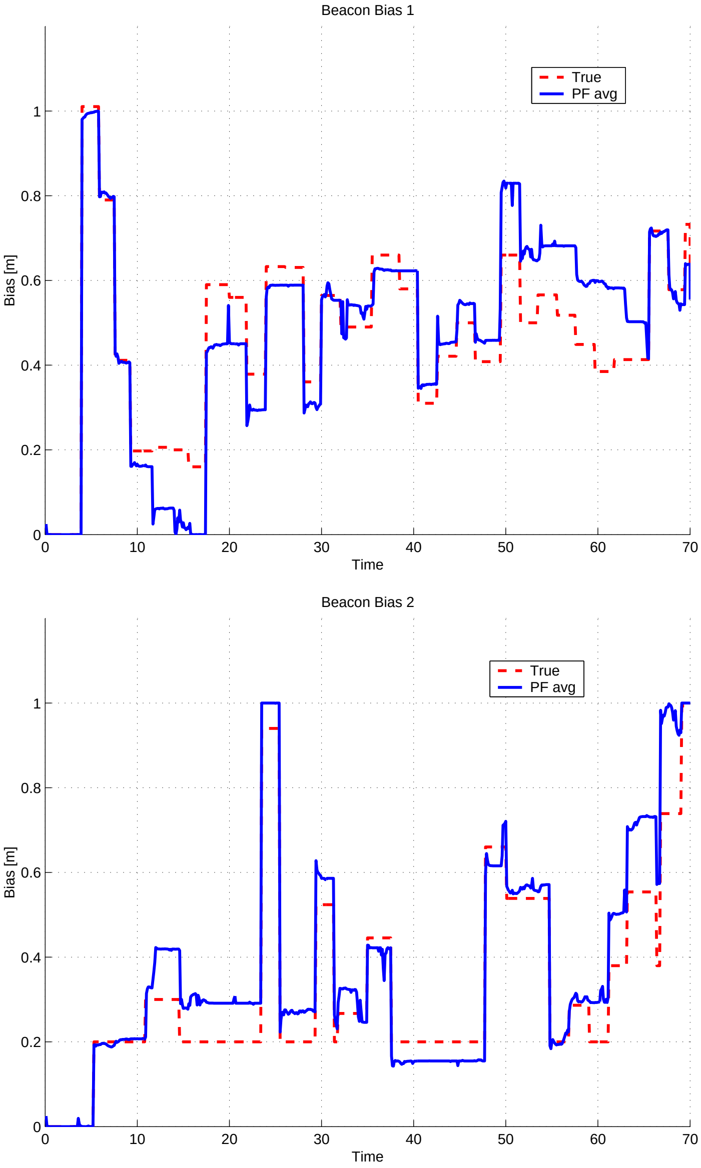

Figure 7-4: Estimates of the biases of beacon 1 (left) and beacon 2 (right) are shown by the solid line. Note that the bias of beacon 1 comes from actual UWB measurements described in Chapter **??**. Beacon 2 is simulated, and its bias is artificial induced.

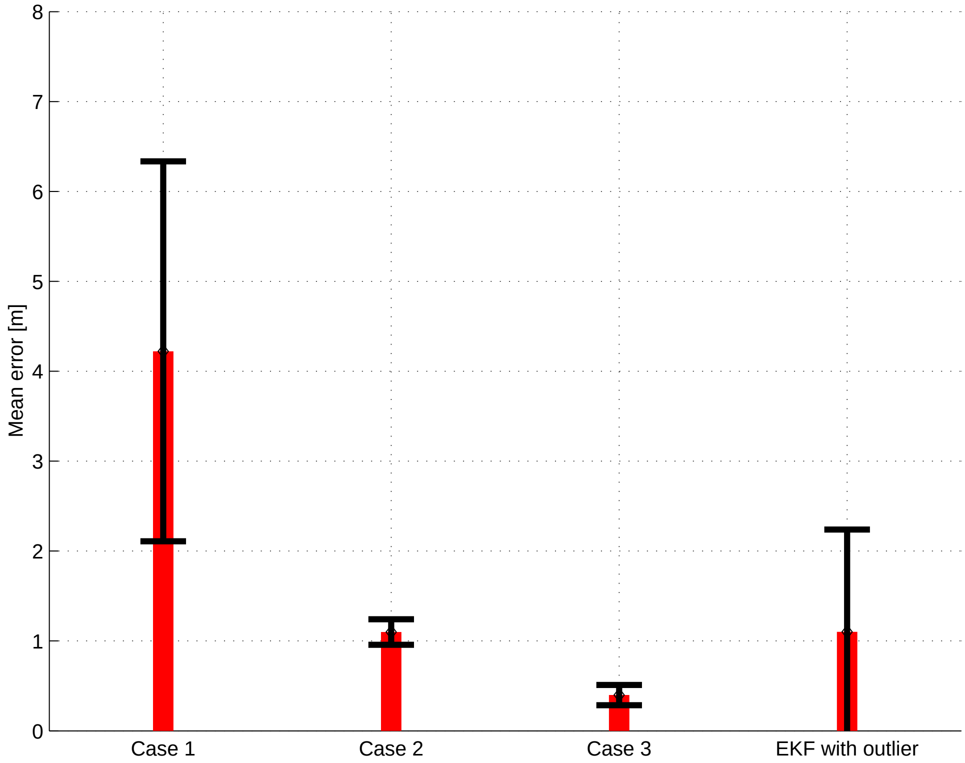

Figure 7-5: Mean error between the true position and its estimate after 100 simulation runs for Cases 1, 2, 3 and the EKF. The standard deviation is shown.

Chapter 8

Summary and Recommendations

8.1 Summary

We proposed a strategy for the planning and deployment of WSNs divided into 3 phases, Phases I, II, and III. This framework addresses the optimization challenges of the planning process, and takes into account the major sources of uncertainty inherent to the deployment process, so that the WSN deployed on the ground performs as best as possible.

We first outlined a general purpose algorithm for Phase I, the Multi-Objective Genetic Algorithm. We showed that its flexibility to different mission scenarios with multiple objectives makes it ideally suited to the planning of Phase I in general. We also showcased the benefits of Phase I on a few examples. Because the performance of WSNs strongly depends on the layout of the network, using a smart algorithm to place the sensors yields far superior designs than approaches without planning.

We then proposed an approximation algorithm for Phase II. Numerical studies indicated that the deployment of the network can have a strong influence on the performance of the WSN once it is on the ground. We showed in particular that the choice of the drop sequence is important and thus justifies the need for Phase II.

We followed these general results by a specific WSN application, localization in GPS-denied environments. We started by motivating the use of UWB signals to infer ranges between agents and beacons. Because of its high resolution, UWB is able to provide high ranging accuracy, even in the kind of harsh environments typically found in GPS-denied areas (indoor, urban canyon, etc.). However we also noted some limitations of UWB ranging, namely the presence of positive biases in the range measurements due to NLOS propagation, and the increase of the measurement variance with distance.

Using these observations we then derived the Position Error Bound (PEB), a lower bound on the localization accuracy. The PEB accounts for both the bias in range measurements and the increase of the variance with distance. Not only is the PEB useful in general to analyze the localization accuracy throughout an area, but it can also serve as the objective to be minimized when placing the beacons.

We then developed a placement algorithm, RELOCATE, which finds the beacon configuration that minimizes the PEB. RELOCATE was shown to converge to the optimal solution in the special case when a single agent location is considered and when the range measurements are unbiased and have a constant variance. Numerical studies showed that it also works very well in more general scenarios, where the average or expected PEB is minimized over a path, and where the quality of the range measurements varies with the agent's position. We showed that this algorithm can be used for the initial planning of Phase I, as well as when incremental planning is needed, such as in Phases II and III. The benefits of planning the network with RELOCATE were again outlined.

Finally a Monte Carlo localization (MCL) method was developed to fuse the range measurements from the UWB beacons and inertial measurements from an onboard IMU, so as to optimally estimate the state of an agent moving through a GPS-denied environment. This algorithm was shown to provide high positioning accuracy, notably thanks to its ability to accurately estimate the bias embedded in the range measurements. Once the network is deployed on the ground, the MCL then optimally uses the resources at hand to complete the mission objective. All aspects of Phases I, II, and III were covered on this localization example, and for each phase we demonstrated the importance of smart algorithm in planning, deploying, and operating the WSN.

In conclusion we have shown in this thesis that smart algorithms should be used for the planning and deployment of WSN, as they can greatly influence the performance of the network once it is on the ground.

8.2 Recommendations for Future Work

We have shown that the deployment process has a strong impact on the performance of the WSN, thus the need for Phase II. However we have limited our analysis to an approximation algorithm, especially when it comes to the selection of the drop sequence. More work is needed on this problem.

More UWB range measurements should also be performed in order to validate the modeling described in Chapter 4. In particular a better understanding of the bias dynamics and the increase of variance with distance is crucial. Another interesting question is whether velocity information can be inferred from the UWB signals, potentially through Doppler effect.

Some practical issues remain unresolved. In particular we assumed perfect clock synchronization between the transmitter and receiver, an assumption unlikely in practice. Solutions to this problem like round-trip measurement have been mentioned, but they need to be implemented and validated in practice. At a deeper level, understanding and quantifying how the synchronization error impacts the accuracy will help in designing a practical system.

The RELOCATE algorithm seems to have a lot of potential. In particular its ability to incorporate uncertainty in modeling and beacon placement should be exploited. Also, because the PEB is flexible to range measurements with vastly different characteristics (from narrow-band to UWB), RELOCATE can be used to plan localization networks that use any kind of signals, from 802.11 to UWB.

The MCL presented in Chapter 7 should be generalized to a more realistic scenario, where the agent has 6 degrees of freedom and carries a 3-axis IMU with gyroscopes and accelerometers. In order to cope with the increase in variables, a Rao-Blackwellized particle filter (or Mixture Kalman filter) should be implemented, and we expect it to perform well. Also, the method can be extended to several agents moving through the same area. These agents may have the ability to estimate their relative distance and share position information. The impact of this collaborative localization is interesting and deserves more attention.

Finally a complete simulation of the WSN planning and deployment process should be implemented to demonstrate the combined power of all the phases. A specific scenario such as the localization example can be used, and an actual deployment of agents and beacons can be performed.

Appendix A

Additional Case Studies for the MOGA

A.1 Case Scenarios 3 and 4

In the third study (CS3), movements in and out of a circular area are to be detected. This is akin to CS1, except that there are no threats to the sensors. In the fourth one (CS4), a square area is to be uniformly covered by the WSN, so that every point in it can be monitored. These case studies (including CS1 and CS2 described in chapter 3) are basic mission scenarios that can form the building blocks (or primitives) of a more complex mission scenario. For example, consider a factory suspected to manufacture dangerous chemicals. An area close to it may have been identified as a possible location for waste disposal, and chemical sensors are to cover it uniformly in order to analyze the soil (CS4). Also, seismic sensors placed around the plant can detect the movements in and out of it, which gives and indication of the plant's activity (CS2 or CS3, depending on the threat level). All these sensors must then relay their data to a HECN placed nearby. A Multi-Objective Genetic Algorithm (MOGA) can then be used to find non-dominated network designs. It is shown that the exact same MOGA works in all cases, which shows its flexibility and its possible use for more complex missions built upon these basic building blocks.

A.2 Results for CS3

The facility considered is a circle of radius 3, centered on the origin, as shown in Figure A-1. It is assumed that there are no threats to the sensors, and the goal is to obtain as much coverage with as few sensors as possible. Like in CS2, the sensors cannot be placed inside the facility.

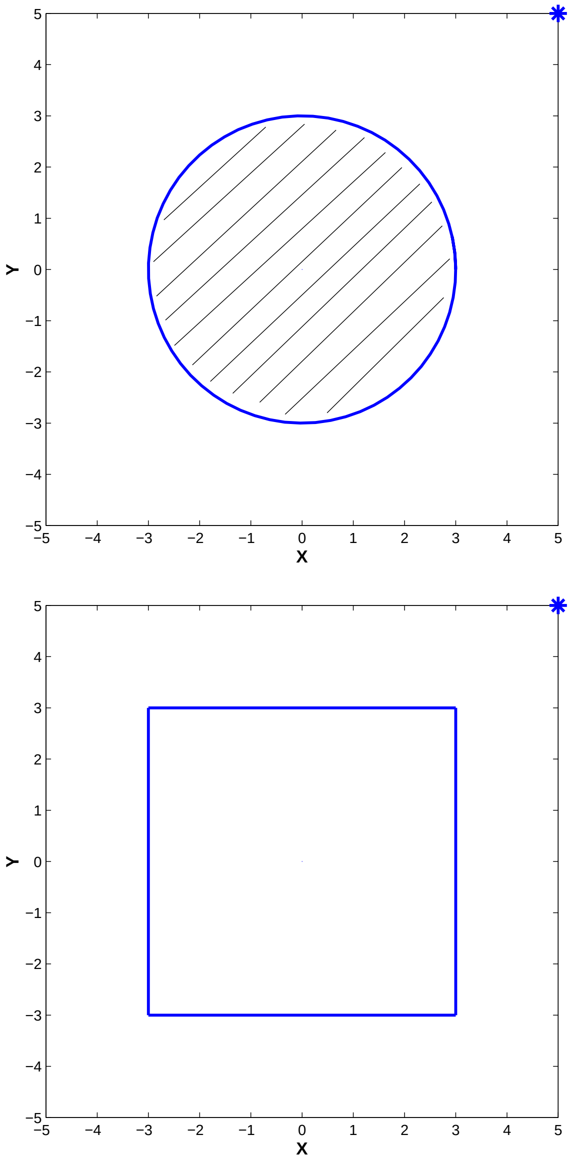

Figure A-1: Map for CS3 and CS4, with the HECN in the upper-right corner.

The first objective is the detection, as defined in CS2 (3.5). Lines are generated around the circle, and the calculation is the same as before. The second objective is the number of sensors, to be minimized. Again, these two objectives are competing since placing more sensors yields a better coverage.

The objectives graph is shown in Figure A-2. The points part of the APF are linked by a line. The APF shows the trade-off between coverage of the area versus number of sensors. This optimization took 20 minutes to complete. Figure A-2 also displays the layout with almost complete detection (obtained with 6 sensors). These results are intuitive, the MOGA provides the actual numerical values of the trade-off. The goal of this example is to show that the same MOGA than in CS1 and CS2 can be used. This is shown again in the following example.

A.3 Results for CS4

The area is a square of side 6, centered on the origin, as shown on Figure A-1. The two objectives are the coverage from CS1 (3.2) and the number of sensors. The goal is to cover uniformly the whole square, that is the WSN must be able to detect any movement inside the area, not only in and out. These two objectives are again competing.

The objectives graph produced by the MOGA is shown in Figure A-3, and 3 non-dominated designs are plotted on Figure A-4. These results agree again with the intuition, showing that the MOGA finds optimal layouts (notice the symmetrical structures that appear). This illustrates the flexibility of this algorithm to the modeling and the choice of objectives. This optimization took 25 minutes to complete.

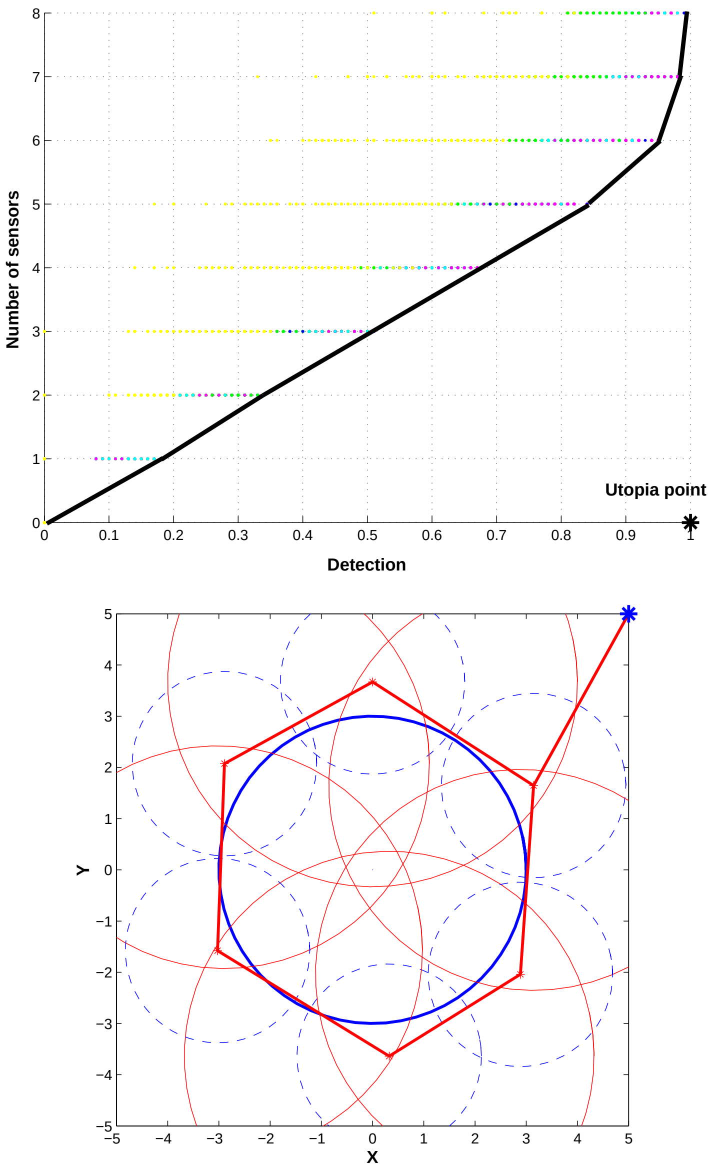

Figure A-2: Objectives graph for CS3 and design with largest detection.

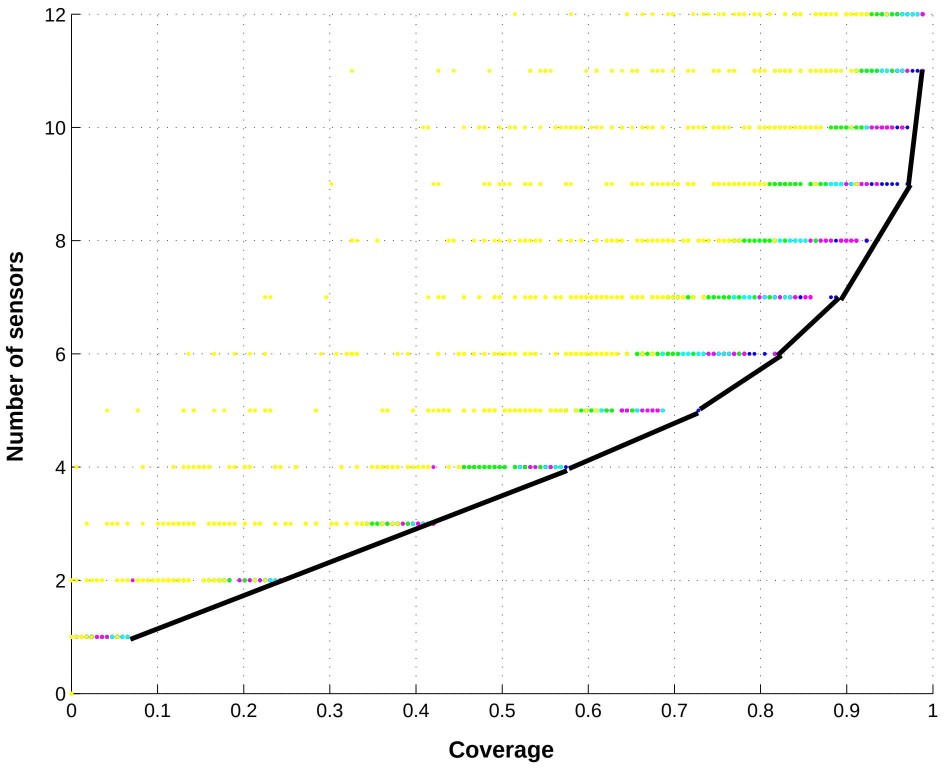

Figure A-3: Objectives graph for CS4.

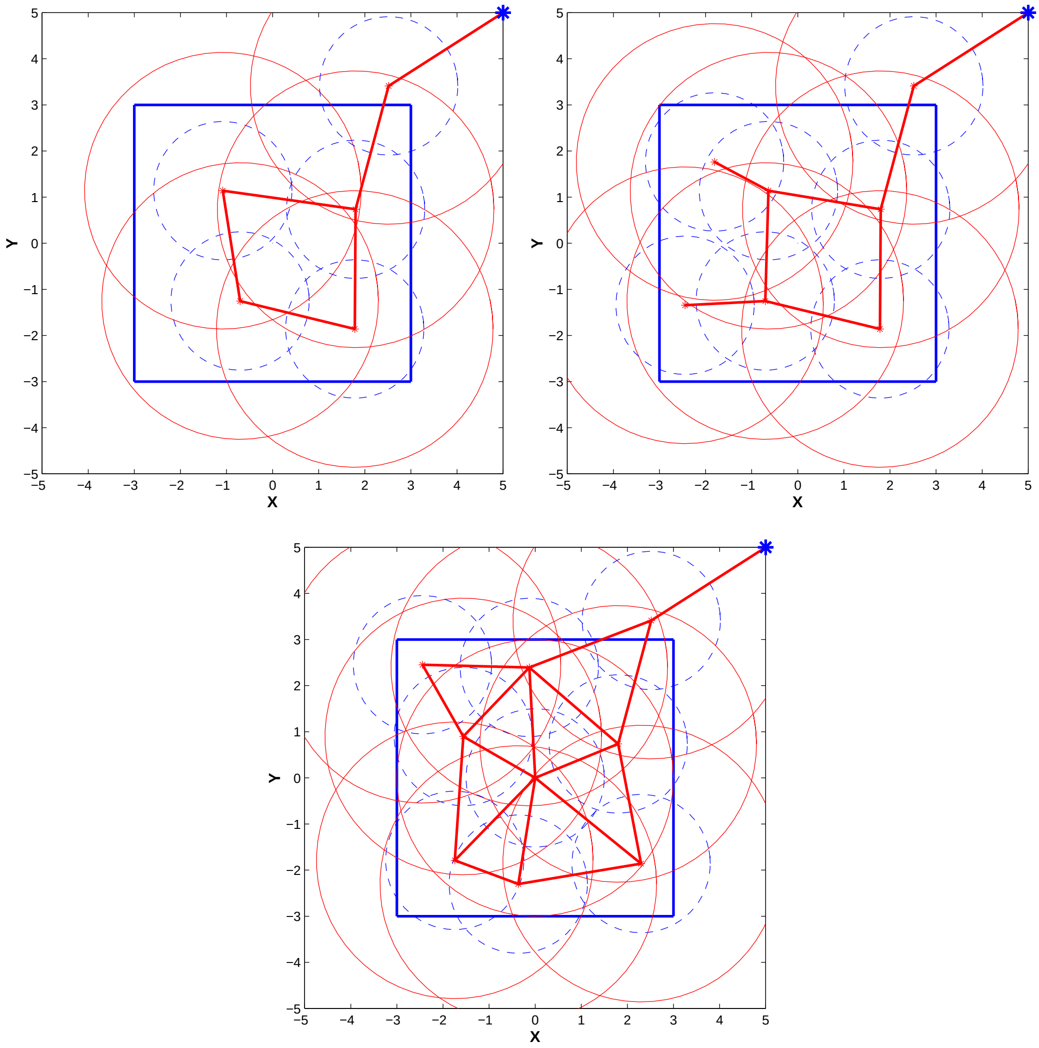

Figure A-4: Non-dominated designs with 5, 7, and 9 sensors.

Appendix B

Additional proofs for Chapter 5

B.1 Biases as additional parameters

As we mentioned at the beginning of Chapter 5, the biases were treated as additional noise terms. Another approach would be to treat them as additional parameters to be estimated, as in [77, 78]. Our goal here is to show that if we follow this approach with the bias distribution of (4.9), the information from NLOS beacons is not taken into account in the PEB.

Let us assume that the *m* first beacons are NLOS, while the others are LOS. We assume the variance of the measurements does not depend on the distance.

The vector to be estimated is $\mathbf{l} = (x_{\mathrm{A}}, y_{\mathrm{A}}, b_1, ..., b_m)$, where $\mathbf{p}_{\mathrm{A}} = (x_{\mathrm{A}}, y_{\mathrm{A}})$ is the position and $\mathbf{b} = (b_1, ..., b_m)$ is the vector of biases from the *m* NLOS beacons. We divide $\mathbf{l}$ into nonrandom and random components: the position $\mathbf{p}_{\mathrm{A}}$ and the biases $\mathbf{b}$, respectively. The *a priori* joint probability density function of $\mathbf{b}$, $f_{\mathbf{b}}(\mathbf{b})$, is given by (4.3). For simplicity we assume that $K^{(i)} = 1$ for all i, and $\beta^{(i)}$ is the maximum bias (just as in Section 5.2.3). Since we will have to take derivatives, we add some artificial zero-mean Gaussian noise of variance σ_b^2. As σ_b goes to zero, we converge to the exact pdf.

The joint pdf of $\mathbf{b}$ therefore is:

$$f_{\mathbf{b}}(\mathbf{b}) = \prod_{i=1}^{m} \frac{1}{\beta^{(i)}} \left[Q\left(\frac{b_i + \beta^{(i)}/2}{\sigma_b}\right) - Q\left(\frac{b_i - \beta^{(i)}/2}{\sigma_b}\right)\right]. \tag{B.1}$$

The joint probability density function of the range measurements, conditioned on **l**, is now a zero-mean Gaussian with variance σ^2 (where σ^2 is the variance of the range measurements, which we assumed constant):

$$f(\mathbf{r}|\,\mathbf{l}) \propto \prod_{i=1}^{m} e^{-(r_i-d_i-b_i)^2/2\sigma^2} \prod_{i=m+1}^{n_{\mathrm{B}}} e^{-(r_i-d_i)^2/2\sigma^2}, \tag{B.2}$$

which is the equivalent of (4.9). The Fisher Information Matrix is replaced by its Bayesian version [93]:

$$\mathbf{J} = \mathbf{J}_{\mathrm{D}} + \mathbf{J}_{\mathrm{P}}, \tag{B.3}$$

where:

$$\begin{aligned} \mathbf{J}_{\mathrm{D}} &= \mathbb{E}_{\mathbf{r},\,\mathbf{b}}\left\{\left[\nabla_{\mathbf{l}} \ln\left(f(\mathbf{r}|\mathbf{l})\right)\right]\left[\nabla_{\mathbf{l}} \ln\left(f(\mathbf{r}|\mathbf{l})\right)\right]^{\mathrm{T}}\right\}, && \text{(B.4)} \\ \mathbf{J}_{\mathrm{P}} &= \mathbb{E}_{\mathbf{b}}\left\{\left[\nabla_{\mathbf{l}} \ln\left(f_{\mathbf{b}}(\mathbf{b})\right)\right]\left[\nabla_{\mathbf{l}} \ln\left(f_{\mathbf{b}}(\mathbf{b})\right)\right]^{\mathrm{T}}\right\}. && \text{(B.5)} \end{aligned}$$

We have:

$$\begin{aligned} \nabla_{\mathbf{l}} \ln\left(f(\mathbf{r}|\mathbf{l})\right) &= \frac{1}{2\sigma^2} \begin{bmatrix} \sum_{i=1}^{m} \cos\theta_i \left(\frac{r_i-d_i-b_i}{\sigma^2}\right) + \sum_{i=m+1}^{n_{\mathrm{B}}} \cos\theta_i \left(\frac{r_i-d_i}{\sigma^2}\right) \\ \sum_{i=1}^{m} \sin\theta_i \left(\frac{r_i-d_i-b_i}{\sigma^2}\right) + \sum_{i=m+1}^{n_{\mathrm{B}}} \sin\theta_i \left(\frac{r_i-d_i}{\sigma^2}\right) \\ \frac{r_1-d_1-b_1}{\sigma^2} \\ \vdots \\ \frac{r_m-d_m-b_m}{\sigma^2} \end{bmatrix} && \text{(B.6)} \\ &= \frac{1}{2\sigma^2} \begin{bmatrix} H_{\mathrm{NLOS}}^{\mathrm{T}} & H_{\mathrm{LOS}}^{\mathrm{T}} \\ I_m & 0 \end{bmatrix} \cdot \mathbf{V}, && \text{(B.7)} \end{aligned}$$

where I_m is the identity matrix of size m,

$$\mathbf{V} = \begin{bmatrix} r_1 - d_1 - b_1 \\ \vdots \\ r_m - d_m - b_m \\ r_{m+1} - d_{m+1} \\ \vdots \\ r_{n_\mathrm{B}} - d_{n_\mathrm{B}} \end{bmatrix}, \tag{B.8}$$

and

$$H = \begin{bmatrix} H_{NLOS} \\ H_{LOS} \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ \vdots & \vdots \\ \cos\theta_{n_\mathrm{B}} & \sin\theta_{n_\mathrm{B}} \end{bmatrix}. \tag{B.9}$$

We can now calculate the first component of the FIM:

$$\begin{aligned}
\mathbf{J}_\mathrm{D} &= \frac{1}{4\sigma^4}\mathbb{E}_{\mathbf{r},\,\mathbf{b}}\left\{\begin{bmatrix} H_\mathrm{NLOS}^\mathrm{T} & H_\mathrm{LOS}^\mathrm{T} \\ I_m & 0 \end{bmatrix} \cdot \mathbf{V} \cdot \mathbf{V}^\mathrm{T} \cdot \begin{bmatrix} H_\mathrm{NLOS} & I_m \\ H_\mathrm{LOS} & 0 \end{bmatrix}\right\} && \text{(B.10)} \\
&= \frac{1}{4\sigma^4}\begin{bmatrix} H_\mathrm{NLOS}^\mathrm{T} & H_\mathrm{LOS}^\mathrm{T} \\ I_m & 0 \end{bmatrix} \mathbb{E}_{\mathbf{r},\,\mathbf{b}}\left\{\mathbf{V} \cdot \mathbf{V}^\mathrm{T}\right\} \begin{bmatrix} H_\mathrm{NLOS} & I_m \\ H_\mathrm{LOS} & 0 \end{bmatrix} && \text{(B.11)} \\
&= \frac{1}{4\sigma^2}\begin{bmatrix} H_\mathrm{NLOS}^\mathrm{T} & H_\mathrm{LOS}^\mathrm{T} \\ I_m & 0 \end{bmatrix} \cdot \begin{bmatrix} H_\mathrm{NLOS} & I_m \\ H_\mathrm{LOS} & 0 \end{bmatrix} && \text{(B.12)} \\
&= \begin{bmatrix} H^\mathrm{T} H & H_\mathrm{NLOS}^\mathrm{T} \\ H_\mathrm{NLOS} & I_m \end{bmatrix}, && \text{(B.13)}
\end{aligned}$$

where we have used the fact that $\mathbb{E}_{\mathbf{r},\,\mathbf{b}}\left\{\mathbf{V} \cdot \mathbf{V}^\mathrm{T}\right\} = \sigma^2 I_{n_\mathrm{B}}$.

The derivation of $\mathbf{J}_\mathrm{P}$ is similar to that of the importance coefficients in Section 5.2. We need to obtain the gradient of $p_\mathbf{b}(\mathbf{b})$ with respect to **l**. The derivative with respect to $\mathbf{p}_\mathrm{A}$ are 0, and those with respect to **b** turn out to lead to the importance coefficients $A(\beta^{(i)}, \sigma_b)$ of

(**??**), where we make the dependence on σ_b explicit. We obtain:

$$\mathbf{J}_{\mathrm{P}} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & A(\beta^{(1)}, \sigma_b) & & \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & & & A(\beta^{(m)}, \sigma_b) \end{bmatrix}. \tag{B.14}$$

The Bayesian FIM is therefore:

$$\mathbf{J} = \frac{1}{4\sigma^2} \begin{bmatrix} H^{\mathrm{T}}H & & H_{\mathrm{NLOS}}^{\mathrm{T}} & \\ & 1 + 4\sigma^2 A(\beta^{(1)}, \sigma_b) & & \\ H_{\mathrm{NLOS}} & & \ddots & \\ & & & 1 + 4\sigma^2 A(\beta^{(m)}, \sigma_b) \end{bmatrix}. \tag{B.15}$$

To obtain the PEB we need to invert this matrix. Given a partitioned matrix M

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \tag{B.16}$$

its inverse is given by [9]:

$$M^{-1} = \begin{bmatrix} Q & -QBD^{-1} \\ -D^{-1}CQ & D^{-1} + D^{-1}CQBD^{-1} \end{bmatrix}, \tag{B.17}$$

with $Q = (A - BD^{-1}C)^{-1}$. We let $A = H^{\mathrm{T}}H$, $B = H_{\mathrm{NLOS}}^{\mathrm{T}}$, $C = H_{\mathrm{NLOS}}$ and D is a $m \times m$ diagonal matrix with diagonal elements equal to $1 + 4\sigma^2 A(\beta^{(i)}, \sigma_b)$ for $i = 1 \dots m$. We are only interested in the 2x2 upper left elements, so all we need is $4\sigma^2 Q = 4\sigma^2(H^{\mathrm{T}}H - H_{\mathrm{NLOS}}^{\mathrm{T}} D^{-1} H_{\mathrm{NLOS}})^{-1}$. We have:

$$Q = (H_{\mathrm{LOS}}^{\mathrm{T}} H_{\mathrm{LOS}} + H_{\mathrm{NLOS}}^{\mathrm{T}} H_{\mathrm{NLOS}} - H_{\mathrm{NLOS}}^{\mathrm{T}} D^{-1} H_{\mathrm{NLOS}})^{-1} \tag{B.18}$$

$$= (H_{\mathrm{LOS}}^{\mathrm{T}} H_{\mathrm{LOS}} + H_{\mathrm{NLOS}}(I - D^{-1}) H_{\mathrm{NLOS}})^{-1} \tag{B.19}$$

$$= \begin{bmatrix} \sum_{i=1}^{n_{\mathrm{B}}} C_i \cos^2\theta_i & \sum_{i=1}^{n_{\mathrm{B}}} C_i \cos\theta_i \sin\theta_i \\ \sum_{i=1}^{n_{\mathrm{B}}} C_i \cos\theta_i \sin\theta_i & \sum_{i=1}^{n_{\mathrm{B}}} C_i \sin^2\theta_i \end{bmatrix}^{-1}, \tag{B.20}$$

where $C_i = 1$ for the LOS beacons, and $C_i = 4\sigma^2 A(\beta^{(i)}, \sigma_b)/(1 + 4\sigma^2 A(\beta^{(i)}, \sigma_b))$ for the NLOS beacons. The PEB on the position estimate is the trace of this matrix. We have:

$$\mathsf{PEB} = \sqrt{\mathbb{T}\left\{4\sigma^2 Q\right\}} \tag{B.21}$$

$$= \sigma \sqrt{\frac{4\sum_{i=1}^{n_{\mathrm{B}}} C_i}{(\sum_{i=1}^{n_{\mathrm{B}}} C_i \cos^2\theta_i)(\sum_{i=1}^{n_{\mathrm{B}}} C_i \sin^2\theta_i) - (\sum_{i=1}^{n_{\mathrm{B}}} C_i \cos\theta_i \sin\theta_i)^2}}. \tag{B.22}$$

We can now make σ_b go to 0 in order to model the bias statistics as a uniform distribution. From (**??**) we see that if σ goes to 0, then the importance weight goes to infinity. This in turns means that $C_i = 1$ for $i = 1 \ldots m$. Therefore the PEB is equal to:

$$\mathsf{PEB} = \sigma \sqrt{\frac{4(n_{\mathrm{B}} - m)}{(\sum_{i=m+1}^{n_{\mathrm{B}}} \cos^2\theta_i)(\sum_{i=m+1}^{n_{\mathrm{B}}} \sin^2\theta_i) - (\sum_{i=m+1}^{n_{\mathrm{B}}} \cos\theta_i \sin\theta_i)^2}} \tag{B.23}$$

We recognize here the PEB when the information of the NLOS beacons is disregarded (it is actually equal to the standard deviation times the GDOP given by the LOS beacons). We conclude that if we model the bias statistics as in Chapter 4 and consider the biases as additional parameters to be estimated, the PEB does not depend on the NLOS beacons. However our analysis in Chapter 5 shows that the PEB can in fact be lowered by the NLOS information if the biases are treated as noise.

B.2 Simplification of matrix J

Referring to (5.7), let us examine the generic element in the sum of the upper left element of $\mathbf{J}$. For $i \neq j$ we have:

$$\mathbb{E}_{\mathbf{r}} \left\{ \frac{1}{f_i(r_i|\mathbf{p}_{\mathrm{A}})} \frac{1}{f_j(r_j|\mathbf{p}_{\mathrm{A}})} \frac{\partial f_i(r_i|\mathbf{p}_{\mathrm{A}})}{\partial x_{\mathrm{A}}} \frac{\partial f_j(r_j|\mathbf{p}_{\mathrm{A}})}{\partial x_{\mathrm{A}}} \right\} = \mathbb{E}_{\mathbf{r}} \left\{ \frac{1}{f_i(r_i|\mathbf{p}_{\mathrm{A}})} \frac{\partial f_i(r_i|\mathbf{p}_{\mathrm{A}})}{\partial x_{\mathrm{A}}} \right\} \cdot \\ \mathbb{E}_{\mathbf{r}} \left\{ \frac{1}{f_j(r_j|\mathbf{p}_{\mathrm{A}})} \frac{\partial f_j(r_j|\mathbf{p}_{\mathrm{A}})}{\partial x_{\mathrm{A}}} \right\} \tag{B.24}$$

since the measurements from beacons i and j are independent. We can write:

$$\mathbb{E}_{\mathbf{r}}\left\{\frac{1}{f_i(r_i|\mathbf{p}_\mathrm{A})}\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}}\right\} = \int_{-\infty}^{+\infty}\frac{1}{f_i(r_i|\mathbf{p}_\mathrm{A})}\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}}f_i(r_i|\mathbf{p}_\mathrm{A})dr_i \tag{B.25}$$

$$= \int_{-\infty}^{+\infty}\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}}dr_i. \tag{B.26}$$

We know that $f_i(r_i|\mathbf{p}_\mathrm{A})$ is a continuous function of (r_i, x_A). It is also easy to show that its derivative $\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}}$ (see equation (5.9)) is also continuous, and that its absolute value is integrable in r_i for all x_A. We can therefore exchange the integral with the derivative so that:

$$\int_{-\infty}^{+\infty}\frac{\partial f_i(r_i|\mathbf{p}_\mathrm{A})}{\partial x_\mathrm{A}}dr_i = \frac{\partial}{\partial x_\mathrm{A}}\int_{-\infty}^{+\infty}f_i(r_i|\mathbf{p}_\mathrm{A})dr_i = 0 \tag{B.27}$$

since $\int_{-\infty}^{+\infty} f_i(r_i|\mathbf{p}_\mathrm{A})dr_i = 1$. The same holds for the derivative with respect to y_A.

Appendix C

Maximum Likelihood Estimation in Chapter 5

In this appendix we describe the non-linear least-squares (NLLS) method used in Chapter 5 to estimate the location of the agent given a set of range measurements.

The range measurements are described by (4.1) as $\mathbf{r} = \mathbf{d}(\mathbf{p}_{\mathrm{A}}) + \boldsymbol{\epsilon}$, where $\mathbf{r}$ is the vector containing the n_{B} range measurements, $\boldsymbol{\epsilon}$ is a vector of n_{B} independent, zero-mean Gaussian random variables, and $\mathbf{d}(\mathbf{p}_{\mathrm{A}})$ is the vector containing the n_{B} distance components. We assume that we have some prior estimate $\mathbf{p}_{\mathrm{A}}^{\mathbf{0}}$ of $\mathbf{p}_{\mathrm{A}}$, with covariance P_0, and we call R the covariance of the measurement error ϵ. Our goal is to find the value of $\mathbf{p}_{\mathrm{A}}$ that will minimize the following non-linear weighted least-squares L

$$L = (\mathbf{p}_{\mathrm{A}} - \mathbf{p}_{\mathrm{A}}^{\mathbf{0}})^T P_0^{-1} (\mathbf{p}_{\mathrm{A}} - \mathbf{p}_{\mathrm{A}}^{\mathbf{0}}) + (\mathbf{r} - \mathbf{d}(\mathbf{p}_{\mathrm{A}}))^T R^{-1} (\mathbf{r} - \mathbf{d}(\mathbf{p}_{\mathrm{A}})). \tag{C.1}$$

Let $D_{\mathbf{p}_{\mathrm{A}}^{\mathbf{0}}}$ be the Jacobian of $\mathbf{d}(\mathbf{p}_{\mathrm{A}})$ at $\mathbf{p}_{\mathrm{A}}^{\mathbf{0}}$, that is,

$$D_{\mathbf{p}_{\mathrm{A}}^{\mathbf{0}}} = \nabla_{\mathbf{p}_{\mathrm{A}}} \mathbf{d}|_{\mathbf{p}_{\mathrm{A}}^{\mathbf{0}}}. \tag{C.2}$$

By taking the gradient of J we obtain

$$\nabla_{\mathbf{p}_{\mathrm{A}}} J = (\mathbf{p}_{\mathrm{A}} - \mathbf{p}_{\mathrm{A}}^{\mathbf{0}})^T P_0^{-1} - (\mathbf{r} - \mathbf{d}(\mathbf{p}_{\mathrm{A}}))^T R^{-1} D_{\mathbf{p}_{\mathrm{A}}}. \tag{C.3}$$

We define $G^{\mathrm{T}}(\mathbf{p}_{\mathrm{A}})$ as the right-hand side of this equation, and we desire find the value of $\mathbf{p}_{\mathrm{A}}$ which verifies $G(\mathbf{p}_{\mathrm{A}}) = 0$. Since this equation is non-linear, we use the Newton-Raphson method to find a solution. The procedure is as follows for each agent location [47]:

1. Guess $\mathbf{p}_{\mathrm{A}}$;
2. Calculate $\mathbf{d}(\mathbf{p}_{\mathrm{A}})$ and $D_{\mathbf{p}_{\mathrm{A}}}$;
3. Find $P_1^{-1} = P_0^{-1} + D_{\mathbf{p}_{\mathrm{A}}}^T R^{-1} D_{\mathbf{p}_{\mathrm{A}}}$ (which is equal to $\nabla_{\mathbf{p}_{\mathrm{A}}} G(\mathbf{p}_{\mathrm{A}})$);
4. Compute $G(\mathbf{p}_{\mathrm{A}})$;
5. Compute $\mathbf{dp}_{\mathrm{A}} = -P_1 G(\mathbf{p}_{\mathrm{A}})$;
6. Stop if $\|\mathbf{dp}_{\mathrm{A}}\| < \delta$;
7. Otherwise $\mathbf{p}_{\mathrm{A}} := \mathbf{p}_{\mathrm{A}} + \mathbf{dp}_{\mathrm{A}}$, go to 2).

We use this NLLS method on a set of n_{A} given agent locations. The initial guess $\mathbf{p}_{\mathrm{A_j}}^0$ for agent point $\mathbf{p}_{\mathrm{A_j}}$ ($j \in \{1, ..., n_{\mathrm{A}}\}$) is sampled from a Gaussian distribution centered at $\mathbf{p}_{\mathrm{A_j}}$, with a variance of $0.1m^2$. The actual value of the variance does not influence the final result. We therefore initialize the covariance at:

$$P_0 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}. \tag{C.4}$$

The measurement covariance is taken to be:

$$R = \begin{bmatrix} (10^{-3})^2 & 0 \\ 0 & (10^{-3})^2 \end{bmatrix}, \tag{C.5}$$

that is the value of the variance at $1m$. The NLLS algorithm is then run until convergence to the least-squares estimate $\hat{\mathbf{p}}_{\mathrm{A_j}}$ given the range measurements. We thus obtain a NLLS estimate for each agent point.

The comparison with the PEB must be done with care. Recalling the PEB expression (5.4), the mean square error (MSE) for agent location j is defined by as

$$\mathsf{MSE}_j = \sqrt{\frac{1}{n_\mathrm{S}} \sum_{k=1}^{n_\mathrm{S}} \left\{ \big(x_\mathrm{A} - \hat{x}_{\mathrm{A_j}}(k)\big)^2 + \big(y_\mathrm{A} - \hat{y}_{\mathrm{A_j}}(k)\big)^2 \right\}}, \tag{C.6}$$

where n_S is the number of times each agent location is estimated, and $\big(\hat{x}_{\mathrm{A_j}}(k), \hat{y}_{\mathrm{A_j}}(k)\big)$ is the k^th estimate of agent location j. Having a value of the MSE for each agent location, we can then obtain MSE_{90} as the value of the MSE for 90% of the agent locations (to mirror the definition of PEB_{90}).

Appendix D

Additional proofs for Chapter 6

D.1 Additional Results on the Coordinate Transform

Following the definitions of Section 6.3.1, we derive more results that will be helpful for visualizing the complex vectors and using them in the subsequent proofs.

Lemma D.1 (Fundamental relationships)**.** *The following relationships hold:*

$$\mathbf{z_i}(\boldsymbol{\theta}) + A_i = e^{-2j\theta_i}\mathbf{r}(\boldsymbol{\theta}), \quad \forall\, i = 1, \ldots, n_B \tag{D.1}$$

$$r(\boldsymbol{\theta}) = |\mathbf{z_i}(\boldsymbol{\theta}) + A_i|, \quad \forall\, i = 1, \ldots, n_B \tag{D.2}$$

Proof. These results follow directly from the definitions. □

Corollary D.2. *For $k = 1, \ldots, n_B$, $\mathbf{z_k}(\boldsymbol{\theta})$ is the vector starting at $(0, 0)$ and ending on the circle of radius $r(\boldsymbol{\theta})$, centered at $(-A_k, 0)$.*

Proof. This follows from (D.1), which states that $\mathbf{z_k}(\boldsymbol{\theta})$ is obtained by rotating vector $\mathbf{r}(\boldsymbol{\theta})$ by $-2\theta_k$, and adding vector $(-A_k, 0)$ to it (the latter is equivalent to a translation along the negative real axis). □

Corollary D.3. *If θ_i is varied while all other θ_k remain fixed, $\mathbf{z_i}(\boldsymbol{\theta})$ rotates about $(0, 0)$ by an angle $-2\theta_i$, but* its magnitude does not change. *However all the other $\mathbf{z_k}(\boldsymbol{\theta})$ rotate* and *change magnitude as θ_i varies.*

Proof. Equation (6.2) states that $\mathbf{z_i}(\boldsymbol{\theta})$ depends on θ_i only through $e^{-2j\theta_i}$, which appears as a factored term, so that the norm of θ_i does not depend on θ_i. For all the other vectors however, the dependence on θ_i is through $A_i e^{-2j\theta_i}$, which is inside the sum and cannot be factored out. □

Corollary D.4. *Let* $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_i, \ldots, \theta_{n_B})$. *If* θ_i *is changed to* $\tilde{\theta}_i$ *such that* $\mathbf{z_i}(\tilde{\boldsymbol{\theta}})$ *lies on the real axis (with* $\tilde{\boldsymbol{\theta}} = (\theta_1, \ldots, \tilde{\theta}_i, \ldots, \theta_{n_B})$*), then the resulting* $r(\tilde{\boldsymbol{\theta}})$ *is equal to* $|A_i - |\mathbf{z_i}(\boldsymbol{\theta})||$*, i.e. it does not explicitly depend on the actual value of the new* $\tilde{\theta}_i$*.*

Proof. According to Corollary D.3, when θ_i varies to $\tilde{\theta}_i$ the magnitude of $\mathbf{z_i}(\tilde{\boldsymbol{\theta}})$ does not change, so that $|\mathbf{z_i}(\tilde{\boldsymbol{\theta}})| = |\mathbf{z_i}(\boldsymbol{\theta})|$. Also, according to Corollary D.2 the tip of $\mathbf{z_i}(\tilde{\boldsymbol{\theta}})$ is on the circle of radius $r(\tilde{\boldsymbol{\theta}})$ centered at $(-A_i, 0)$. But since both $(-A_i, 0)$ and $\mathbf{z_i}(\tilde{\boldsymbol{\theta}})$ are on the real axis, we have $|\mathbf{z_i}(\tilde{\boldsymbol{\theta}}) + A_i| = |A_i - |\mathbf{z_i}(\tilde{\boldsymbol{\theta}})||$, which by (D.2) is equal to $r(\tilde{\boldsymbol{\theta}})$. □

These corollaries help to to visualize the vectors $\mathbf{z_k}(\boldsymbol{\theta})$ and $\mathbf{r}(\boldsymbol{\theta})$ in the plane. This is illustrated in Figure D-1 for 3 beacons. The beacons are placed at θ_1, θ_2 and θ_3, and the A_k are assumed constant with $A_3 = 3$, $A_2 = 2$, and $A_1 = 1.3$. As indicated in Corollary D.2, the 3 corresponding vectors $\mathbf{z_1}(\boldsymbol{\theta})$, $\mathbf{z_2}(\boldsymbol{\theta})$, and $\mathbf{z_3}(\boldsymbol{\theta})$ all lie on circles of equal radius $r(\boldsymbol{\theta})$, respectively centered at $(-A_1, 0)$, $(-A_2, 0)$, and $(-A_3, 0)$.

If θ_3 varies while θ_1 and θ_2 remain fixed, $\mathbf{z_3}(\boldsymbol{\theta})$ will rotate about $(0, 0)$ with its magnitude unchanged (Corollary D.3). Note that since the tip of this vector must always lie on the circle of radius $r(\boldsymbol{\theta})$ centered at $(-A_3, 0)$, this radius will change as $\mathbf{z_3}(\boldsymbol{\theta})$ rotates. Note in particular that $r(\boldsymbol{\theta})$ *will be minimum when* $\mathbf{z_3}(\boldsymbol{\theta})$ *lies on the negative real axis.* In that case the radius is then equal to $|A_3 - |\mathbf{z_3}(\boldsymbol{\theta})||$ (Corollary D.4). This is an important observation that will be key to proving the correctness of RELOCATE.

D.2 Corollary of Lemma 6.3

Note that Corollary 6.4 is in accordance with our observation about Fig. D-1: when rotating $\mathbf{z_i}$, the error radius is minimized when $\mathbf{z_i}$ lies on the negative real axis. This is just what θ_i^* does.

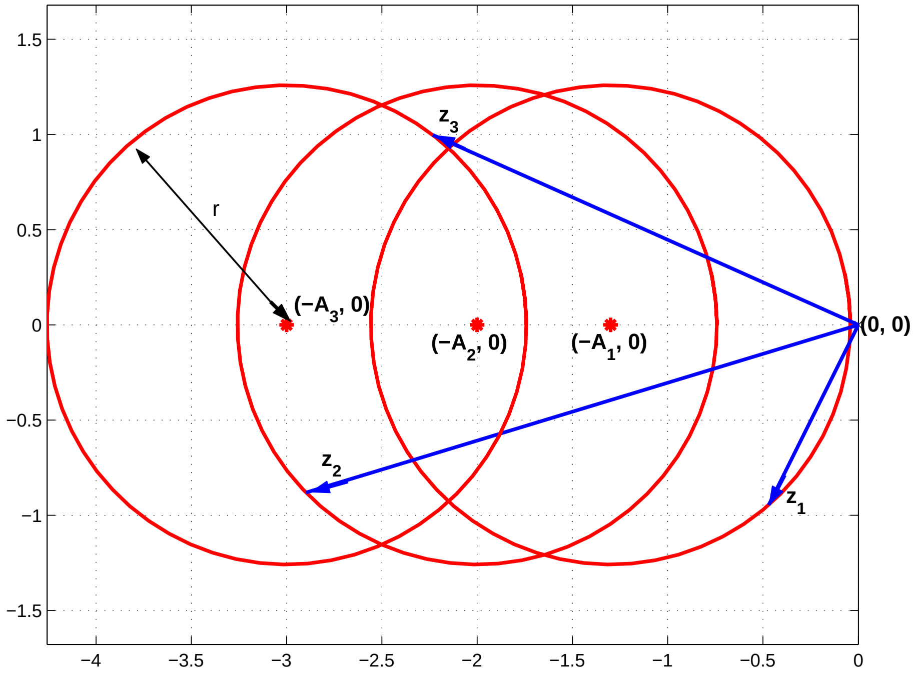

Figure D-1: Plot of the $\mathbf{z_k}(\theta)$ for 3 beacons, with $A_1 = 1.3$, $A_2 = 2$ and $A_3 = 3$. Note that the 3 vectors lie on the circles of equal radius (equal to $r(\theta) = |A_k + \mathbf{z_k}(\theta)|$) centered on $(-A_k, 0)$, for $k = 1 \ldots 3$.

Corollary D.5. *If beacon i_p is selected for relocation at iteration p, the error radius $r(\boldsymbol{\theta}^{p+1})$ after relocation will be equal to $|A_{i_p} - |\mathbf{z_{i_p}}(\boldsymbol{\theta}^p)||$*

Proof. Since relocating beacon i_p makes $\mathbf{z_{i_p}}$ lie on the negative real axis (Corollary 6.4), the result is a direct consequence of Corollary D.4. □

Given the n_B $\mathbf{z_k}(\boldsymbol{\theta}^p)$ at iteration p, we can therefore easily calculate the error radius resulting in relocating any of the beacons, so that the decrease in error radius can be predicted beforehand.

D.3 Proof of Lemma 6.7: Optimal Convergence of RELOCATE

Proof. For a given value of $\boldsymbol{\theta}^p$, we write $r^p \triangleq r(\boldsymbol{\theta}^p)$, $\mathbf{r}^p \triangleq \mathbf{r}(\boldsymbol{\theta}^p)$, and $\mathbf{z_k}^p \triangleq \mathbf{z_k}(\boldsymbol{\theta}^p)$ for all k and p.

Suppose that at iteration p RELOCATE produces a configuration $\boldsymbol{\theta}^p$ such that all $\mathbf{z_k}^p$ are on the negative real axis, while $r^p > r^*$. We first note that if nothing is done, the algorithm "stalls" at this point: because RELOCATE is a coordinate descent algorithm, it stops yielding improved results once a stationary point is reached (Lemma 6.6).

But we will show that by entering the INCREASE_RADIUS routine in step (5), the algorithm escapes this stationary point (notably by finding a configuration that is not stationary and has a radius strictly smaller). We also will show that step (5) can only be satisfied a finite number of times, so that is there are only a finite number of suboptimal stationary points. This implies that RELOCATE will not stop until it reaches an *optimal* stationary point. We show that this optimal stationary point is that with error radius equal to r^*.

Let us then assume that all the $\mathbf{z_k}^p$ lie on the negative real axis. By Corollary D.4 we then have $r^p = |A_k - |\mathbf{z_k}^p||$. Since in addition the $\mathbf{z_k}^p$ are negative real numbers, we either have $r^p = A_k + \mathbf{z_k}^p$ or $r^p = -A_k - \mathbf{z_k}^p$. But by (D.1) we also know that $e^{-2j\theta_k^p}\mathbf{r}^p = \mathbf{z_k}^p + A_k$. Combining these we conclude that we must have

$$\frac{\mathbf{r}^p}{r^p} = \begin{cases} e^{2j\theta_k^p} & \text{or} \\ -e^{2j\theta_k^p}, & \end{cases} \tag{D.3}$$

so that the angles θ_k^p are equal modulo $\pi/2$. There therefore exists a partition $(S, \overline{S})$ of $\{1, \dots, n_\text{B}\}$ such that (6.3) can be written

$$r^p = \sum_{k \in S} A_k - \sum_{k \in \overline{S}} A_k. \tag{D.4}$$

Without loss of generality we assume that $\theta_k^p = 0$ for $k \in S$, and $\theta_k^p = \pi/2$ for $k \in \overline{S}$. We therefore have $\mathbf{r}^p = r^p$. Step (1) of INCREASE_RADIUS is therefore feasible.

The set S cannot be empty, otherwise r^p would be negative. Suppose that S is a singleton, so that $r^p = A_S - \sum_{k \in \overline{S}} A_k$. This is only possible (i.e. it is only positive) if $S = \{n_\text{B}\}$ and $A_{n_\text{B}} > \sum_{k=1}^{n_\text{B}-1} A_k$. But in that case $r^* = A_{n_\text{B}} - \sum_{k=1}^{n_\text{B}-1} A_k$ and $r^p = r^*$, which contradicts our assumption. S must then have at least 2 elements, so step (2) of INCREASE_RADIUS is feasible. Let l and m be 2 distinct indices of S.

Our strategy is as follows. The error radius cannot be reduced by relocating one sensor, which is what the normal operation of RELOCATE does. Instead we will relocate sensor l to a location different than that given by (6.7). This will result in an error radius r^{p+1} greater than r^p. But then we will show that relocating sensor m according to (6.7) will yield an error radius r^{p+2} such that $r^{p+2} < r^p$.

Assume then that at iteration p we relocate sensor l to θ_l. As a shorthand notation we write $\mathbf{z_m}^{p+1}(\theta_l) \triangleq \mathbf{z_m}(\theta_1^p, \ldots, \theta_l, \ldots, \theta_{n_\mathrm{B}}^p)$. At iteration $p+1$ sensor m will in turn be relocated, this time to a θ_m^{p+2} given by (6.7). According to Corollary D.4 the resulting error radius at iteration $p+2$ is then equal to $|A_m - |\mathbf{z_m}^{p+1}(\theta_l)||$ (i.e. the actual value of θ_m^{p+2} does not matter, what matters is the fact that by (6.7) $\mathbf{z_m}(\boldsymbol{\theta}^{p+2})$ will lie on the negative real axis).

Let us then study how $r^{p+2}(\theta_l) = |A_m - |\mathbf{z_m}^{p+1}(\theta_l)||$ varies as a function of θ_l (where we have made the dependence of r^{p+2} on θ_l explicit). In particular we want to show that there exists a θ_l such that $r^{p+2}(\theta_l) < r^p$. This is true if and only if

$$A_m \in \left(\min_{\theta_l \in [0,\pi)} |\mathbf{z_m}^{p+1}(\theta_l)| - r^p, \quad \max_{\theta_l \in [0,\pi)} |\mathbf{z_m}^{p+1}(\theta_l)| + r^p \right). \tag{D.5}$$

We write $a_1 = \min_{\theta_l \in [0,\pi)} |\mathbf{z_m}^{p+1}(\theta_l)| - r^p$ and $a_2 = \max_{\theta_l \in [0,\pi)} |\mathbf{z_m}^{p+1}(\theta_l)| + r^p$. Note that $a_2 \geq a_1$ since r^p is non-negative. There therefore exists a θ_l such that $r^{p+2}(\theta_l) < r^p$ if and only if the interval $\mathcal{A} \triangleq (a_1, a_2)$ contains A_m.

Let us first obtain $\mathbf{z_m}^{p+1}(\theta_l)$ as an explicit function of θ_l. We have

$$\begin{aligned}
\mathbf{z_m}^{p+1}(\theta_l) &= e^{-2j\theta_m^p}\mathbf{r}(\boldsymbol{\theta}^{p+1}) - A_m && \text{(D.6)}\\
&= e^{-2j\theta_m^p}\left(\sum_{k\neq l} A_k e^{-2j\theta_k^p} + A_l e^{-2j\theta_l}\right) - A_m && \text{(D.7)}\\
&= e^{-2j\theta_m^p}\left(\mathbf{r}^p - A_l e^{-2j\theta_l^p} + A_l e^{-2j\theta_l}\right) - A_m && \text{(D.8)}\\
&= r^p - A_m + A_l\left(e^{-2j\theta_l} - 1\right), && \text{(D.9)}
\end{aligned}$$

where the first equation is from (D.1), in the second we use (6.3) to expand the error vector, in the third we write $\mathbf{r}^p = \sum_{k=1}^n A_k e^{-2j\theta_k^p}$, and in the fourth we use the fact that $\theta_l^p = \theta_m^p = 0$ since l and m belong to S. The extrema of $|\mathbf{z_m}^{p+1}(\theta_l)|$ are thus found at $\theta_l = 0$ and $\theta_l = \pi/2$, and take the following values

- $|\mathbf{z_m}^{p+1}(0)| = |r^p - A_m|$;
- $|\mathbf{z_m}^{p+1}(\pi/2)| = |r^p - A_m - 2A_l|$.

We can now study the interval $\mathcal{A}$ for the different cases, remembering that $A_l > 0$, $A_m > 0$, and $r^p > 0$.

- $\underline{r^p - A_m - A_l \geq 0}$

  In this case we have $|\mathbf{z_m}^{p+1}(0)| = r^p - A_m$ and $|\mathbf{z_m}^{p+1}(\pi/2)| = |r^p - A_m - 2A_l|$. Since

  $$\begin{aligned} |\mathbf{z_m}^{p+1}(0)| - |\mathbf{z_m}^{p+1}(\pi/2)| &= r^p - A_m - |r^p - A_m - 2A_l| && \text{(D.10)} \\ &= 2A_l, \ \text{ or } \ 2(r^p - A_m - A_l), && \text{(D.11)} \end{aligned}$$

  where both are non-negative, we conclude that the maximum of $|\mathbf{z_m}^{p+1}(\theta_l)|$ is achieved for $\theta_l = 0$, and the minimum for $\theta_l = \pi/2$.

  - If $r^p - A_m - 2A_l \geq 0$, then $|\mathbf{z_m}^{p+1}(\pi/2)| = r^p - A_m - 2A_l$ and:

    $$\begin{cases} a_1 = (r^p - A_m - 2A_l) - r^p < 0 < A_m \\ a_2 = r^p - A_m + r^p = 2(r^p - A_m) + A_m > A_m \end{cases} \tag{D.12}$$

    where the second inequality holds because $r^p - A_m \geq A_l > 0$.

  - If $r^p - A_m - 2A_l < 0$, then $|\mathbf{z_m}^{p+1}(\pi/2)| = -(r^p - A_m - 2A_l)$ and:

    $$\begin{cases} a_1 = -(r^p - A_m - 2A_l) - r^p = -2(r^p - A_m - A_l) - A_m < 0 < A_m \\ a_2 = r^p - A_m + r^p = 2(r^p - A_m) + A_m > A_m \end{cases} \tag{D.13}$$

  In both cases, we have $a_1 < A_m < a_2$.

- $\underline{r^p - A_m - A_l < 0}$

  In this case we have $|\mathbf{z_m}^{p+1}(0)| = |r^p - A_m|$ and $|\mathbf{z_m}^{p+1}(\pi/2)| = -(r^p - A_m - 2A_l)$.

Since

$$|\mathbf{z_m}^{p+1}(\pi/2)| - |\mathbf{z_m}^{p+1}(0)| = -(r^p - A_m - 2A_l) - |r^p - A_m| \quad \text{(D.14)}$$

$$= -2(r^p - A_m - A_l), \text{ or } 2A_l, \quad \text{(D.15)}$$

where both are non-negative, we conclude that the maximum of $|\mathbf{z_m}^{p+1}(\theta_l)|$ is achieved for $\theta_l = \pi/2$, and the minimum for $\theta_l = 0$.

- If $r^p - A_m \geq 0$, then $|\mathbf{z_m}^{p+1}(0)| = r^p - A_m$ and:

  $$\begin{cases} a_1 = (r^p - A_m) - r^p < 0 < A_m \\ a_2 = -(r^p - A_m - 2A_l) + r^p = A_m + 2A_l > A_m \end{cases} \quad \text{(D.16)}$$

- If $r^p - A_m < 0$, then $|\mathbf{z_m}^{p+1}(0)| = -(r^p - A_m)$ and:

  $$\begin{cases} a_1 = -(r^p - A_m) - r^p = -2r^p + A_m < A_m \\ a_2 = -(r^p - A_m - 2A_l) + r^p = A_m + 2A_l > A_m \end{cases} \quad \text{(D.17)}$$

  where the first inequality holds because $r^p > 0$.

In both cases, we have again $a_1 < A_m < a_2$.

We conclude that in any case $A_m \in \mathcal{A}$ so that (D.5) is satisfied. Step (3) of INCREASE_RADIUS is therefore feasible. Let θ_l^{p+1} receive a value such that $|A_m - |\mathbf{z_m}^{p+1}(\theta_l^{p+1})|| < r^p$ (just choose $\theta_l^{p+1} = 0$ or $\pi/2$ depending on the relative values of r^p, A_l and A_m). At iteration $p+1$ relocating sensor m according to (6.7) will then be guaranteed to yield an error radius r^{p+2} strictly smaller than r^p.

Each time the INCREASE_RADIUS routine is entered, a smaller radius can therefore be found. But in order to enter the INCREASE_RADIUS routine in the first place, we have also shown that $\boldsymbol{\theta}$ must be such that (D.4) is verified. Since there are only a finite number of partitions of $(1, \ldots, n_{\mathrm{B}})$, the INCREASE_RADIUS routine can only be entered a finite number of times. We conclude that the error radius is guaranteed to decrease until it converges to its optimal value r^*. □

A consequence of the convergence of RELOCATE to the global minimum is the following corollary, which strengthens Lemma 6.2.

Corollary D.6. *There exists $\boldsymbol{\theta}$ such that $r(\boldsymbol{\theta}) = 0$ if and only if $A_{n_B} \leq \sum_{k=1}^{n_B-1} A_k$.*

D.4 Derivation of the Expected Rate of Convergence of RELOCATE

Proof. For any iteration p, let $|\cos(\alpha_M)| = \min_{k \neq i} |\cos \alpha_k|$, $X_M = |\cos(\alpha_M)|$, and $X_k = |\cos(\alpha_k)|$ for all k. Note that the X_k are independent, identically distributed because of our assumptions. We have

$$\begin{aligned}
\mathbb{P}\left\{X_M \leq x\right\} &= 1 - \mathbb{P}\left\{X_M > x\right\} && \text{(D.18)}\\
&= 1 - \mathbb{P}\left\{(X_1 > x) \cap \ldots \cap (X_{n_\mathrm{B}} > x)\right\} && \text{(D.19)}\\
&= 1 - \mathbb{P}\left\{X_1 > x\right\}^{n'} && \text{(D.20)}\\
&= 1 - \left(1 - \mathbb{P}\left\{X_1 \leq x\right\}\right)^{n'}, && \text{(D.21)}
\end{aligned}$$

where $\mathbb{P}(A)$ denotes the probability of event A, and $n' = n_\mathrm{B} - 1$. The pdf of X_M can be written as

$$f_{X_M}(x) = n'(1 - \mathbb{P}\left\{X_1 \leq x\right\})^{n'-1} f_X(x), \tag{D.22}$$

where $f_X(x)$ is the pdf of X_1. The statistics of X_1 are

$$\begin{aligned}
\mathbb{P}\left\{X_1 \leq x\right\} &= \mathbb{P}\left\{|\cos \alpha| \leq x\right\} && \text{(D.23)}\\
&= \mathbb{P}\left\{\cos \alpha \leq x \mid 0 \leq \alpha < \pi/2\right\} && \text{(D.24)}\\
&= 1 - \frac{2 \cos^{-1} x}{\pi}. && \text{(D.25)}
\end{aligned}$$

The pdf of X_1 is then $f_X(x) = \frac{2}{\pi} \frac{1}{\sqrt{1-x^2}}$, so that

$$f_{X_M}(x) = n' \left(\frac{2}{\pi}\right)^{n'} \frac{\left(\cos^{-1} x\right)^{n'-1}}{\sqrt{1-x^2}}. \tag{D.26}$$

We can now calculate the expected value of X_M

$$\begin{aligned}\mathbb{E}[X_M] &= \int_0^1 x f_{X_M}(x)dx && \text{(D.27)}\\ &= \left(\frac{2}{\pi}\right)^{n'} \int_0^1 n' \frac{(\cos^{-1} x)^{n'-1}}{\sqrt{1-x^2}} dx. && \text{(D.28)}\\ & && \text{(D.29)}\end{aligned}$$

By integrating by parts and replacing n' by $n_{\mathrm{B}} - 1$ we obtain the desired result

$$\mathbb{E}[\tau] = \int_0^1 \left(\frac{2\cos^{-1} x}{\pi}\right)^{n_{\mathrm{B}}-1} dx. \tag{D.30}$$

□

D.5 Proof of Lemma 6.9: Minimization of Step (2) for Piecewise Constant Importance Weights

Proof. Let $c_k = \cos\theta_k$ and $s_k = \sin\theta_k$ for any k. $\mathsf{PEB}(\theta_{i_p})$ can be written as

$$\begin{aligned}\mathsf{PEB}(\theta_{i_p}) &= \sqrt{\frac{\sum_{k\neq i_p} A_k + A_{i_p}(\theta_{i_p})}{(\sum_{k\neq i_p} A_k c_k^2 + A_{i_p} c_{i_p}^2)(\sum_{k\neq i_p} A_k s_k^2 + A_{i_p} s_{i_p}^2) - (\sum_{k\neq i_p} A_k c_k s_k + A_{i_p} c_{i_p} s_{i_p})^2}}\\ &= \sqrt{\frac{\overline{A_{i_p}} + A_{i_p}(\theta_{i_p})}{\overline{A_{i_p}}/\overline{\mathsf{PEB}}_{i_p}^2 + A_{i_p}(\sum_{k\neq i_p} A_k(c_k^2 s_{i_p}^2 + s_k^2 c_{i_p}^2 - 2c_k s_{i_p} s_k c_{i_p}))}}\\ &= \overline{\mathsf{PEB}}_{i_p} \sqrt{\frac{\overline{A_{i_p}} + A_{i_p}(\theta_{i_p})}{A_{i_p}(\theta_{i_p})G(\theta_{i_p})\overline{\mathsf{PEB}}_{i_p}^2 + \overline{A_{i_p}}}}, && \text{(D.31)}\end{aligned}$$

where $\overline{A_{i_p}} = \sum_{k\neq i_p} A_k(\theta_k)$, $\overline{\mathsf{PEB}}_{i_p}$ is the PEB when sensor i_p is removed, that is

$$\overline{\mathsf{PEB}}_{i_p} = \sqrt{\frac{\sum_{k\neq i_p} A_k(\theta_k)}{\sum_{k\neq i_p} A_k(\theta_k)c_k^2 \sum_k A_k(\theta_k)s_k^2 - (\sum_{k\neq i_p} A_k(\theta_k)c_k s_k)^2}}, \tag{D.32}$$

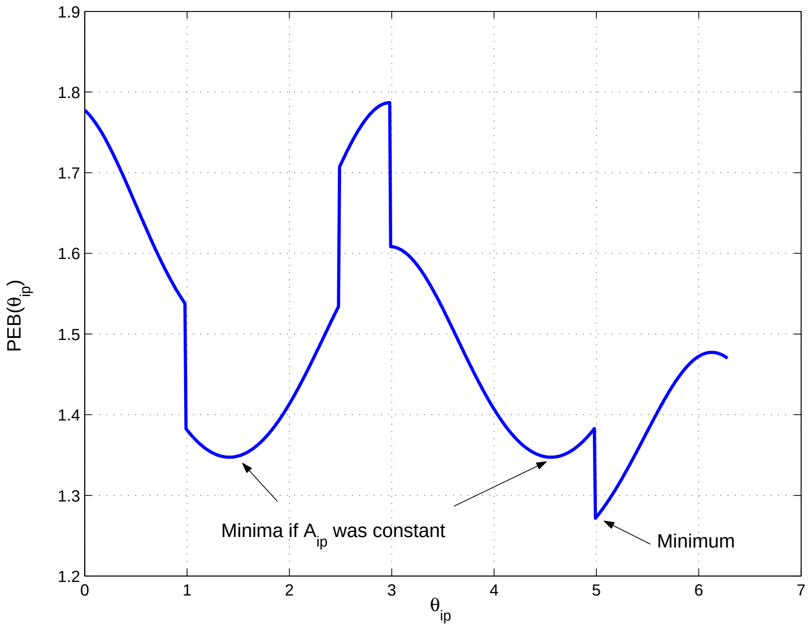

Figure D-2: When $A_{i_p}(\theta_{i_p})$ is piecewise constant, $\mathsf{PEB}(\theta_{i_p})$ shows discontinuity at the 4 transitions between arcs (compare to Fig. 6-2). The minimum may not be at one of the 2 points given by (6.7).

and where

$$G(\theta_{i_p}) = \sum_{k \neq i_p} A_k \sin^2(\theta_{i_p} - \theta_k). \tag{D.33}$$

If $A_{i_p}(\theta_{i_p})$ were constant with respect to θ_{i_p}, then $\mathsf{PEB}(\theta_{i_p})$ would be similar to the case considered in Section 6.3, with 2 minima and 2 maxima in $[0, 2\pi)$ (see Figure 6-2). However because A_{i_p} is piecewise constant, $\mathsf{PEB}(\theta_{i_p})$ is discontinuous at the L arc transitions, as shown on Figure D-2.

For each arc there are several possibilities:

- $\mathsf{PEB}(\theta_{i_p})$ is monotonous, in which case the minimum PEB on this arc occurs at one of its boundary;
- $\mathsf{PEB}(\theta_{i_p})$ is not monotonous on this arc:
  - If there is no inflection point that is a minimum, the minimum PEB on this arc occurs at one of its boundary;
  - If there is at least one inflection point that is a minimum, the minimum PEB on

this arc occurs at those. Its possible values are $\tilde{\theta_{i_p}}$ or $\tilde{\theta_{i_p}} + \pi$, given by (6.7).

In any case, to find the minimum of PEB we only need to test its value at the boundary points of each arc, and at $\tilde{\theta_{i_p}}$ and $\tilde{\theta_{i_p}} + \pi$. $\square$

Bibliography

[1] `http://peoc3t.monmouth.army.mil/FBCB2/FBCB2.html`.

[2] `http://robotics.eecs.berkeley.edu/ pister/SmartDust/`.

[3] Personal communications with Mark Homer, Charles Stark Draper Laboratory, May 2003.

[4] June 2004. Personal communications with Dr. Sanghyuk Park, Joshua Torgersen, and Huygil Ahn, MIT.

[5] J. S. Abel. Optimal sensor placement for passive source localization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, NM, April 1990.

[6] E. Amaldi, A. Capone, F. Malucelli, and F. Signori. UMTS radio planning: optimizing base station configuration. In *Proceedings of the IEEE Vehicular Technology Conference*, volume 2, pages 768–772, Vancouver, BC, September 2002.

[7] B. D. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, New Jersey, 1979.

[8] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Application to Tracking and Navigation*. Wiley, New York, NY, 2001.

[9] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2003.

[10] E. S. Biagioni and K. W. Bridges. The application of remote sensor technology to assist the recovery of rare and endangered species. *International Journal of High Performance Computing Applications*, 16(8):315–324, Fall 2002.

[11] Peter J. Bickel and Kjell Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*, volume 1. Prentice Hall, Upper Saddle River, NJ, second edition, 2001.

[12] R. Biswas, L. Guibas, and S. Thrun. A probabilistic approach to inference with limited information in sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, 2004.

[13] J. Borras, P. Hatrack, and N. B. Mandayam. Decision theoritic framework for NLOS identification. In *Proc. IEEE Vehicular Technology Conference*, volume 2, pages 1583–1587, Ottawa, Canada, May 1998.

[14] R. R. Brooks and S. S. Iyengar. *Advances in Distributed Sensor Technology*. Prentice Hall PTR, Upper Saddle River, NJ, 1995.

[15] R. R. Brooks and S. S. Iyengar. *Multi-Sensor Fusion: Fundamentals and Applications with Softwares*. Prentice Hall PTR, Upper Saddle River, NJ, 1998.

[16] N. Bulusu, J. Heidemann, and D. Estrin. Adaptive beacon placement. In *Proceedings of the Int. Conf. on Distributed Computing Systems*, pages 489–498, Phoenix, AZ, April 2001.

[17] J. Caffery, editor. *Wireless Location in CDMA Cellular Radio Systems*. Kluwer, Boston, MA, 2000.

[18] Dajana Cassioli, Moe Z. Win, and Andreas F. Molisch. The ultra -wide bandwidth indoor channel: from statistical model to simulations. *IEEE J. Select. Areas Commun.*, 20(6):1247–1257, August 2002.

[19] J. Chaffee and J. Abel. GDOP and the Cramer-Rao bound. In *Proceedings of the Position, Location and Navigation Symposium (PLANS)*, pages 663–668, Las Vegas, NV, April 1994.

[20] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Transactions on Computers*, 51:1448–1453, December 2002.

[21] N. Chandra and C. Knapp. Optimal hydrophone placement under random perturbations. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(2):860–864, May 1990.

[22] C. Chang and A. Sahai. Estimation bounds for localization. In *Proceedings of the IEEE Conference on Sensor and Ad Hoc Communications and Networks*, pages 415–424, Santa Clara, CA, October 2004.

[23] R. Chen and J. S. Liu. Mixture kalman filters. *J. R. Statist. Soc. B*, 62:493–508, 2000.

[24] K. W. Cheung and R. D. Murch. Optimising indoor base-station locations in coverage and interference limited indoor environments. In *IEE Proceedings in Communications*, volume 145, pages 445–450, December 1998.

[25] P. L. Chiu and Frank Y. S. Lin. A simulated annealing algorithm to support the sensor placement for target location. In *Proceedings of the CCECE*, volume 2, pages 867–870, Niagara Falls, Ontario, May 2004.

[26] R. Church and C. ReVelle. The maximal covering location problem. *Papers of The Regional Science Association*, (32):101–118, 1974.

[27] C. A. Coello Coello and alt., editors. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, 2002.

[28] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, second edition, 2001.

[29] F. Daum. Nonlinear filters: Beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine, Part 2*, 20(8):57–69, August 2005.

[30] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar. Sensor placement for grid coverage under imprecise detections. In *Proceedings of the Int. Conf. on Information Fusion*, pages 1581–1587, July 2002.

[31] Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors. *An introduction to sequential Monte-Carlo methods*, chapter Sequential Monte Carlo Methods in Practice. Springer-Verlag, New York, NY, 2001.

[32] Z. Drezner and H. W. Hamacher, editors. *Facility location: applications and theory*. Springer-Verlag, Berlin, 2002.

[33] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the ACM/IEEE IEEE International Conference on Mobile Computing and Networks*, pages 263–270, Seattle, WA, August 1999.

[34] C. Falsi, D. Dardari, L. Mucchi, and M. Z. Win. Time of arrival estimation for UWB localizers in realistic environments. In *Proceedings of the International Conference on Communications*, Istanbul, Turkey, June 2006. to appear.

[35] C. Fonseca and P. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization. In *Proceedings of the Fifth Int. Conf. on Genetic Algorithms*, pages 416–423, 1993.

[36] C. Fonseca and P. Fleming. *Genetic Algorithms in Engineering Systems*, chapter Multiobjective Genetic Algorithms. IEE, London, 1997.

[37] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: efficient position estimation for mobile robots. In *Proceedings of the National Conference on Artificial Intelligence*, Orlando, Florida, 1999.

[38] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

[39] M. Gavish and E. Fogel. Effect of bias on bearing-only target location. *IEEE Transactions on Aerospace and Electronics Systems*, 26:22–26, 1990.

[40] J. Geweke. Bayesian inference in economic models using Monte Carlo integration. *Econometrica*, 45(7):788–799, 1989.

[41] S. Gezici, H. Kobayashi, and H.V. Poor. Non-parametric non-line-of-sight identification. In *Proc. IEEE Vehicular Technology Conference*, volume 4, pages 2544–2548, Orlando, FL, October 1998.

[42] S. Gezici, Zhi Tian, G.B. Giannakis, H. Kobayashi, A.F. Molisch, H.V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 22:70–84, July 2005.

[43] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Reading, MA, 1989.

[44] N. J. Gordon, D. J. Salmon, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, 1993.

[45] J. K. Han, B. S. Park, Y. S. Choi, and H. K. Park. Genetic approach with a new representation for base station placement in mobile communications. In *Proceedings of the IEEE Vehicular Technology Conference*, volume 4, pages 2703–2707, Atlantic City, NJ, October 2001.

[46] T. Hegazy and G. Vachtsevanos. Sensor placement for isotropic source localization. In *Proceedings of the Second International Workshop on Information Processing in Sensor Networks*, Palo Alto, CA, April 2003.

[47] J. P. How and J. J. Deyst. Advanced estimation for GPS and inertial navigation. MIT 16.324 classnotes, Spring 2004.

[48] A. Howard and M. J. Mataric. Cover me! a self-deployment algorithm for mobile sensor networks. In *Proceedings of the Int. Conf. on Robotics and Automation*, Washington, DC, May 2002.

[49] A. Howard, M. J. Mataric, and G. S. Sukhatme. chapter Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem, pages 299–308. Springer, 2002.

[50] A. Howard, M. J. Mataric, and G. S. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Autonomous Robots Special Issue on Intelligent Embedded Systems*, 13(2):113–126, 2002.

[51] S. S. Iyengar and Richard R. Brooks, editors. *Distributed Sensor Networks*. CRC Press, Boca Raton, FL, 2004.

[52] D. B. Jourdan, D. Dardari, and Moe Z. Win. Position error bound and localization accuracy outage in dense cluttered environments. In *Proceedings of IEEE International Conference on Ultra-Wideband*, Waltham, MA, September 2006.

[53] D. B. Jourdan, D. Dardari, and Moe Z. Win. Position error bound for UWB localization in dense cluttered environments. In *Proceedings of IEEE International Conference on Communications*, Istanbul, Turkey, June 2006.

[54] D. B. Jourdan, D. Dardari, and Moe Z. Win. Position error bound for UWB localization in dense cluttered environments. *IEEE Transactions on Aerospace and Electronics Systems*, March 2006. Submitted.

[55] D. B. Jourdan and O. L. de Weck. Layout optimization for a wireless sensor network using a multi-objective genetic algorithm. In *Proceedings of the IEEE Vehicular Technology Conference*, volume 5, pages 2466–2470, Milan, Italy, May 2004.

[56] D. B. Jourdan and O. L. de Weck. Multi-objective genetic algorithm for the automated planning of a wireless sensor network to monitor a critical facility. In *Proceedings of the SPIE Defense and Security Symposium*, volume 5403, pages 565–575, Orlando, FL, April 2004.

[57] D. B. Jourdan, J. J. Deyst, M. Z. Win, and N. Roy. Monte-Carlo localization in dense multipath environments using UWB ranging. In *Proceedings of IEEE International Conference on Ultra-Wideband*, pages 314–319, Zurich, CH, September 2005.

[58] D. B. Jourdan and N. Roy. Optimal sensor placement for agent localization. In *Proceedings of the IEEE/ION Conference on Position, Location, and Navigation (PLANS)*, San Diego, CA, April 2006.

[59] D. B. Jourdan and N. Roy. Optimal sensor placement for agent localization. *ACM Transactions on Sensor Networks*, March 2006. Submitted.

[60] Elliott D. Kaplan, editor. *Understanding GPS: Principles and Applications*, chapter Performance of Standalone GPS. Artech House, Boston, MA, 1996.

[61] K. Kar and S. Banerjee. Node placement for connected coverage in sensor networks.

[62] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220, 4598:671–680, 1983.

[63] A. M. Ladd, K. E. Bekris, A. Rudys, L. E. Kavraki, and D. S. Wallach. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Robotics and Automation*, 20(3), June 2004.

[64] E. G. Larsson. Cramér-Rao bound analysis of distributed positioning in sensor networks. *IEEE Signal Processing Letters*, 11(3):334–337, March 2004.

[65] Joon-Yong Lee and Robert A. Scholtz. Ranging in a dense multipath environment using an UWB radio link. *IEEE Journal on Selected Areas in Communications*, 20(9):1677–1683, December 2002.

[66] N. Levanon. Lowest GDOP in 2-D scenarios. *IEE Proceedings-Radar, Sonar Navigation*, 147(3):149–155, March 2000.

[67] P. H. Lommel. An extended kalman filter extension of the augmented markov decision process. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, June 2005.

[68] Z. N. Low, J. H. Cheong, C. L. Law, W. T. Ng, and Y. J. Lee. Pulse detection algorithm for Line-of-Sight (LOS) UWB ranging applications. *IEEE Antennas and Wireless Propagation Letters*, 4:63–67, 2005.

[69] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, November 2004. To appear.

[70] J.B. McKay and M. Pachter. Geometry optimization for GPS navigation. In *Proceedings of the 36th Conference on Decision and Control*, San Diego, CA, December 1997.

[71] S. Meguerdichian and M. Potkonjak. Low power 0/1 coverage and scheduling techniques in sensor networks. Technical Report 030001, UCLA, January 2003.

[72] A. Mehzer and A. Stulman. The maximal covering location problem with facility placement on the entire plane. *Journal of Regional Science*, 22(3), 1982.

[73] H. Meunier, E. Talbi, and P. Reininger. A multiobjective genetic algorithm for radio network optimization. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 317–324, San Diego, CA, July 2000.

[74] P. Minvielle. Decades of improvements in re-entry ballistic vehicle tracking. *IEEE Aerospace and Electronic Systems Magazine, Part 1*, 20(8):CF1–CF14, August 2005.

[75] J. Nemeroff, L. Garcia, D. Hampel, and S. DiPierro. Application of sensor network communications. In *Proceedings of the IEEE Military Communications Conference*, volume 1, pages 336–341, October 2001.

[76] Kaveh Pahlavan, Xinrong Li, and Juha-Pekka Makela. Indoor geolocation science and technology. *IEEE Communications Magazine*, pages 112–118, February 2002.

[77] Y. Qi and H. Kobayashi. Cramér-Rao lower bound for geolocation in non-line-of-sight environment. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 2473–2476, Orlando, FL, May 2002.

[78] Y. Qi and H. Kobayashi. On geolocation accuracy with prior information in non-line-of-sight environment. In *Proceedings of the IEEE Vehicular Technology Conference (VTC 2002)*, pages 285 – 288, Vancouver, BC, September 2002.

[79] Y. Qi, H. Suda, and H. Kobayashi. On time-of-arrival positioning in a multipath environment. In *Proceedings of the 60th IEEE Vehicular Technology Conference (VTC 2004-Fall)*, Los Angeles, CA, September 26-29 2004.

[80] Andrea Ridolfi and Moe Z. Win. Ultrawide bandwidth signals as shot-noise: a unifying approach. *IEEE J. Select. Areas Commun.*, 2006. to appear.

[81] R. Rose. A smart technique for determining base-station locations in an urban environment. In *Proceedings of the IEEE Vehicular Technology Conference*, volume 50, pages 43–47, Rhodes, Greece, May 2001.

[82] S. Salapaka and A. Khalak. Capacity constraints on locational optimization problems. Technical report, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, Cambridge, MA, March 2003.

[83] A. Savvides, W. Garber, S. Adlakha, R. Moses, and M.B. Srivastava. On the error characteristics of multihop node localization in ad-hoc sensor networks. In *Proceedings of the Second International Workshop on Information Processing in Sensor Networks (IPSN'03)*, pages 317–332, Palo Alto, CA, April 2003.

[84] C. Schurgers and M. B. Srivastava. Energy efficient routing in wireless sensor networks. In *Proceedings of the MILCOM Conference*, pages 357–361, McLean, VA, October 2001.

[85] X. Sheng and Y. H. Hu. Sensor deployment for source localization in wireless sensor network system. 2003.

[86] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, Dallas, TX, October 1998.

[87] A. Sinha, T. Kirubarajan, and Y. Bar-Shalom. Optimal cooperative placement of GMTI UAVs for ground target tracking. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, March 2004.

[88] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka Priyantha. Tracking moving devices with the Cricket location system. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, Boston, MA, June 2004.

[89] J. J. Spilker, Jr. GPS signal structure and performance characteristics. *Journal of the Institute of Navigation*, 25(2):121–146, Summer 1978.

[90] Watcharapan Suwansantisuk, Moe Z. Win, and Lawrence A. Shepp. On the performance of wide-bandwidth signal acquisition in dense multipath channels. *IEEE Trans. Veh. Technol.*, 54(5), September 2005.

[91] A. Suzuki and Z. Drezner. The p-center location problem in an area. *Location Science*, 4(1/2):69–82, 1996.

[92] F. Tehoffo-Talom, B. Uguen, E. Plouhinec, and G. Chassay. A site-specific tool for UWB channel modeling. In *2004 International Workshop on Ultra Wideband Systems*, pages 61–65, Kyoto, Japan, May 2004.

[93] H. L. Van Trees. *Detection, Estimation and Modulation Theory*. Wiley, New York, NY, 1968.

[94] S. Venkatraman and J. Caffery. A statistical approach to non-line-of-sight BS identification. In *Proc. IEEE 5th Int. Symposium Wireless Personal Multimedia Communications*, volume 1, pages 296–300, Honolulu, Hawaii, October 2002.

[95] H. Wang, L. Yip, K. Yao, and D. Estrin. Lower bounds of localization uncertainty in sensor networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 917–920, Montreal, QC, May 2004.

[96] Moe Z. Win. A unified spectral analysis of generalized time -hopping spread-spectrum signals in the presence of timing jitter. *IEEE J. Select. Areas Commun.*, 20(9):1664–1676, December 2002.

[97] Moe Z. Win and Robert A. Scholtz. Impulse radio: How it works. *IEEE Commun. Lett.*, 2(2):36–38, February 1998.

[98] Moe Z. Win and Robert A. Scholtz. On the energy capture of ultra-wide bandwidth signals in dense multipath environments. *IEEE Commun. Lett.*, 2(9):245–247, September 1998.

[99] Moe Z. Win and Robert A. Scholtz. On the robustness of ultra-wide bandwidth signals in dense multipath environments. *IEEE Commun. Lett.*, 2(2):51–53, February 1998.

[100] Moe Z. Win and Robert A. Scholtz. Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications. *IEEE Trans. Commun.*, 48(4):679–691, April 2000.

[101] Moe Z. Win and Robert A. Scholtz. Characterization of ultra-wide bandwidth wireless indoor communications channel: A communication theoretic view. *IEEE J. Select. Areas Commun.*, 20(9):1613–1627, December 2002.

[102] R. Yarlagadda, I. Ali, N. Al-Dhahir, and J. Hershey. GPS GDOP metric. *IEE Proceedings-Radar, Sonar Navigation*, 147(5):259–264, May 2000.

[103] Hong Zhang. Two-dimensional optimal sensor placement. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(5), May 1995.

[104] Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. In *Proceedings of the IEEE Infocom Conference*, volume 2, pages 1293–1303, San Francisco, CA, April 2003.