Optimization of Anchor Nodes Placement in Wireless Localization Networks

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera Italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Katarina Balać

under the supervision of
Prof. Miroslaw Malek and Mauro Prevostini

March 2019

Dissertation Committee

Prof. Cesare Alippi	Università della Svizzera Italiana, Switzerland
Prof. Evanthia Papadopoulou	Università della Svizzera Italiana, Switzerland
Prof. Antonio Capone	Politecnico di Milano, Italy
Prof. Hans-Peter Schwefel	Aalborg University, Denmark

Dissertation accepted on 20 March 2019

Research Advisor

Prof. Miroslaw Malek

Co-Advisor

Mauro Prevostini

PhD Program Director

Prof. Walter Binder and Prof. Olaf Schenk

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Katarina Balać
Lugano, 20 March 2019

Abstract

This work focuses on optimizing node placement for time-of-flight-based wireless localization networks. Main motivation are critical safety applications. The first part of my thesis is an experimental study on in-tunnel vehicle localization. In-tunnel localization of vehicles is crucial for emergency management, especially for large trucks transporting dangerous goods such as inflammable chemicals. Compared to open roads, evacuation in tunnels is much more difficult, so that fire or other accidents can cause much more damage. We provide distance measurement error characterization inside road tunnels focusing on time of flight measurements. We design a complete system for in-tunnel radio frequency time-of-flight-based localization and show that such a system is feasible and accurate, and that few nodes are sufficient to cover the entire tunnel.

The second part of my work focuses on anchor nodes placement optimization for time-of-flight-based localization networks where multilateration is used to obtain the target position based on its distances from fixed and known anchors. Our main motivation are safety at work applications, in particular, environments such as factory halls. Our goal is to minimize the number of anchors needed to localize the target while keeping the localization uncertainty lower than a given threshold in an area of arbitrary shape with obstacles. Our propagation model accounts for the presence of line of sight between nodes, while geometric dilution of precision is used to express the localization error introduced by multilateration. We propose several integer linear programming formulations for this problem that can be used to obtain optimal solutions to instances of reasonable sizes and compare them in terms of execution times by simulation experiments. We extend our approach to address fault tolerance, ensuring that the target can still be localized after any one of the nodes fails.

Two dimensional localization is sufficient for most indoor applications. However, for those industrial environments where the ceiling is very high and the worker might be climbing or be lifted from the ground, or if very high localization precision is needed, three-dimensional localization may be required. Therefore, we extend our approach to three-dimensional localization. We derive the

expression for geometric dilution of precision for 3D multilateration and give its geometric interpretation.

To tackle problem instances of large size, we propose two novel heuristics: greedy placement with pruning, and its improved version, greedy placement with iterative pruning. We create a simulator to test and compare all our proposed approaches by generating multiple test instances. For anchor placement for multilateration-based localization, we obtain solutions with below 2% anchors overhead with respect to the optimum on average, with around 5s average execution time for 130 candidate positions. For the fault-tolerant version of the same problem, we obtain solutions of around 1% number of anchors overhead with respect to the optimum on average, with 0.4s execution time for 65 candidate positions, by using greedy heuristic with pruning. For 3D placement, the greedy heuristic with iterative pruning produced results of 0.05% of optimum on average, with average execution time of around 6s for 250 candidate positions, for the problem instances we tested.

Acknowledgements

I would like to thank my advisor Prof. Miroslaw Malek for his continuous support of my PhD research, for numerous fruitful discussions, for his patience, enthusiasm and his wisdom. Also, I would like to thank my coadvisor Mauro Prevostini for his support and useful suggestions regarding my research, as well as for always helping provide funding for my work. Working with Mauro not only helped increase my technical and research skills but also my knowledge of Italian.

Furthermore, I am grateful to many current and former members of ALaRI institute, particularly to Dr. Marcello Mura, Dr. Antonio Taddeo, Dr. Onur Derin, Dr. Igor Kaitović, Dr. Jelena Milošević, Dr. Giovanni Mariani, Dr. Francesco Regazzoni, Rami Baddour, Dr. Slobodan Luković, Dr. Alberto Ferrante, Umberto Bondi and many others for collaboration, support and their friendship.

Discussions with Murodzhon Akhmedov from IDSIA institute have been very helpful in shaping my research. Also, many of my former students contributed to enhancing my work and my knowledge. My research greatly benefited from many insightful comments of my committee members.

I am grateful to Interreg and Università della Svizzera Italiana for funding my work. My special gratitude goes to Piano del Traffico Luganese for providing access to Vedeggio-Cassarate road tunnel.

Finally, I would like to thank my family for their unconditional love and support.

Contents

Figures

Tables

Chapter 1

Introduction

Localization aims at finding position of an object of interest. A wide range of modern applications, from entertainment to critical safety systems, can benefit from determining positions of objects, persons or animals. Some of the examples are navigation of vehicles, wildlife monitoring, care for elderly and infirm people, museum guides, and safety at work. Global positioning system (GPS) is a preferred solution for outdoor localization. However, GPS requires the line of sight to satellites, and therefore, cannot be used indoors or even outdoors where large buildings obstruct the visibility to the sky. Therefore, effective solutions for indoor localization systems are needed.

Localization techniques that take advantage of wireless modules are commonly used for indoor localization. A set of wireless nodes, called *anchors*, are placed at fixed and known positions in the area that needs to be monitored. We will refer to the object of interest, that needs to be located, as *target* or *tag*. The target object is equipped with a wireless node as well. The target is mobile and needs to be localized at all times. As the target moves, different anchors will be used to localize it, always those that are near the target.

The placement of anchors greatly influences the system functionality and localization precision. The number of nodes placed directly affects the system cost and energy consumption. Therefore, our goal is finding placement strategies for localization networks. Our work primarily focuses on critical safety applications.

We address several scenarios. All of them use time-of-flight (ToF)-based localization, where distances are measured by measuring the time a signal takes to travel between the anchor and the target. Such localization offers high precision.

Our first scenario is in-tunnel vehicle localization. We design a system for localization and characterize the distance measurement error in road tunnels. We perform experiments in a road tunnel and propose best practices for placing

anchors in this scenario.

In the second scenario, our goal is to optimize the placement of anchor nodes for a ToF-based localization network, so as to minimize their number, thus minimizing the cost and the power consumption of the network, while ensuring that the target can be reliably localized at each point in the area of interest, in 2D. Multilateration is a process used to combine the measured distances to obtain the target position. We primarily have in mind critical safety at work applications in factory halls. We extend this problem to include fault tolerance. Specifically, we still want to be able to localize the target at any point in the area of interest, even if one of the anchor nodes fails. Two-dimensional localization is sufficient in many cases, as we can often assume that the target is only moving on the ground. However, it is not always the case. In factories, the worker may climb or be lifted from the floor. Therefore, we extend our approach to 3D scenarios. Thus, our work covers 1D, 2D and 3D scenarios.

We take into consideration the obstacles that can obstruct wireless propagation or significantly affect the distance measurement precision. The metric we use is Geometric Dilution of Precision (GDoP), which is commonly used for expressing precision of localization systems. We also derive the expression for GDoP for 3D multilateration.

Much of our work focuses on exact approaches. We show that exact solutions can be found for problem instances of anchor placement for localization of reasonably large sizes. Exact solutions to these problem instances allow to evaluate the quality of non-exact algorithms, by running simulation experiments and comparing the number of nodes to that placed by exact methods. Our exact approaches are based on integer linear programing (ILP). ILP enables to obtain exact solutions to instances of NP hard problems in feasible computation times by using state of the art ILP solvers. Where the problem instance size is too large to find the exact solution, we propose greedy placement with pruning, and its improved version, greedy placement with iterative pruning, that allow to solve problem instances of very large sizes in short computing times. For the problem instances we tested, our non-exact approaches produced results of between 0.05 % and 1.75 % in terms of average overhead of the number of anchors placed compared to the optimum.

1.1 Motivation

Our work is motivated by two different scenarios we encountered in two projects. Both of them involve localization for critical safety applications. Our first sce-

nario is in tunnel vehicle localization, that was required by PTA [2010] project. The project focuses on tracking large trucks transporting dangerous chemicals, and our group focused on localizing vehicles inside tunnels. GPS signals that are usually used to localize vehicles are not available in tunnels. Accidents, such as fire, are especially dangerous and cause much more damage in tunnels than in the open areas, as evacuation is more difficult. Therefore, localization in tunnels is particularly important, especially for vehicles transporting dangerous and inflammable chemicals. As wireless propagation properties in road tunnels are significantly different than in other environments, localization systems need to be designed for road tunnels specifically.

The second scenario, required by the PTA-DESTNATION [2015] project, focuses on safety at work applications. In particular, the system can be deployed in a factory hall, in order to localize workers in case of danger and thus increase their safety. Apart from the localization functionality, both the anchors and the target node may have other functionalities that can enhance the workers' safety. For example, the target node can be equipped with an inertial system that can detect the fall, and anchor nodes can be equipped with sensors that detect smoke or dangerous chemicals. In case of an accident, the system can identify the emergency situation and send a notification about it. This notification has to include the worker's position, so that the person in danger can be rescued. We include 3D scenarios as the factory halls typically have tall roofs, and the worker may be climbing or lifted off the floor.

For highly critical applications such as safety systems, using high precision localization provided by ToF is preferred to cheaper and less precise alternatives, such as WiFi or bluetooth-based solutions. For the same reason, the application criticality, we also address fault-tolerant anchor placement in our work.

1.2 Evaluation Criteria

In our simulation experiments we often use the execution time for given problem parameters in order to compare and evaluate different solution approaches. We generate a large number of random instances of a problem with the same parameters, run the optimization for all the problem instances and look at the execution time statistics. Even though the execution time depends on multiple factors such as the processor used, the memory architecture and the operating system, it provides a way to compare different approaches. Looking at how the execution time increases with increasing problem size gives us an insight into the size of a problem instance for which we can get a solution in realistic times,

or in a given time. Comparing execution times is also accepted when addressing practical problems in literature.

Our goal is to minimize the number of nodes in the network while the network meets the localization requirements. In order to deal with non-polynomial complexity of exact algorithms, we propose heuristic algorithms. In case of heuristics, where no optimal solution is guaranteed, we use the number of nodes placed by the heuristic compared to the number placed by the exact approach as the measure of quality. Again, we run the optimizations for a large number of randomly generated instances and look at the statistics on the number of nodes placed. We aim to set the problem parameters such that they correspond to real-life problem instances.

1.3 Contents and Contributions

Our main goal is to optimize the placement of anchor nodes for a ToF-based localization network, so as to minimize their number, thus minimizing the cost and power consumption of the network, while ensuring that the target can be reliably localized at each point in the area of interest. We extend this problem to include fault tolerance. Specifically, we still want to be able to localize the target at any point in the area of interest, even if one of the anchor nodes fails. We address 1D, 2D and 3D localization.

Main contributions are:

- An experimental study of in-tunnel vehicle localization which shows that radio frequency (RF) ToF-based in-tunnel vehicle localization is feasible and provides recommendations for node placement in such systems.

- Time of flight distance measurement error characterization for road tunnels which is useful for designing in-tunnel localization systems.

- An error model and an uncertainty function for a UWB-based localization system that takes into account both Line of Sight (LoS) and GDoP effect on localization uncertainty, for both 2D and 3D localization.

- Three different ILP formulations for anchor placement for trilateration-based localization in 2D and their comparison by simulation experiments. The formulations can be used to obtain optimal solutions to problem instances of reasonable sizes.

- ILP formulation for fault-tolerant placement for multilateration-based localization in 2D that can be used to obtain optimal solutions to problem instances of sizes that are relevant for practical applications.
- ILP formulation for placement for multilateration-based localization in 3D that can be used to obtain optimal solutions to problem instances of sizes that are relevant for practical applications.
- Two novel heuristics which allow to tackle instances of all three above problems of large sizes: greedy placement with pruning, and its improved version - greedy placement with iterative pruning. For the problem instances we tested, these heuristics produced results of between 0.05 % and 1.75 % average overhead of the number of anchors placed compared to the optimum values.
- GDoP expression for 3D multilateration and its geometric interpretation.
- A simulator for testing and comparing proposed approaches as well as solving instances of real-life problems.

1.3.1 Simulator

We created a simulator that either accepts an existing floor plan as input, or generates the floor plan randomly with given parameters. Random generation of floor plan is useful for testing and comparing proposed approaches, while the existing floor plan feature can be used to tackle the real-life problems. More information about the simulator is given in Section 6.6. We used this simulator to generate test scenarios and run all our simulation experiments.

1.3.2 Contents

In Chapter 2, an overview of different localization techniques, with the emphasis on ToF-based techniques, as well as the description of localization system we consider in this work is given. In Chapter 3, the problem statement is given, and in Chapter 4, we present a literature review on node placement for wireless sensor networks, with the emphasis on placement for localization networks. In Chapter 5 we present our system for in-tunnel vehicle localization, the experimental study in a road tunnel, as well as our guidelines for node placement in road tunnels. The chapter also includes a literature overview on wireless propagation in road tunnels. In Chapter 6 we propose three ILP formulations and the greedy

placement with pruning heuristic for 2D placement of nodes for a localization network. In chapter 7, we extend our solutions to 3D scenarios, propose a faster heuristic: greedy placement with iterative pruning, and derive the expression for GDoP for 3D multilateration. In Chapter 8 we conclude the work.

Chapter 2

Preliminaries

Outdoor localization is almost exclusively based on global navigation satellite systems (GNSSs). There are several such systems. The United States NAVSTAR GPS, The European Union's Galileo and the Russian GLONASS are fully globally operational GNSSs. The People's Republic of China is in the process of expanding its regional Beidou navigation system into the global navigation system. The different GNSSs are using almost the same positioning principles. These principles are based on the assumption that at a given place on the earth surface a GNSS receiver is able to "see" a certain number of GNSS satellites. Each satellite of a given GNSS constellation has a very accurate atomic on board time clock which is continuously synchronized with respect to a common reference. Each satellite individually broadcasts a set of parameters that can be inserted into a set of known orbital equations (Ephemeris) allowing a receiver to compute, within a few meters, the position of a satellite at a given time. GNSSs were initially designed for military applications, but are now widely available for civil uses. GNSS receivers are inexpensive, therefore, many devices in everyday use are equipped with such receivers and can take advantage of GNSS localization. However, the supplier of satellite navigations signals can also deny their availability. Another disadvantage of GNSSs is that the line of sight to the satellites is necessary, which means, the sky has to be visible in order to use it. This condition is not met in indoor environments, tunnels, underground, and underwater environments, and in so called urban canyons, where tall buildings obstruct the visibility to the sky. Therefore, many different solutions and systems have been devised for indoor environments, with different implementation costs and different precision, suitable for different environments and needs. Many of these systems are based on radio signals. Other options are cameras, or in case of underwater networks, sound waves.

Localization systems can be roughly divided into range-free and range-based systems. Range-free systems are quick and easy to implement as a rule, but they do not provide the same precision and reliability as range-based systems. They are suitable for applications where high precision is not necessary, for example location-based advertisements in shopping malls. We know that the target is in the intersection of areas covered by the anchors it can detect, so we can approximately determine its position. As the areas covered by different anchors are of irregular shapes, this is most commonly implemented as so called fingerprinting. There is a setup phase, which needs to be executed only once, in which signals from all anchors are recorded in the entire area of interest. Then, during the localization phase, the signals that the target can detect are compared with the prerecorded database of signals and the position is determined based on the results. Existing WiFi access points can be used as anchors for such localization, but usually additional anchors are needed. Bluetooth nodes are often deployed for this purpose, as it is an inexpensive solution. Another range-free method is estimating distances based on the number of hops. Signal cannot always reach the receiver directly from the transmitter, but may have to be forwarded via other nodes. The number of these intermediate nodes is the number of hops. As range-based solutions are much more precise and reliable, and our work focuses on safety critical applications, we use range-based solutions.

For range-based systems, localization process consists of two phases. In the first phase, distances between the anchors and the target are determined. Measuring distances is often referred to as ranging. Alternatively, angles can be measured instead of distances. In the second phase, distances (or angles) are combined in order to obtain the target position. Angles can be measured by using directional antennas and antenna arrays. These solutions require dedicated hardware and their precision is only moderate. Also, angles can be measured by using cameras.

The most commonly used methods for measuring distances by radio signals are received signal strength indication (RSSI) and time of flight (ToF). RSSI measures the signal strength at the receiver, and estimates distance based on the path loss, the signal attenuation from transmitter to receiver. In theory, the signal strength decreases logarithmically with the distance. However, different signal components take different paths and reflect from obstacles in the environment. The sum of these components is what is detected at the receiver, and it is significantly different from what we would obtain if the signal only reached the receiver through the direct path and without any reflections. This effect is called multipath propagation. It is the main reason why RSSI measurements are not very reliable. Compared to RSSI, ToF-based methods are much more precise (see

for example Vossiek et al. [2003]). Part of our work was comparing ToF and RSSI distance measurements precision in the tunnel environment. See Chapter 5 for more details on our work on in-tunnel vehicle localization and our experimental comparison of RSSI with ToF. Even though the road tunnel causes far fewer reflections than the typical indoor or factory environment, we show experimentally that ToF outperforms RSSI localization by far. Therefore, we will focus on ToF-based localization in this work.

2.1 ToF-based Distance Measurements

While ToF measurements are based on the simple principle of measuring the signal propagation time and multiplying by the known speed of signal propagation, they require complex hardware for very precise time measurements and time synchronization. There are several different variants of distance measurements all using ToF as the basic principle. They all have different requirements on synchronization and time measurement precision on both anchor and target nodes. Time Difference of Arrival (TDoA) is one of the commonly used methods (Gustafsson and Gunnarsson [2003]). Target broadcasts a message to multiple anchor nodes at the same time. The anchors record the reception time of the message, and send the reception time-stamp to one central location. Using the time difference from several pairs of anchors, the unknown transmission time can be calculated along with the unknown position of the target. Each pair of anchor measurements constraints the target position to a hyperbola. The downside of this method is that very precise time synchronization of all anchor nodes is required. As it is difficult to achieve such synchronization wirelessly, cables are usually used for this purpose in indoor environments.

Another distance measurement method based on ToF is two way time of flight (TWToF) (Jiang and Leung [2007]). The target sends a message to each anchor in turn and waits for the response. This measurement procedure is shown in Figure 2.1. The propagation time is calculated as:

$$t_p = \frac{t_{round} - t_{reply}}{2}, \tag{2.1}$$

where t_{round} is the time measured by the target from sending the request message until the response arrives. The anchor processing time t_{reply} can be recorded by the anchor and transmitted back to the target. Another solution is to use special hardware which can guarantee that the response is sent at exactly defined time after the reception of the message. With TWToF the synchronization constraint

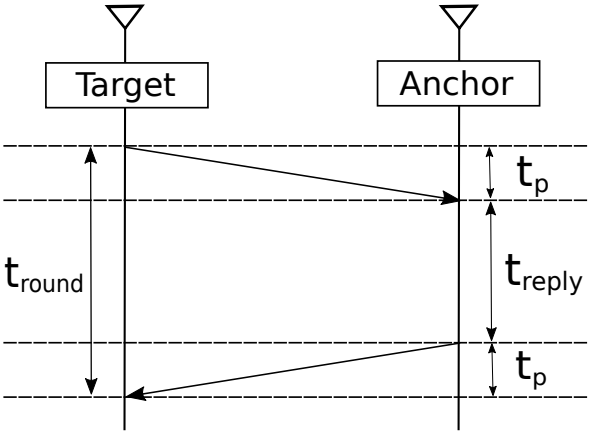


Figure 2.1. Two way time of flight.

can be removed for all the devices in the system. In this work, we assume that the distances are measured by using TWToF. In this way, there is no need for synchronization cables and deployment is simplified.

Silva et al. [2014] propose and test an Ultra Wide Band (UWB) based localization system using multilateration in a small setting with four anchor nodes, and show that such localization is accurate, fast and non-complex when line of sight is present.

2.2 Ultra Wide Band

While it is possible to use time of flight with radio frequency signals, ultra wide band is much more commonly used with time-of-flight-based localization, especially when high precision is necessary, in cluttered environments and for safety critical applications. The use of UWB has many advantages making it an attractive solution for positioning and tracking applications. The use of ultra-short pulses makes measurements with excellent time resolution possible. The transmitted signals are well protected from unauthorized interception. The transmission is immune to narrowband interference and multipath phenomenon. The average power of UWB emission is very low, so transmitters have low energy requirements (Dardari et al. [2009], Gezici et al. [2005]).

Due to all these reasons, ultra wide band has become a standard for high precision indoor localization. As it is cheaper and provides sufficient precision in tunnel environment, we have used radio frequency time of flight for our in-tunnel experiments (see Chapter 5). In the rest of our work, for 2D and 3D localization, we assume that UWB modules are used. UWB module measures the

distances from anchor nodes to the target by using ToF technique. Two nodes can communicate only if the distance between them is less than the node range. The typical range of UWB nodes is around $20\,m$.

2.3 Line of Sight

Line of Sight (LoS) is a term that is used to indicate that there are no objects in between the nodes that can significantly obstruct their communication or distance measurements, they can "see" each other. While UWB signals are highly resistant to multipath propagation due to their high bandwidth, the absence of line of sight between nodes adversely affects the UWB distance measurements and significantly reduces the ranging accuracy. Therefore, it has to be taken into account when designing UWB localization systems. Ranging error of UWB systems has been studied by Alavi and Pahlavan [2006]. They show that the error can be modeled as a Gaussian random variable, with parameters depending on presence of LoS. In presence of LoS, the mean error value is 0, while standard deviation is low, and increases with the distance d between the transmitter and the receiver as $\sigma = c \cdot log(1 + d)$. In the absence of LoS, error standard deviation is much higher, while its mean value is positive. While small objects such as chairs do not present an obstacle to UWB measurements, some walls, pillars and large machines do. Therefore, we take LoS requirement into consideration in our work. Moreover, our experiments in the tunnel environment have shown that RF ToF is also sensitive to Non Line of Sight (NLoS) conditions (See Chapter 5).

Many solutions have been proposed to mitigate the adverse effect of NLoS measurements on localization accuracy once the localization system is deployed and operating. See, for example, Khodjaev et al. [2010], Schroeder et al. [2007], Chen et al. [2012], Marano et al. [2010]. These works focus on detecting distance measurements that are affected by NLoS condition, among all the measurements between the target and different anchors. Some of them try to mitigate the effect of NLoS by countering the bias introduced. Others focus on detection only and discard the affected measurements. NLoS condition can be identified by using variance of distance measurements, channel statistics such as signal to noise ratio at the receiver, or by using the maps of the environment and observing the previous position of the target. Our work is complementary to these efforts. We address the problem of how to deploy the localization system by optimizing anchor node placement, to ensure that there is sufficient number of measurements not affected by NLoS at each point of interest.

2.4 Combining Distances to Obtain Position

The second stage of localization aims at combining distance or angle measurements obtained in the first stage to calculate the target position. The method used depends on the measurement principle. When angles are measured, the process used to combine them is called triangulation. At least two anchors are needed for triangulation in 2D. The measured angles will determine the directions from anchors to the target. The target is located where those directions intersect.

The method for combining the distance measurements from the target to several anchors to obtain the target position is known as multilateration, or trilateration, when exactly three anchors are used. This technique is based on the principle that, knowing the anchor positions and the distances from target to the anchors $\{r_i\}_{i=1,2,3}$, the target position can be determined as the intersection of circles (or spheres, for 3D localization) placed around the anchors, with radii equal to those distances $\{r_i\}_{i=1,2,3}$ (Figure 2.2(a)). At least three non-collinear

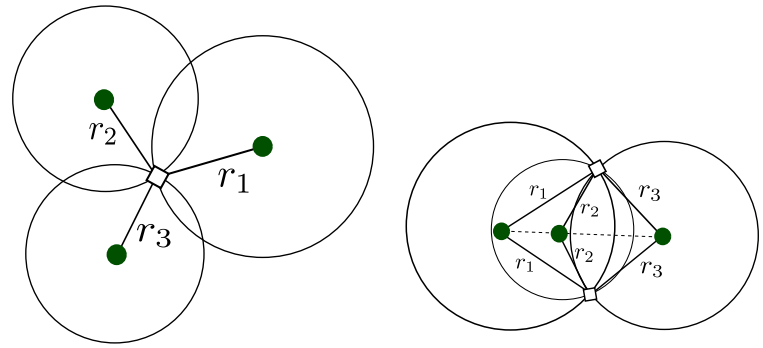


Figure 2.2. Trilateration (a) The target, indicated by the diamond shape, is located at the intersection of circles around three anchors, indicated by dots (green). (b) Three collinear anchors cannot localize the target.

anchor nodes are needed to localize the target in two dimensions. As two circles intersect at two different points, the third anchor is required to distinguish between these two and determine the target position. The three anchors must not be collinear, because three circles would intersect at two different points and that would create ambiguity (Figure 2.2(b)). Similarly, for 3D localization, at least four non-coplanar anchors are needed.

As there is always a measurement error, the circles will not intersect at one point. Weighted Least Squares (WLS) method can be used to determine the tar-

get position based on distance measurements (see for example Spirito [2001]). For this method, we need to have an estimate of the target position $(x^{(0)}, y^{(0)}, z^{(0)})$. The estimate is typically the last known position of the target. By defining

$$\mathbf{x} = [\Delta x \ \Delta y \ \Delta z]^T \tag{2.2}$$

as the vector difference between the current target position and the previous estimated position.

$$\Delta x = x - x^{(0)}, \ \Delta y = y - y^{(0)}, \ \Delta z = z - z^{(0)} \tag{2.3}$$

and linearizing around the target, we obtain the current target position (x, y, z) by finding the solution of the linear problem $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} = [u_{i,x} \ u_{i,y} \ u_{i,z}]_{i \in \mathsf{A}}, \tag{2.4}$$

is a unit vector originating at the target, directed towards the anchor i and

$$\mathbf{b} = [d_i^0 - d_i]_{i \in \mathsf{A}}, \tag{2.5}$$

where d_i^0 and d_i are distance at previous iteration and measured distance from anchor i to target respectively and A is the set of anchors. The WLS solution of this system is the one that that minimizes the scalar cost function:

$$(\mathbf{Ax} - \mathbf{b})^{\mathrm{T}} \mathbf{Q_b}^{-1} (\mathbf{Ax} - \mathbf{b}), \tag{2.6}$$

where $\mathbf{Q_b}$ is the covariance matrix of $\mathbf{b}$ (see Kay [1993]). If distance measurement errors are independent and identically distributed with standard deviation σ_D, $\mathbf{Q_b}$ is a diagonal matrix $\mathbf{Q_b} = \mathrm{diag}(\sigma_D^2)$.

2.5 Geometric Dilution of Precision

As discussed, the localization process consists of measuring distances, and calculating the target position based on these measurements. Thus, the errors in distance measurements reflect on the localization error. The extent to which the errors in distance measurements affect the localization error depends on the position of anchors with respect to target, or in other words geometry. Geometric Dilution of Precision (GDoP) is a measure that expresses how much the errors in measured data, in our case distances, affect the localization error: $GDoP = \frac{\Delta(output\ location)}{\Delta(measured\ distances)}$, where $output\ location$ is the calculated target location and Δ is any useful measure of localization error, typically the error standard

deviation. GDoP has been used to express precision of satellite based systems (Yarlagadda et al. [2000]) robot localization systems (Kelly [2003]), as well as for mobile GSM device localization by using base stations (Spirito [2001]).

GDoP is the most popular metric used today for GNSS systems to express the effect of geometric configuration of satellites on precision of localization. It is common for modern GNSS receivers to show GDoP values as information for the user. Unlike the system we consider in this work, which uses TWToF, GNSS systems involve pseudo-ranges exclusively and include time offset error estimation. GNSSs use time of flight, but while satellites have highly accurate clocks, the target clock is not very accurate. Therefore, apart from the distances from satellite to target, there is one more unknown, which is the time offset. It can be shown that GDoP for satellite systems is approximately inversely proportional to volume of the tetrahedron formed by connecting the endpoints of unit vectors directed from target to four satellites. As we will show, this does not hold for 3D TWToF systems. For more detail on GDoP used in GNSS systems see Yarlagadda et al. [2000].

Much valuable work on the topic of metrics expressing the effect of geometry on system performance has been done for geolocation systems predating GNSS. Lee [1975] is one of the first to address error estimation for multilateration systems using WLS and provide the basis for the future results in the area.

In Spirito [2001] GDoP for a multilateration system is derived and its geometric interpretation is given. If the estimated target position is $\hat{\boldsymbol{x}} = [\hat{x}\ \hat{y}\ ]^T$ then the accuracy can be expressed by the covariance matrix $Cov(\hat{\boldsymbol{x}})$. The diagonal entries of this matrix are the variances of the location error along x and y axes. The accuracy measure considered is the square root of the sum of these variances, the square root of the trace of the covariance matrix $\sqrt{Tr\{Cov(\hat{\boldsymbol{x}})\}}$. The value of GDoP obtained is:

$$GDoP = \frac{\sigma_0}{\sigma_D} = \sqrt{\frac{N}{\sum_{i,j} \sin^2 \alpha_{ij}}}, \tag{2.7}$$

where N is the number of anchors used to localize the target, and α_{ij} are angles formed by each pair of anchors and the target. GDoP depends on sines of bearing angles, which are proportional to surfaces of parallelograms formed by each pair of anchors with the target. However, the result is only valid for 2D systems. As part of this work, we derive an expression for GDoP for 3D multilateration systems.

Chapter 3

Problem Formulation

In this chapter we give our problem formulation. The work is divided in three parts: one dimensional, two dimensional and three dimensional localization.

3.1 One Dimensional Localization

The goal of the part of my thesis focusing of one dimensional localization is to design and test a working prototype for in-tunnel vehicle localization based on sensors placed along the tunnel. The prototype should enable localization of maximum 10 m precision, as required by Swiss regulations.

3.2 Two and Three Dimensional Localization

The second and the third part of my thesis consist of optimizing node placement for ToF-based localization, in rooms of a given shape, taking into consideration the obstacles. We describe our localization system, list our main assumptions, explain how the model and the assumptions relate to the real-life scenario, give the formal problem formulation and describe the error model we use for localization.

We assume that positions where anchor nodes can be placed are given. We call these positions *candidate positions*. We assume that the positions of obstacles are also known. See Figure 3.1(a) for 2D localization. Two nodes can communicate if the distance between them is lower than the known node range, and there are no obstacles between them. Our goal is to select the minimized subset of candidate positions, such that, if we place the anchor nodes at these positions, localization uncertainty will be below the given threshold in the entire area of

interest. We discretize the area of interest, which is the area that needs to be monitored, to obtain the set of *target positions*. Localization quality at each target position depends on the configuration of nearby anchors, and is expressed by the uncertainty function. We are optimizing for the UWB-based localization system using TWToF and multilateration.

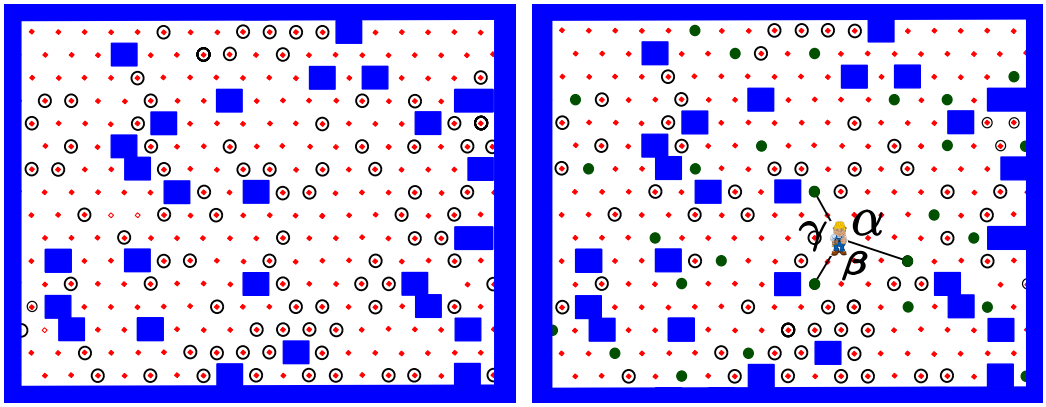


Figure 3.1. A problem instance (left) and a solution (right) for 2D placement for localization. Obstacles are represented as large blue squares, the candidate positions as black circles and possible target positions are marked as small red diamonds. The target is indicated by a human figure and the positions where anchors should be placed by green filled dots. Bearing angles are α, β and γ.

3.2.1 Localization System Description

Here, we briefly describe a localization system that can benefit from our optimization of anchor placement.

The target estimates its distance to each of the anchors it can communicate with, sequentially. The target can communicate with an anchor if they are closer than the node range and there are no obstacles that obstruct communication between them. Anchors communicate their identification and coordinates to the target. As three anchors are the minimum required for 2D trilateration, our placement approaches will guarantee that there are always at least three anchors from which the distance can be measured (four in 3D). The target does not have to know which three (or four) anchors are 'assigned' to its current position by the optimization algorithm, but can use all anchors that are available.

It may happen that the distance measurements to some of the anchors are affected by non line of sight condition. NLoS significantly reduces ranging accuracy

of UWB measurements. Therefore, NLoS measurements may have to be detected and discarded. Multiple methods to detect and discard NLoS measurements can be found in literature. See Section 2.3 for a short overview. For example, since the map of the area is available, knowledge of the anchors positions, the obstacles positions, and the previous position of the moving target can be used to predict the loss of the line of sight from one of the anchors. Detecting measurements affected by NLoS is out of the scope of this work. The task of the optimization algorithm is to ensure that there are at least three (or four, in 3D) measurements that are not affected by NLoS at each possible target position.

The distances from the target to the anchors are measured by using TWToF. The target position is determined by using the WLS method. The target can send the calculated position data to a central server either directly, or via the network of anchors, depending on the application requirements. Also, depending on the requirements, the data can be sent periodically, as often as the application requires, or only in case of an emergency.

3.2.2 Assumptions

We list the main assumptions of our model and explain how they relate to real-life applications. Our assumptions are as follows:

- The positions of obstacles are given and fixed. This assumption is realistic because, as UWB is resistant to multipath propagation, smaller obstacles should not affect the signal. Typically, we only have to consider large obstacles to wireless propagation such as walls, pillars and large machines.

- We assume that the positions where anchor nodes can be placed, candidate positions, are given. The scenario where the number of positions is limited corresponds to real-life, because UWB nodes may require power plugs, as well as support structures such as walls. If the anchor nodes can only be placed at a set of points, those points are our candidate positions. If the anchor nodes can be placed anywhere in the given area, this area will be discretized to obtain the candidate positions.

- The node range is known, equal for all nodes and uniform in all directions. To obtain the distance measurements between two nodes, their distance has to be smaller than the node range.

- The distance measurement error can be modeled as Gaussian distribution of mean 0 and known standard deviation σ_D if there are no obstacles between the target and the anchor and the distance between them is lower

than the node range. Otherwise, the ranging error is modeled as infinite. For the explanation of why this model is realistic for an UWB system, see Section 3.2.4.

3.2.3 Placement Optimization for Localization Problem Formulation

We address anchor nodes placement optimization for UWB trilateration-based localization and the fault-tolerant version of the same problem. We call the first problem Placement Optimization for Localization (POL) for short, and the second problem, Fault-tolerant Placement Optimization for Localization (FPOL). We first define the POL problem formally, and then we extend the definition to FPOL.

We are given a set T, consisting of all possible target positions, and a set C, consisting of all positions where an anchor can be placed, candidate positions. For 2D version of the problem, we are also given a function $U : C \times C \times C \times T \rightarrow \mathbb{R}$, such that $U(a_i, a_j, a_k, t_u)$ returns the uncertainty of localizing the target at position t_u by using anchors at positions a_i, a_j, a_k. [1] For the 3D version, we are given a function $U : C \times C \times C \times C \times T \rightarrow \mathbb{R}$, such that $U(a_i, a_j, a_k, a_l, t_u)$ returns the uncertainty of localizing the target at position t_u by using anchors at positions a_i, a_j, a_k, a_l. Finally, we are given an uncertainty threshold U^* . Let $P = \{a_1...a_n\} \in C^n$ be a placement of anchors. An uncertainty of localizing a target at position t_u, is given as:

$$U(P, t_u) = \min_{a_1..a_N \in P} U(a_1..a_N, t_u), \tag{3.1}$$

where $N = 3$ for the 2D problem and $N = 4$ for the 3D problem. The target at position t_u is localized by choosing the best triple/quadruple of anchors for that target position. A set P is a solution to the POL problem, if uncertainty for every target position in T is less than U^*:

$$\forall t_u \in T, \quad U(P, t_u) < U^*. \tag{3.2}$$

Optimal solution to the POL problem is such set P of minimum cardinality.

A set P is a solution to the FPOL problem, if after removing any one element from the set P, uncertainty for every target position in T is less than U^*:

$$\forall t_u \in T, \quad \forall a_i \in P, \quad U(P \setminus \{a_i\}, t_u) < U^*. \tag{3.3}$$

Optimal solution to the FPOL problem is such set P of minimum cardinality.

[1] The uncertainty function U can be any measure that reflects the localization error, such as its standard deviation.

3.2.4 Error Model for UWB Multilateration-based Localization

Both ranging, or determining distances from anchors to target, and multilateration, which combines those distances to determine the target position, introduce an error into the localization process.

Ranging Error

Ranging error of UWB systems has been studied in Alavi and Pahlavan [2006]. The error is modeled as a Gaussian random variable, with parameters depending on presence of LoS. In presence of LoS, the mean error value is 0, while standard deviation is low, and increases with the distance d between transmitter and receiver as $\sigma = c \cdot log(1 + d)$. In absence of LoS, error standard deviation is much higher, while its mean value is positive. UWB nodes are only able to communicate up to a certain distance we call node range.

Our ranging error model is an approximation of this model. As the error standard deviation in the presence of LoS has been shown to increase slowly with distance, we model the ranging error as Gaussian distribution of mean 0 and known standard deviation σ_D if LoS is present and distance between nodes is lower than the node range. Otherwise, if there is no LoS, or the distance between nodes is higher than the node range, the ranging error is infinite. The value of σ_D depends on the hardware used and the environment, and can be determined experimentally, as in Alavi and Pahlavan [2006].

In our experimental study on in-tunnel localization, we show that modeling the time of flight distance measurement error as Gaussian distribution matches the reality well for RF ToF systems (see Section 5.7.1). We cannot however directly use the experimental results from our in-tunnel study for our design of 2D and 3D localization systems, as UWB systems have much higher accuracy than RF ToF-based systems, and also properties of wireless propagation in tunnels are significantly different than in other indoor environments. Therefore, in this part of our work, we consider σ_D as known.

The information on whether a target can see an anchor is used in range-free localization systems. See our discussion on range-free localization in Chapter 2. However, precision of such localization systems is by far inferior to UWB systems. Therefore, the information that a target cannot detect an anchor, although it does eliminate the area covered by the anchor as a likely target position, has little value compared to the measured distance between a target and an anchor and we model the ranging error as infinite if the target cannot detect an anchor.

Trilateration Error in Two-dimensional Space

As discussed in Section 2.4, in 2D space, each possible target position needs to be monitored by at least three anchor nodes. The three anchors that monitor a target must not be collinear. For precision of trilateration, the relative placement with respect to a target of anchor nodes that are used to localize it is particularly important. Geometric Dilution of Precision (GDoP) is a measure that uses the angles, called bearing angles (see Figure 3.1), formed by the anchors and the target, to quantify the effect that an error in measured distances will have on the error of the position estimation (Spirito [2001]). GDoP for trilateration in 2D space is given as:

$$GDoP = \frac{\sigma_0}{\sigma_D} = \sqrt{\frac{3}{\sin^2\alpha + \sin^2\beta + \sin^2\gamma}}, \tag{3.4}$$

where α, β and γ are the bearing angles between the anchors and the target position, and σ_D and σ_0 are standard deviations of distance measurement error and localization error respectively.

Uncertainty Function

The uncertainty function can be any measure that reflects the localization error. For uncertainty function $U(a_1...a_N, t_u)$, we choose the standard deviation of the localization error σ_0. Given anchors, $a_1...a_N$, and a target position t_u, we define the localization uncertainty function as:

$$U(a_1...a_N, t_u) = GDoP \cdot \sigma_D, \tag{3.5}$$

if the distance from each of the anchors to the target is lower than the node range, anchors are not collinear (not coplanar in 3D), and there is LoS between each of the anchors and the target. Otherwise:

$$U(a_1...a_N, t_u) = \infty. \tag{3.6}$$

This definition relies on the assumptions listed in Section 3.2.2, that nodes can only communicate if the distance between them is lower than the node range, and the standard deviation of distance measurement error is known and constant. This uncertainty function can be interpreted as follows. To guarantee that a target at position t_u can be localized with uncertainty lower than U^*, we need the target position to be monitored by three non-collinear anchors (four non-coplanar anchors in 3D), in line of sight, not further from it than

the node range, with GDoP formed by those anchors and the target position $GDoP < GDoP^* = \frac{U^*}{\sigma_D}$.

Chapter 4

State of the Art - Placement Problems in Wireless Networks

Optimal node placement depends on the network functionality, goals and constraints it has to meet. Therefore, the research field of network node placement includes a wide range of problems as well as different approaches to these problems. In this chapter we review node placement problems and approaches in literature, with the focus on placement for localization networks.

It is difficult to compare the different approaches to node placement problems quantitatively, as different authors address different optimization problems, and even a small change in problem formulation may significantly influence the optimization problem complexity. Moreover, many authors do not provide comparison of different approaches, algorithm execution times or experimental comparison with exact solutions. However, to the extent to which it is possible, we strive to provide quantitative results and the comparison of different approaches in our literature review.

We review the related work by using several different taxonomies: by different approaches to addressing the problem, by problem dimension, from 1D to 3D, and by network function. The most common goals the networks have to meet are coverage and connectivity, as well as their extensions k-coverage and k-connectivity, that provide fault tolerance, where the network will still satisfy the coverage or connectivity condition after $k-1$ nodes are removed. Finally, we focus on placement for localization networks.

4.1 Placement Optimization Approaches

As node placement problems are NP hard as a rule, different approaches to optimizing node placement can be found in literature. For us, exact solutions, as well as different heuristics and metaheuristics are particularly interesting, as our approach is based on exact solutions and heuristics. We also review regular geometric patterns, as well as approximations.

4.1.1 Exact Approaches

Not many works focus on exact approaches for optimizing node placement as these problems are NP hard as a rule. Most works focus on heuristics, approximations, as well as addressing simplified versions of problems. Some papers give the ILP formulation but do not test it by experimental simulation as the time would be prohibitive. However, as computing power is increasing, potential for finding exact solutions to problem instances of larger sizes increases. In addition to finding exact solutions to problem instances of reasonable sizes, exact approaches are useful to have for comparison in order to evaluate non-exact approaches.

Integer Linear Programs (ILP) can be used for finding exact solutions to reasonably sized problem instances of NP-hard problems in manageable computing times. ILP formulation consists of a system of linear equations, where variables represent the solution to the problem instance. A good problem formulation is very important in order to shorten computing times.

Point coverage problem is a problem of placing a minimum set of sensor nodes, so that given points of interest are covered. Covering a plane can be reduced to a point coverage problem by discretizing the plane, so that all points on the grid need to be covered. This is known as grid coverage. For each position where a node can be placed, we know which points of interest the node is going to cover if placed there. Therefore, coverage problems can often be reduced to different versions of classic set cover problem. Chakrabarty et al. [2002] provide an ILP formulation for fault-tolerant coverage of bounded regular 2D and 3D grids and tests the ILP on a small example. Xu and Sahni [2007] propose an ILP formulation for the same problem that reduces the number of ILP variables and constraints, thus improving the ILP formulation.

The problem of placing additional wireless nodes to achieve connectivity between existing nodes can be reduced to Steiner tree problem on graphs. [1] Polzin

[1] Given an undirected graph with non-negative edge weights and a subset of vertices, referred

[2003] addresses exact solutions for instances of Steiner Tree problem. Connectivity problem can be expressed as multicomodity flow packing Plotkin et al. [1991]. Multicomodity flow packing can be used for formulating 2-connectivity placement problem as an ILP as shown in Misra et al. [2010] The authors do not provide results to quantify the execution times. In Amaldi et al. [2012] an ILP solution for wireless sensors placement to detect mobile targets traversing a given area is proposed. Equations based on multicomodity flow packing are used to add connectivity properties to detection solution.

Kirchhof [2013] has suggested ILP formulations for 2D placement for both triangulation and trilateration, but with only two nodes. Tekdas and Isler [2010] proposes an ILP formulation for triangulation problem, tests it on one problem instance and reports the solution was not reached after 35 CPU hours.

4.1.2 Greedy Heuristics

Heuristics are often used to address NP-hard problems. The goal of a heuristic is to produce a solution that is close to the optimal in short computing time. Greedy heuristics are based on selecting the best local solution at each step. For example, if a number of points, on a grid or not, needs to be covered, the most straightforward greedy solution would consist of placing a sensor at each step, that covers the largest number of points that are not covered so far, until all points are covered. Xu and Sahni [2007] propose a greedy approach for fault-tolerant point coverage with different types of nodes and compare it with the approximation algorithms they propose for the same problem. Zou and Chakrabarty [2004] and Dhillon and Chakrabarty [2003] propose greedy heuristics to address sensor placement in a probabilistic optimization framework. Sensors are thrown from an airplane or carried by water, so that there is an uncertainty in actual sensor positions. Intended sensor position serves as a mean of a Gaussian distribution for actual sensor position. The algorithms are targeted at maximizing average coverage or coverage of the most vulnerable regions. The authors report that their solutions outperform both random and uniform placement, when testing on several case studies with obstacles and preferential coverage.

Howard et al. [2002] addresses mobile sensor networks, where nodes, apart from sensing abilities, also have locomotion abilities and can deploy themselves based on the information they get from previously deployed nodes. The algorithm is designed to maximize coverage while ensuring that line of sight is present

to as terminals, the Steiner tree problem in graphs requires a tree of minimum weight that contains all terminals and may include additional vertices.

between the nodes, so the nodes can use other nodes to localize themselves. The later constraint is necessary so that the nodes can localize themselves by using other nodes as landmarks. The authors propose a greedy heuristic to deploy nodes incrementally, one at a time. The algorithm is model free, meaning that the floor plan of the area where the network is deployed is not needed. The authors report that their model performs nearly as well as a map-based model, with coverage value between 70 % and 85 % that of a map-based model.

4.1.3 Metaheuristics

A metaheuristic is a high-level procedure designed to generate a search algorithm for an optimization problem. Metaheuristics use different techniques in order to avoid local optima, and try to find the global optimum solution. For that purpose they may accept temporary deterioration of a solution. Most literature on metaheuristics is experimental in nature. Metaheuristic approaches are frequently used to tackle all kinds of optimization problems, including placement for wireless networks. We describe some of the most frequently used metaheuristics and their application to placement for wireless networks.

Genetic algorithms are inspired by the process of natural selection. Each generation contains a set of candidate solutions. A subset of these solutions is selected, based on fitness scores assigned to each candidate solution, to breed the next generation by crossover. Crossover combines two of the solutions in order to obtain new solutions. After crossover, random changes, mutations, are applied to some of the solutions. The goal of Wu et al. [2007] is to optimize coverage within the given fixed cost. They use probabilistic sensing model, where the probability of the target being detected by the sensor decreases as the target gets further from the sensor. Sensors of several types, with different detection probability density function are considered. The authors prove that the sensor deployment problem is NP-complete. Crossover, in this example is the operation of exchanging two square regions in two parent solutions. Mutation consists of adding or deleting sensors, moving them to another place, or replacing by a different type of sensors. The authors show by simulation experiments that their algorithm outperforms the greedy heuristic. Qu and Georgakopoulos [2011]) address the scenario where nodes are mobile. They use a genetic algorithm for node relocation, to provide trade-off between coverage and traveled distance.

Ant colony metaheuristics are inspired by ants behavior. When ants search for food, they leave pheromones where they have passed, so that other ants will know where they can find food. (Li et al. [2010]) propose an ant colony algorithm for ensuring k-coverage of critical points and connectivity, where nodes can

be placed at given candidate points. Obstacles are considered. Number of ants and the total number of iterations are parameters of the algorithm. Ants move and place sensor nodes at candidate points. At each step, an ant moves from current point to the candidate, depending on the pheromone value on the link and giving advantage to those candidate positions that can cover critical points that are not covered thus far. They show their algorithm performs better than MAX-MIN ant system (Stutzle and Hoos [1997]) and validate it in a commercial sensor network system.

Amaldi et al. [2012] use tabu search to position wireless sensors to detect mobile targets traversing a given area by using the concept of path exposure as a measure of detection quality. Set of candidate positions where the sensors can be placed are given. They address two problems. Maximizing exposure of the least exposed path given maximum cost, and minimizing cost giving minimum exposure of the least exposed path. Tabu search works as follows. Starting from an initial feasible solution, a set of neighboring solutions are explored by applying moves to the solution. Moves consist of adding or removing some sensors. Best neighbor solution is selected as the next iterate. In order to prevent cycling (returning to feasible solutions that have already been generated) a list of tabu moves is maintained. Tabu list is implemented as follows: sensors that are installed (deleted) cannot be deleted (installed) during next L iterations. They show by simulation experiments that, for the instances they tested, tabu search resulted in solutions that are very close to optimal.

Simulated annealing is motivated by a physical process of annealing which finds the low energy state of a metal by heating it and then cooling slowly. Temperature is the controlling variable of the process which slowly decreases and determines how random the energy state is. Desired properties of the system are encoded as energy. Moves are generated randomly, and it is randomly determined whether they will be accepted, depending on decrease in energy they introduce, as well as the temperature. At the beginning, while the temperature is high, moves will be accepted even if they increase the energy significantly. As temperature gets lower, moves that increase the energy are less likely to be accepted. This approach is used for placement optimization in Lin and Chiu [2005]. Two points in a field can be discriminated if they are covered by different combinations of sensors. Within a limited budget, the goal is to minimize maximum distance between points that cannot be discriminated. The authors compare their algorithm against exhaustive search on a field of very small dimensions and show that in each of the experiments they performed, the algorithm reached the optimal solution. On a bigger field, they compare their algorithm against random placement, and show their algorithm performs much better.

In each iteration of greedy randomized adaptive search procedure (GRASP) a greedy randomized feasible solution is found, which is later improved by searching in the neighborhood of the solution. The best overall solution is kept as the result. The greedy stage is randomized, because one of the best local solutions is selected, but not necessarily the best one. Sitanayah et al. [2011] proposed to use a modification of GRASP algorithm for k-connectivity.

4.1.4 Regular Geometric Patterns

Under idealized conditions, where node placement problems on the plane with no obstacles are addressed, it is often possible to find regular geometric patterns that provide full coverage or connectivity. The problem of placement for coverage is related to the covering problem in computational geometry, covering points on a plane using a minimum number of given geometric bodies. Kershner [1939] has shown that the regular triangle lattice pattern is optimal in terms of the numbers of discs needed to achieve full coverage. This result naturally provides 6-connectivity when $R_c \geq \sqrt{3}R_s$, where R_c and R_s are communication and sensing range respectively. Bai et al. [2006], Bai et al. [2008], Yun et al. [2010] complete the set of deployment patterns to achieve full coverage and k-connectivity for all $k \leq 6$ and all different R_c/R_s ratios.

Alam and Haas [2006] found that the placement of sensors in the middle of truncated octahedron cells, which are created by Vornoi tessellation of 3D space is the best way to cover the space so that the number of nodes is minimized. They also derive the minimum ratio of communication and sensing range required for such network to be connected. Zhang et al. [2010] address the problem of finding optimal patterns for full coverage and k-connectivity for $k \leq 4$ in three dimensions.

These formulations do not consider obstacles or constraints in sensor deployment, which are especially important in indoor environments.

4.1.5 Approximations

Connectivity problems are usually presented in the form where some nodes, needed to meet the network functionality, are already placed and additional nodes need to be placed for the network to be connected, or k-connected, to achieve fault tolerance. The problem of connectivity has been addressed by approximations in Hao et al. [2004], Liu et al. [2005], Misra et al. [2010], Bredin et al. [2010], Pu et al. [2009]. Many of these results rely on classic results from

graph theory. Misra et al. [2010] use survivable network design problem (Fleischer [2001]). They obtain approximation factor of 5.5 for simple connectivity and 7 for 2-connectivity. Bredin et al. [2010] use minimum-weight k-vertex connected subgraph (Cheriyan et al. [2002]). They address the k-connectivity problem for any k and obtain the approximation factor of $O(k^4 \log k)$. None of these works provides experimental comparison with other approaches or with exact solutions.

Wang and Zhong [2006] propose an approximation algorithm for fault-tolerant point coverage, for nodes of different ranges and different costs. The approximation factor is $\sum_{i=1}^{l} k_i - \sigma + 1$, where k_i denotes the maximum number of points that can be covered by the sensor of $i-th$ type and each point has to be covered by at least σ nodes. The authors provide numerical results that show that in simulation experiments they perform, the cost is at most three times the optimum. Xu and Sahni [2007] propose approximation algorithms for fault-tolerant point coverage.

Placement for localization has also been addressed by approximations. Tekdas and Isler [2010] and Efrat et al. [2005] address placement for triangulation problem with no visual occlusions. Both approaches are based on GDoP. Tekdas and Isler [2010], propose an approximation algorithm that places at most $3n_{opt}$ sensors that observe each target position with at most 5.5 $GDoP^*$, where n_{opt} is the optimal number of nodes that observe each target position with at most $GDoP^*$. Efrat et al. [2005] propose an approximation algorithm that places at most $O(n_{opt} \log n_{opt})$ sensors that observe each target position at the angle at least $\alpha/2$, where n_{opt} is the optimal number of nodes that observe each target position at angle at least α.

4.2 Problem Dimension

One dimensional placement problems are rarely addressed. The most notable case of one-dimensional structures are tunnels. Liu et al. [2010] address node placement to provide 2-tier network connectivity for in-tunnel networks. Relay nodes need to be added to connect the sensor nodes that are already placed. They use the greedy heuristic for the minimum set cover problem to choose cluster heads among these sensor nodes, so that each sensor node has a 1-hop connection to a cluster head. Then they use the search-and-find heuristic to add more nodes and make all the relay nodes connected. They test the solution by simulation. On the other hand, in our 1D placement study presented in Chapter 5, we use a real tunnel to derive empiric conclusions on how the nodes should

be placed and test our solution.

Most wireless terrestrial networks operate in 3D. However, their deployment is usually based on 2D design. This is justified in many cases, as usually, the width and length of the areas where such networks are deployed are much larger than differences in height, therefore, the resulting errors of such approximation are not large. In this section, we provide an overview of node placement problems focusing on 3D scenarios. A good overview of coverage and connectivity in wireless sensor networks is given in Mansoor and Ammari [2014]. Three-dimensional networks are both more difficult to visualize and their analysis is much more computationally intensive compared to 2D networks.

Three-dimensional placement problems naturally appear in underwater networks, so it has been these environments that have motivated most research in the area. Unlike in indoor spaces, in underwater networks there is more freedom as to where the nodes can be placed, therefore, scenarios with no obstacles and no constraints for placing anchors correspond well to reality. Atmospheric and space communications are other such examples. Chakrabarty et al. [2002] address k-coverage as well as a localization problem on a 3D grid, where every grid point needs to be uniquely identifiable by the set of sensors that cover it. Wang and Zhong [2006] address k-coverage in 3D by using an approximation algorithm. They provide numerical results showing that actual approximation ratios in their examples are never bigger than 3. Ammari and Das [2010] address the sensor density necessary for k-coverage and connectivity of 3D space with no obstacles. They prove that if every Reuleaux tetrahedron in such space contains k nodes, the space will be k-covered. They also prove that connectivity in such space is $9.926(R/r_0)^3 * k$, where R is the communication range, r is the sensing range and $r_0 = r/1.066$. Liu et al. [2017] consider relay node placement and the flow allocation as a joint problem in underwater acoustic sensor networks (UASN). Relay node placement is a problem that appears frequently in node placement for WSNs. Sensor nodes are already placed in a way that is determined by the application, and additional nodes, relay nodes, can be placed to achieve certain network properties, in this case, to minimize power consumption, which increases with the distance between nodes. Relay nodes have fixed x and y coordinates and only z coordinate can change. The problem is formulated into a nonlinear integer programming problem, and a heuristic scheme is proposed. Alam and Haas [2006], Zhang et al. [2010], Ammari [2017] all focus on optimal regular patterns for coverage and connectivity of 3D space (see Section 4.1.4).

Another kind of environments where 2D approximations are often not sufficient, are outdoor environments with significant changes of altitude, such as

mountains or volcanoes. As it is necessary to monitor volcanoes to detect their activity, significant research has been dedicated to these environments. They are modeled as a complex surface in 3D space where sensors can be only deployed on the surface. Liu and Ma [2012] derive the expected coverage ratios under stochastic sensor deployment for rolling terrain (mountainous regions). Jin et al. [2012] studied the sensor deployment on complex surfaces embedded in 3D space with minimized overall sensing unreliability, given the number of sensors. Overall unreliability is defined as the integral unreliability over the entire area to be covered. They designed a number of algorithms for practical implementation. Kong et al. [2014] address two problems on such surface. First, given the number of nodes and stochastic deployment, they find the expected coverage ratio. Second, if sensor deployment can be planned, they look at a strategy to adopt to guarantee full coverage with the least number of sensors. For the second problem, the authors discretize the surface and thus reduce the coverage problem to classic set cover problem. Since the problem is NP-hard, they also propose approximation algorithms.

4.3 Placement Problem Depending on Network Function

Wireless networks have different functionalities and have to meet different goals. The optimal placement will depend on the requirements the network has to meet, the environment, as well as the type of nodes available. In this section we review the literature on node placement depending on network function. The most basic and the most frequently addressed requirement in the literature is coverage, where network nodes have to cover a given area or a set of points in order to sense some parameters in their environment. Connectivity is the second most frequently analyzed requirement and we will give a short overview of works that focus on node placement for connectivity. Finally, we focus on node placement for localization networks.

4.3.1 Coverage

Networks consisting of multiple nodes with processing and communication abilities that are equipped with different kinds of sensors are known as wireless sensor networks (WSN). The main goal of many WSNs is to monitor some parameters in the environment. For that, covering the area of interest is necessary. Coverage is one of the most important performance metrics in wireless sensor networks

and reflects how well the area of interest is monitored. Coverage problems were among the first ones to be explored. A survey on coverage problems can be found in Wang [2011].

Network nodes can be deployed deterministically or randomly. With a random deployment, many nodes are scattered, for example from an airplane, and we do not know the exact positions where they are going to land. Nodes are deployed randomly when the environment is hostile and it is not possible to reach every point deterministically. In that case, the problem that is often explored is a number of nodes that should be deployed to achieve certain network properties with given probability. Such a problem is studied for example by Liu and Ma [2012] and Kong et al. [2014], for 3D coverage (see Section 4.2). Zou and Chakrabarty [2004] and Dhillon and Chakrabarty [2003] propose greedy algorithms for determining sensor placement, where exact sensor locations are not known, but intended coordinates serve as mean values for their positions (see Section 4.1.2).

With deterministic deployment, we have to decide where exactly the nodes should be placed among all available candidate locations. Deterministic deployments are applicable when the number of nodes is not too large and in a friendly environment, where every location can be reached. Given the points of interest to be covered, and given the points where sensors can be placed, placement for coverage problem can be seen as a set cover problem, as placing each sensor covers a set of points of interest. The problem of covering an area can be related to the problem of covering a set of points by discretizing the area to obtain a grid. In Ke et al. [2007] the authors prove that the problem of deploying a minimum number of sensors on a grid vertices to fully cover a given set of critical grids is NP-complete.

To provide fault-tolerant coverage, some works require each point to be covered by at least k nodes, so that the network can continue functioning after $k-1$ nodes fail. Huang and Tseng [2005] address the decision version of this problem. Given the sensor positions and ranges, they show that the problem of deciding whether each point on a plane is covered by at least k nodes, can be solved in polynomial time. Chakrabarty et al. [2002], Wang and Zhong [2006] and Xu and Sahni [2007] address generalization of placement for k-coverage, where different types of nodes are available, with different ranges and costs. Chakrabarty et al. [2002] propose an Integer Linear Programming (ILP) formulation for fault-tolerant coverage of bounded regular 2D and 3D grids. They tests the ILP on a small example. For problem instances of larger sizes they propose a divide and conquer approach: tiling the grid by using optimal solutions for smaller subgrids. Wang and Zhong [2006] propose an ILP formulation and an approximation al-

gorithm (see Section 4.1.5) for grid coverage where both target locations and candidate locations where sensors can be placed are on a regular grid. Xu and Sahni [2007] propose an ILP formulation and a greedy algorithm for a generalization where the set of locations to be monitored and the set of points where it is feasible to place sensors do not have to be on a regular grid, as well as several approximation algorithms for grid coverage. Like fault-tolerant coverage, localization systems also require each point of interest to be covered by several nodes. However, the difference is that in fault-tolerant coverage each node can estimate the target state independently, while localization requires several nodes to jointly estimate the target state, which makes the placement for localization problem more complex.

4.3.2 Connectivity

Connectivity problems require placing additional nodes in an existing network, so that there are k node-disjoint paths between each pair of nodes. These problems have been extensively researched, and we include them here for completeness. Node placement problems to achieve network connectivity can be reduced to connectivity problems in graph theory. Therefore, results from graph theory research are often used to address this class of problems.

As shown in Misra et al. [2010], ILP formulation for 2-connectivity problem can be found based on multicomodity flow packing Plotkin et al. [1991].

The problem of multiple-connectivity has been addressed by approximations in Hao et al. [2004], Liu et al. [2005], Misra et al. [2010], Bredin et al. [2010], Pu et al. [2009]. We gave an overview of these results in Section 4.1.5. In Almasaeid and Kamal [2009] a greedy placement algorithm is used to show how the approach of Bredin et al. [2010] can be improved by giving preference to candidate nodes in the intersection of communication ranges of other nodes. Sitanayah et al. [2011] proposed a GRASP-ARP algorithm, a local search algorithm based on Greedy Randomized Adaptive Search Procedure (GRASP) algorithm for k-connectivity and compared it to the approximation algorithm proposed in Pu et al. [2009].

4.3.3 Localization

Node placement problems for different localization methods have also been addressed in literature. We review them in this section.

Cramer Rao Lower Bound (CRLB) is the lower bound on the variance of an unbiased estimator. It is frequently used to compare different estimators, where

the estimator with the lowest CRLB is best. CRLB is defined as the inverse of the Fisher Information Matrix (FIM) (see Kay [1993]). To minimize localization uncertainty, determinant of FIM should be maximized. The method of minimizing CRLB is used by many authors to determine optimal placement of a predetermined number of sensors around the target on a plane or in 3D space (Glotzbach et al. [2013], Yang [2007]) or in well-defined regular regions with no obstacles (Zhou et al. [2010], Perez-Ramirez et al. [2013]). Best practices and guidelines for node placement can also be obtained by using CRLB (Sheng and Hu [2005], Ash and Moses [2008]).

Sheng and Hu [2005] address localization by acoustic signal energy measurements. They use the CRLB to formally confirm the conventional wisdom that to improve localization, anchors should be placed uniformly and densely. Ash and Moses [2008] use CRLB to show analytically that, for optimal localization, anchors should be placed uniformly around the perimeter of the area of interest.

Optimal Anchor-Target Geometries

Several works address optimum anchor-target geometries for 2D and 3D, for different localization methods and different numbers of nodes. Yang and Scheuing [2005]; Yang [2007] use CRLB to find optimum sensor positioning relative to the target for both 2D and 3D on a surface/space with no obstacles for different numbers of sensors, for TDoA. The results represent, intuitively, a uniform placement on a circle/sphere. In 2D case this implies regular angular separation. For 3D there are only five regular solids offering perfect symmetry, so called Platonic solids: for 4, 6, 8, 12 and 20 vertices. Platonic solids as well as their centered superpositions are CRB optimal. For odd number of nodes, the authors assume there are no CRB optimal solutions in 3D case and propose to use patterns that maximize the minimum distance between any pair of sensors, the problem known as spherical codes problem. Bishop et al. [2010] use CRLB to derive optimal anchor target geometries for range-only, bearing-only and time of arrival based localization, for any number of anchors in 2D. They give necessary and sufficient conditions on the anchor-target angular positions which if satisfied, minimize localization uncertainty and show that in general, these configurations are not unique.

Zhao et al. [2013] unify formulation for optimal anchor-target configurations for range-only, bearing-only and RSSI sensors. They prove necessary and sufficient conditions for optimal geometry in 2D and 3D, based on the given estimated target position and a given number of sensors. propose a control strategy to address 3D placement for range only sensors. They minimize the objective

function which consists of two terms: inter-sensor potential and external potential. Inter-sensor potential is the objective function for optimal sensor placement and sensor trajectory constraints are described as external potential. The formulation can be adapted for a moving target, assuming sensors can also move and follow the target in order to position themselves optimally. Unlike these works that address the local placement problem with the limited number of nodes, our work addresses the global placement problem in the entire area of interest.

Well-Defined Regular Regions

A number of works addresses localization in well-defined regular regions, that is, in indoor spaces of rectangular shape, where there are no obstacles to wireless propagation. These works also assume that the node range is higher than room dimensions, therefore few nodes are sufficient to cover the entire area of interest.

Chen et al. [2006] use nonlinear least squares (minimizing the sum of the error squared) to express analytically and experimentally test the optimal placement of anchors in well-defined regular regions for multilateration systems. They obtain deployment patterns for three to eight nodes, that consist of nested triangles and squares and should scale depending on the size of the room. They do not consider obstacles and assume the room dimensions are smaller than the node range. Zhou et al. [2010] address the scenario where only four anchors are available to be placed in a rectangular facility and optimize their positions. They show analytically that all rectangular landmark layouts have the same minimum CRLB. They use Monte Carlo simulation to show that the optimal layout in terms of the expected CRLB changes with aspect ratio of the facility geometry, though the optimal positions are always at the edge of the area. Perez-Ramirez et al. [2013] also focus on indoor localization. Optimized positions of sensors in 3D space are obtained given their number, assuming the target position is known. The authors address optimal placement of anchors in indoor facilities, where the cubic room has no obstacles, and the given number of anchors can be placed at the walls and ceiling. The optimization relies on the fisher information matrix.

All of the works listed so far in this section study sensor positioning relative to the target, rather than optimizing anchor placement to cover large areas where the target can move.

Range-free Localization

As a simpler alternative to distance-based and angle-based localization, where distances and angles are measured, the target position may be determined based

only on knowing which anchors can detect the target. Then we know that the target is in the intersection of areas these anchors cover. This localization approach is less precise, but sufficient for many applications where high localization precision is not necessary, such as location-based advertisements. It is cheaper as it does not require dedicated hardware. Off the shelf RF, WiFi or bluetooth sensors can be used instead. Chakrabarty et al. [2002] show that for regular grids in 2D and 3D, coding theory framework can be used to determine sensor placement so that every point on the grid is covered by the unique combination of sensors and the target position on every grid point can be uniquely identified. Lin and Chiu [2005] use simulated annealing metaheuristic (see Section 4.1.3) to address the same problem and extend it to irregular sensor fields. They show by comparing their solution with exhaustive search that their algorithm can find optimal solutions for the small number of nodes. For the higher number of nodes, they show their solution performs better than random placement. Liao et al. [2011] propose regular patterns on a plane to achieve full coverage while enabling to localize the target within a certain area not larger than a given parameter, based on the unique set of access points covering the target. Bulusu et al. [2001] address the scenario where the target position is determined as a centroid of positions of all the anchors the target can communicate with. They propose several simple heuristic algorithms to determine the position where an additional node should be placed, given an existing deployment. A robot or a person equipped with GPS and able to measure localization error scans the area and places the additional nodes.

Global Placement Problem for Angle-based and Distance-based Localization

Tekdas and Isler [2010] propose approximation algorithms for triangulation systems, where each point on a plane has to be covered at a good angle by two sensors having infinite ranges. This problem is similar to ours in that it requires more than one node for localization and addresses the global placement problem across the entire workspace. The authors propose an ILP formulation for triangulation problem and report that it takes too long to obtain exact solution for the realistic problem instance they used. Their approximation algorithm does not consider obstacles. Efrat et al. [2005] propose an approximation algorithm for the same problem. For more information on the approximation algorithms see Section 4.1.5.

In Kirchhof [2013] node placement for both triangulation and trilateration is considered. The author also uses GDoP as a quality measure. However, in his approach, only two nodes are required to cover the target in case of trilateration,

while in reality, three nodes are necessary (see Figure 2.2(b)). Kirchhof [2013] provides the execution times of exact algorithms and experimental comparison of proposed non-exact solutions to exact solutions in terms of the number of nodes placed, but only for triangulation. The author reports the number of nodes placed within 1.5 times of the optimal solutions, while our heuristic algorithms achieve much lower overhead of below 2% on the average with respect to the optimal number of anchors, for the problem instances we tested.

The summary of comparison of our approach with the state of the art solutions is given in conclusion, in Section 8.2.

Chapter 5

One-Dimensional Localization

We first focus on 1D localization, in particular, localizing vehicles in road tunnels. Road tunnels generally do not contain forks and prevent the vehicles from going off the road. The position component transversal to the lanes can be neglected because of the consistently small width of the tunnel. Therefore, the majority of road tunnels can be represented as a one dimensional structure.

Numerous vehicular applications and services rely on knowing distances between vehicles and surrounding infrastructure. Vehicle localization systems are among the most common applications that are based on distance measurements. The most widely used technology for vehicle localization is GNSS. As GNSS requires an unobstructed line of sight to satellites. Therefore, it cannot be used in tunnels and other indoor environments.

Wireless Sensors Networks (WSNs) have been widely accepted as an alternative solution to GNSS for vehicle localization: vehicles are equipped with radio transmitter/receiver and interact wirelessly with a network of nodes distributed along the roadside. Wireless nodes are often embedded in existing infrastructure, such as lighting systems, allowing to build ad-hoc WSNs "for free", without dedicated installation. On top of such WSNs, it is possible to design various applications including localization systems. Moreover, safety requirements inside tunnels are particularly important as accidents, such as a fire, will cause more damage due to difficult evacuation. WSNs are ideal for such critical environments as they can be used to monitor other environmental conditions like temperature and smoke (Boukerche et al. [2008]).

Numerous works have shown that in road tunnels, due to tunnel shape, radio communications have significantly different properties than in free space and indoor environments (Dudley et al. [2007]; Liénard et al. [2000]; Sun and Akyildiz [2010]). In this chapter, we estimate in-tunnel performance of Received Signal

Strength and Time of Flight. We then focus on the ToF approach due to its promising results. We analyze the dependency of the ToF ranging error on factors such as node position, distance between transmitter and receiver and measuring at different frequencies. We also derive statistical properties of ToF ranging error. Using the results of our analysis, we design and evaluate a localization system to demonstrate a possible application of ToF-based ranging in a tunnel. The results have shown that such a localization system is not only accurate, but also cheap and scalable as few nodes are necessary even for long tunnels.

We further improve the localization algorithm by performing calibration with the aim of compensating the measurement error. We design an error compensation unit and demonstrate via in-tunnel experiments that the use of such compensation unit significantly improves the performance of an existing localization algorithm.

Our analysis may be very useful to other researchers and engineers aiming to design applications for vehicle ranging and localization, especially considering that testing and measuring campaigns in a real road tunnel are quite complex to arrange (e.g., blocking traffic, sensor nodes installation, etc.).

For our experimental measurements we used two tunnels. The first one is a tunnel located between Vedeggio and Cassarate in Lugano, southern Switzerland. The second one is an underground pedestrian tunnel located at the train station in Lugano, having the similar shape to a road tunnel.

The work in this chapter has been motivated by the PTA project [PTA [2010]], which focused on tracking and localizing vehicles transporting dangerous goods. The results from this chapter are published in Widmann et al. [2013] and Balać et al. [2014].

5.1 Wireless Propagation in Tunnels

The characteristics of wireless propagation in tunnels have been researched in numerous works covering mine tunnels (Exslie et al. [1974]; Lienard and Degauque [2000]), subway tunnels (Didascalou et al. [2001]) and road tunnels (Dudley et al. [2007]; Liénard et al. [2000]; Sun and Akyildiz [2010]).

Generally, a tunnel behaves as a multimodal waveguide for wireless propagation. Attenuation is much lower compared to outdoor or other indoor environments. In smooth tunnels without obstacles path loss can be very close to zero. Communication distance between wireless devices is significantly increased, and so is the distance at which devices can interfere with each other. The tunnel can be divided into two zones. The zone near the transmitter is dominated by

strong fluctuations of the signal power for small changes in position, while in the far zone, path loss increases linearly with distance rather than logarithmically as in the near zone and outdoor areas. The attenuation increases with the number and size of vehicles present in the tunnel. The presence of traffic will result in stronger signal fluctuations, especially close to the vehicles (Yamaguchi et al. [1989]). The propagation in a tunnel strongly depends on the used frequency (Dudley et al. [2007]).

Several localization systems have been created for underground mines, that typically consist of rooms connected with tunnels (Dayekh et al. [2010]). However, wireless propagation in road tunnels has significantly different properties than in mine tunnels. Mine tunnels are much more irregular. Their walls are not covered by smooth concrete and there are variations of the wall surface in order of one meter, which result in significantly increased attenuation and scattering (Lienard and Degauque [2000]).

Our analysis of related work indicates that using localization systems and algorithms designed for other environments may not be an optimal solution for road tunnels. This is due to very different properties of radio wave propagation in road tunnels than in free space, inside buildings, or in mine tunnels. Also, while other works focus on wireless propagation, we analyze the properties of time of flight distance measurement error in road tunnels.

5.2 Hardware

Concerning the hardware, not all transceivers can be used for ToF-based ranging. Exact measurement of frame reception and transmission timestamps are necessary, as well as the right type of modulation. We have chosen NXP JN5148, developed by Jennic NXP [2011]. It implements IEEE 802.15.4 standard, that uses direct sequence spread spectrum modulation, reducing the time quantization error. NXP JN5148 also provides a built-in dedicated hardware unit for precise packet detection and timestamp measurement and libraries for TWToF ranging. Unfortunately, the ranging technique used by Jennic is proprietary and the libraries cannot be customized. We have used the Jennic solution 'as is.'

5.3 Tunnels

We used two different tunnels for our measurements. The first tunnel is the road tunnel named Vedeggio-Cassarate, located in Lugano, Switzerland. It has been

completed in 2012 and at the time of experiments was not open to traffic. The tunnel is 2.6 km long, mainly straight. It has two lanes, and a side-walk on each side. Safety bays are located at every 600 m, and the emergency exits at every 300 m. The Figure 5.1 shows the map of the tunnel.

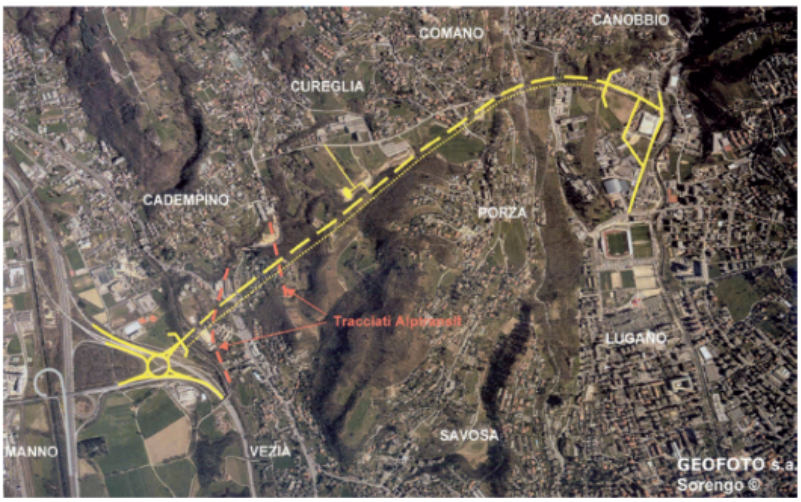


Figure 5.1. Experimental environment: Map of Vedeggio-Cassarate tunnel.

The second tunnel we used is a pedestrian tunnel located near the train station in Lugano. The tunnel is 65 m long, and straight as shown in Figure 5.2. The tunnel is covered by two different kinds of roof: the first part from the stairs is based on concrete square roof and the second part is a curved tin ceiling. Tin ceiling increments the number of reflections and multipath propagation. There is a door in between the two types of roof.

5.4 Comparing RSS to ToF-based Ranging

As RSS systems are cheaper, while ToF is more precise, we characterize and compare RSS and ToF-based ranging in the tunnel environment in order to decide which one is more appropriate for our in-tunnel localization system. These experiments were done in Vedeggio-Cassarate road tunnel. Only two nodes are used: a stationary node on the side of the tunnel and a mobile node placed in the middle of one of the road lanes. The mobile node position was fixed during each measurement phase and then moved to another known location for a new measurement phase. The total distance of about 1 km has been covered with no presence of other vehicles or obstacles. The measurements for RSS and ToF were

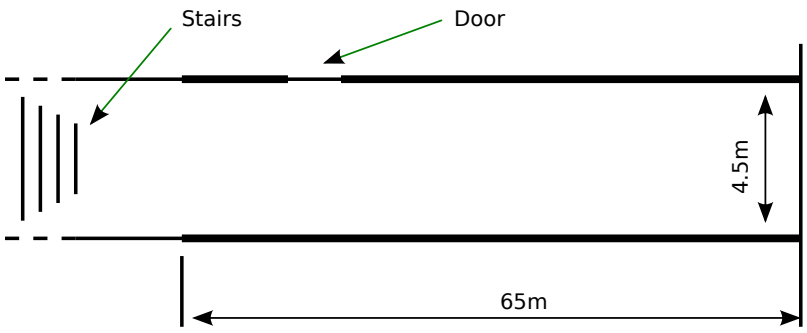


Figure 5.2. Experimental environment: Pedestrian tunnel near the train station in Lugano.

taken at the same time, for the same packets. Their characteristics are reported in the following.

5.4.1 Received Signal Strength

RSS measurement capabilities are included with all WSN hardware which makes this technology cheap to use and readily available.

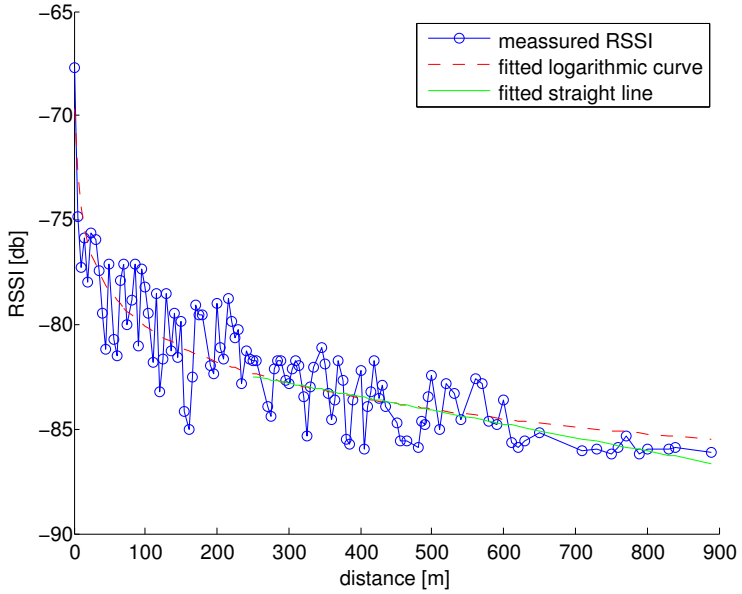


Figure 5.3. RSS measurements depending on distance transmitter-receiver.

For estimating distance based on the RSS, we adopted the well-known approach based on the logarithmic path loss curve (Rappaport [1996]). In order to estimate RSS localization accuracy, we calculated the logarithmic curve parameters as the best fitting curve on the measured data itself. Although this method is very simple and no optimizations have been used, it allowed us to roughly estimate the possibility of using RSS based localization in the tunnel environment.

Figure 5.3 shows the measured RSS with respect to distance between the stationary and the mobile node. We observed two zones: the near zone, up to $200\,m$, with very strong fluctuations, and far zone, above $250\,m$, with less frequent and less intensive fluctuations. In the near zone path loss is approximately logarithmic, while in the far zone it is linear, as can be seen in Figure 5.3, where a logarithmic curve has been fitted to data up to $250\,m$, and a straight line to data above $250\,m$. Our results are in accordance with the results from literature (Dudley et al. [2007]).

The first row of Table 5.1 shows RSS error depending on distance between transmitter and receiver. The RSS ranging error significantly increases with distance.

5.4.2 Time of Flight

Figure 5.4 shows the ToF distance measurements we obtained in the tunnel. The straight line shows the correct distance and the measured distance is shown by circles. In order to reduce the effects of position independent noise, we averaged 100 ToF measurements per target position. We can observe that most of the measurements are very close to the correct position, as well as that for ToF measurements there is no significant influence of distance between transmitter and receiver on the ranging error. However, some measurements still report a significant error that is always higher than the real distance transmitter-receiver. These outliers may be the result of a non line of sight condition between the transmitter and the receiver, caused by obstacles such as signs, emergency exits and safety bays.

The last two rows of Table 5.1 show the mean absolute error for distance estimation with and without outliers. This experiment shows that that the in-tunnel ToF distance measurement precision is high, even for very long distances, as well as that outliers are the main source of error for ToF ranging.

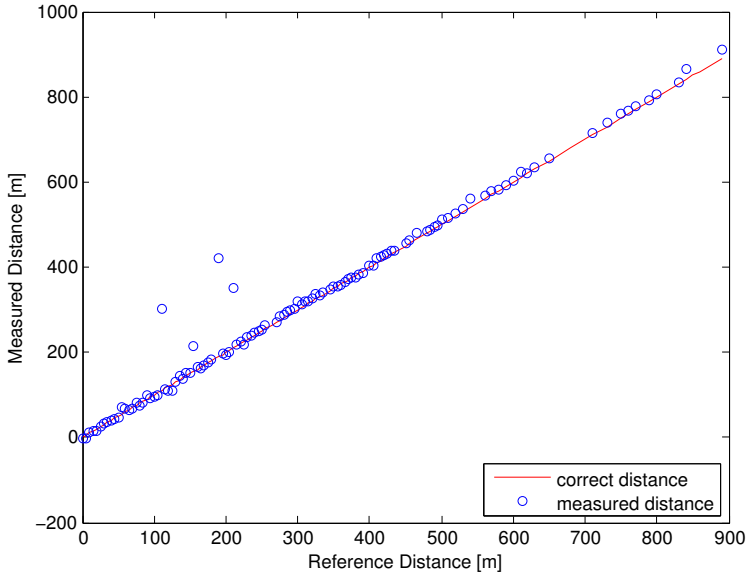


Figure 5.4. ToF measurements in the road tunnel depending on distance transmitter-receiver. The reference, correct, distance is shown by the straight line and measured distance by circles.

Table 5.1. RSS and ToF localization error.

Distance	0-100	100-200	200-300	300-400	400-500	overall
Mean Absolute Error RSS [m]	48.6	175.9	117.2	244.6	278.9	203.3
Mean Absolute Error ToF [m]	5.5	67.7	32.8	5.7	3.8	31.2
Mean Absolute Error ToF Without Outliers [m]	5.5	6.2	5.1	5.7	3.8	6.4

5.4.3 RSS and ToF Comparison

Table 5.1 shows the error comparison for RSS and ToF-based ranging. ToF-based ranging, even in presence of the outliers, gives drastically better results. Note that, in this comparison we considered the RSS and ToF mechanisms available with the Jennic nodes “as is”, without applying any further optimizations. As the RSS measurement error significantly increases with distance, anchor nodes would have to be placed at about every $50\,m$ in order to obtain the vehicle position with sufficient precision using RSS. This would greatly increase the number of nodes necessary for long tunnels and might affect scalability of the system.

On the other hand, ToF is very well suited for ranging in tunnels. Since the error does not strongly depend on distance transmitter-receiver, only few nodes may be necessary to have high ranging accuracy.

Due to the ToF outstanding ranging performance in comparison to the RSS, in the rest of our work we we focus on the ToF approaches.

5.5 Characterization of Time of Flight Ranging

In this Section, we analyze the dependency of ToF ranging error on different environmental factors, as well as statistical properties of ToF ranging error. The goal of our analysis is to help provide the guidelines needed to design our in-tunnel localization system. We also hope our analysis will aid other researchers and engineers in understanding the characteristics of in-tunnel ToF ranging to better model and design localization systems.

Experiments were performed in Vedeggio-Cassarate tunnel, by placing 10 anchor nodes, 5 on each side of the tunnel, with the spacing of 300 m over the whole tunnel length. Four of the nodes were placed in the safety bays as in Figure 5.5. The target was placed in a car, and its distance from the anchors was measured at every 50 m while the vehicle was stopped.

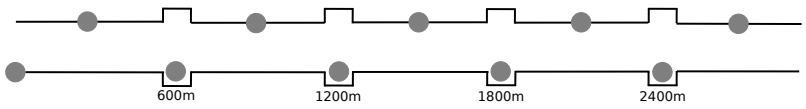


Figure 5.5. Tunnel schematics. Anchor positions are marked with circles.

We provide further details about the ranging error in following sections. To quantify the ranging error, we used its statistical properties: the standard deviation and the mean value.

5.5.1 Time-dependent Error Component

Firstly, we analyze the time-dependent component of ToF ranging error. This is the error component that changes with time, even if the target position is fixed. The ToF distance measurement error can change with time due to changes in the atmospheric conditions, time quantization error, and moving objects in the environment, specifically, traffic in case of tunnels. We calculate the standard deviation of distance measurements depending on the distance between nodes. The anchor node stays at the beginning of the tunnel, and the target moves along the tunnel. At each fixed target position, we take 10 measurements shorty after each other. We assume that the channel characteristics do not change within this short period of time. The results are shown in Figure 5.6.

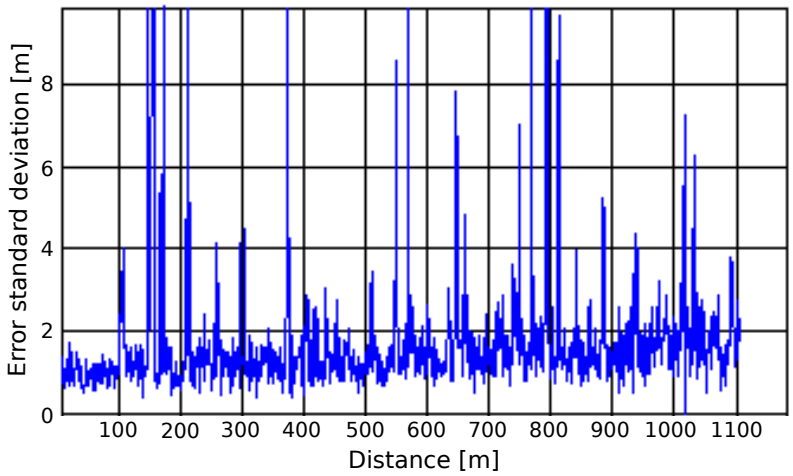


Figure 5.6. Dependency of error standard deviation on distance between nodes.

We can observe that there is a small increase of error standard deviation with the distance between nodes, but it stays well below $3\,m$ over the entire tunnel length, except for some outliers.

Since the atmospheric conditions are not expected to change in short periods of time, the main sources of time-dependent error in tunnels are traffic and quantization. As our experiments were performed with no traffic, the quantization error (see Thorbjornsen et al. [2010]) was probably the main source of time-dependent ranging error we observed.

Averaging measurement samples taken at the same location is an effective way to reduce the error. This method can also be applied to a moving node if the measurements are taken frequently enough with respect to the node speed. In sections 5.5.2 - 5.5.4 we evaluate the position-dependent error component, therefore we average the available data samples at each position, to reduce the time-dependent component.

5.5.2 Ranging Error Dependency on Anchor Position

Another component of ranging error is time independent: it is only influenced by the positions of target and anchors as it is mainly caused by channel characteristics which strongly depend on the environment.

We evaluated the time-independent error component by comparing the measured distance and the real distance between the target and the anchors by using the following experiment: For each vehicle position, we measured the distance

from each anchor at all 16 frequency channels provided by the IEEE 802.15.4 standard. We grouped the measurement errors by the distance from target to anchor into 300 m intervals, then we computed the average measurement error for each interval, using measurements with all available frequencies.

We observed that for most anchors the error mean value was between 0 m and 5 m for all target positions and some other anchors had a significant mean value from 10 to 40 m, again, regardless of the target position. By looking at the anchor positions, we noticed that all of the anchors that had a high measurement error were placed at the safety bays. Moreover, the error mean value was always positive. These errors could be caused by the radio waves reflecting of the walls of the safety bay and resulting in multipath propagation.

Our conclusion is that the anchor nodes should have the best possible line of sight with the target, so to reduce to the minimum the multipath propagation. Thus, anchors should not be placed in the safety bays or close to them.

5.5.3 Ranging Error Dependency on Distance Tag-Anchor

To analyze the error dependency on distance between target and anchor, we used the same data from the experiment described above. We performed statistical analysis on distance intervals of 50 m containing measurements from all anchors and frequencies. The error variance with distance is shown in Figure 5.7.

We can observe distinct peeks of error variance at multiples of 300 m. This mainly happens when the vehicle is close to a safety bay. It could be caused by the same multipath effect that occurs if the anchor is placed close to the safety bay.

5.5.4 Combining Measurements at Different Frequencies

Lanzisera et al. [2011] reported that combining measurements at different frequencies can significantly reduce the noise caused by multipath propagation. To evaluate the effect of frequency on the ToF ranging error, we calculated the distance tag-anchor for each anchor and vehicle position as the median of distances obtained from all 16 frequency channels. We achieved the improvement of error variance from 6.26 m when using only one channel, to 4.7 m.

As can be seen in Figure 5.7, by combining measurements from different channels, not only the average error variance can be significantly reduced, but also the peaks at multiples of 300 m that appear when the vehicle is close to safety bays. However, the outliers could not be eliminated completely, and at 300 m distance there is still a significant error peak.

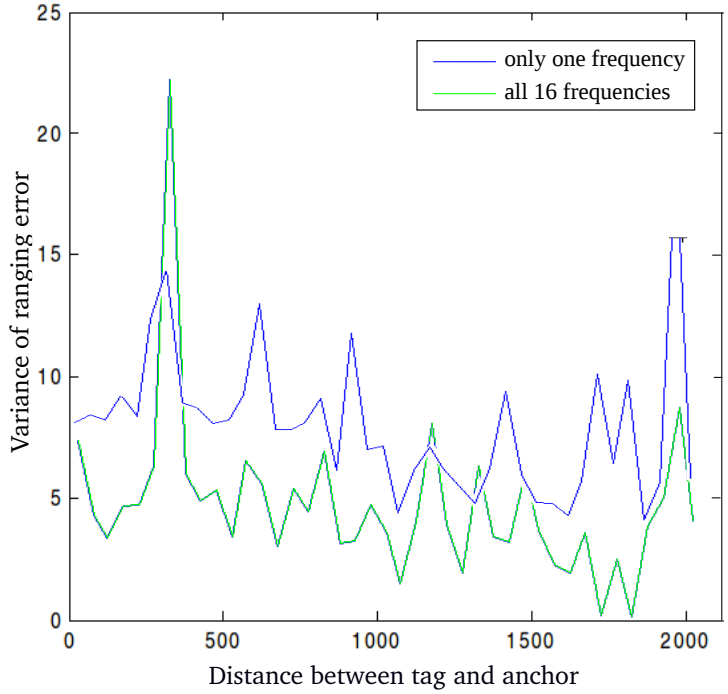


Figure 5.7. Ranging error variance depending on distance between target and anchor.

Table 5.2. Statistical characteristics of ToF ranging error.

	Time-dependent	Time Independent	
		excluding outliers	outliers ($p = 0.03$)
$\mu[m]$	0	0	13.7
$\sigma[m]$	2.05	6.26	31.9

5.5.5 Statistical Properties of Ranging Error

We considered as outliers the measurements that differ by more than 3 error standard deviations from the correct value. In our case, the outliers appear in 3% of all measurements, which is much higher than the 0.3% probability we would expect for a normally distributed random variable. The outliers are biased: in most cases, the measured distance is higher than the correct distance, which strengthens our previous assumption that the non line of sight propagation is the main cause of outliers.

Table 5.2 summarizes the influence of time-dependent and time-independent noise component, as well as the effect of outliers. We further decomposed time-independent noise into two components: noise in the absence of outliers, and another component for outliers only.

Our analysis results in two main conclusions:

- time-independent error component is dominant with respect to time-dependent component;
- the outliers are the dominant source of measurement error.

5.6 Localization Algorithm

The goal of a localization algorithm is to combine the distances obtained in ranging phase in order to get the target position. A straightforward solution to obtain position by using measured distances would be lateration. We will show that by using median filter instead, the localization precision can be significantly increased. To reduce complexity of the algorithm, which is particularly important for resource constrained wireless nodes, we exploit specific shape of a tunnel, representing the tunnel as a one dimensional structure.

The analysis in the previous section showed that the precision can be increased if the algorithm is able to cope with the outliers, that are particularly pronounced when the vehicle is close to a safety bay (see Figure 5.7). Therefore, our main criterion when designing the localization algorithm is robustness in the presence of outliers. Resistance to outliers is achieved by using the median filter, as shown in Algorithm 1. Each of the anchors within the wireless range of the vehicle (*Anchors*) sends its own distance from the vehicle to the central location. Using the obtained set of distances (*Distances*), we first make a rough initial guess of the vehicle position ($initialX$). Note that only one coordinate of vehicle position is computed. We used the lateration algorithm for the initial guess but using any other less complex algorithm is also possible. For each anchor a we compute one candidate vehicle position as follows. The anchor position $AnchorsX[a]$ and its reported distance from the vehicle $Distances[a]$, yield two possible vehicle positions: $AnchorsX[a]+Distances[a]$ and $AnchorsX[a]-Distances[a]$. We chose one of the two options which is closer to the initial guess and add it to the set of candidate positions, $Candidates$. Finally, the vehicle position x is computed as the median of all candidate positions.

Algorithm 1: Localization algorithm for road tunnel.

Data: $Distances$ - from each anchor to target, $PositionsX$ - position of each anchor
Result: x - target position

```
initialX = Lateration(Distances);
forall the a in Anchors do
    if initialX > PositionsX[a] then
        Candidates[a] = AnchorsX[a] + Distances[a];
    else
        Candidates[a] = AnchorsX[a] − Distances[a];
    end
end
x = median(Candidates);
```

We tested our localization algorithm in Vedeggio-Cassarate road tunnel. As our main motivation for designing a localization system is tracking vehicles transporting dangerous goods, the system accuracy is especially important in an emergency scenario, particularly, in case a vehicle would stop inside the tunnel due to an accident. Therefore, we evaluated the algorithm performance for a stationary vehicle. As we can accurately determine the position where the vehicle stops, we can give a precise evaluation of the proposed algorithm.

We placed 6 anchors over the tunnel length of $2.6\,km$. The anchor positions were similar to the ones used in Section 5.5, except for the nodes placed in the safety bays that were not used. Figure 5.8 shows the Cumulative Density function (CDF) of the localization error. Table 5.3 shows the mean absolute error, root mean square error, as well as the ratio of errors that were above 10 m in our experiments, for the median filter and the lateration algorithm. It can be seen that the median filter provides highly superior performance. We believe this is due to its ability to cope with the outliers. The localization error for the median filter is under $6\,m$ in all cases even with only 6 nodes over the entire tunnel length. Using frequency diversity could further improve the results, as shown in Section 5.5.4.

Our results show that the radio frequency time of flight can provide precise in-tunnel localization. Regarding the anchor placement, the main conclusions from the previous section are that the anchors should not be placed in or around the safety bays, and that a small number of anchors should be sufficient to cover the entire tunnel. Due to the wave guide effect a single node placed at one end

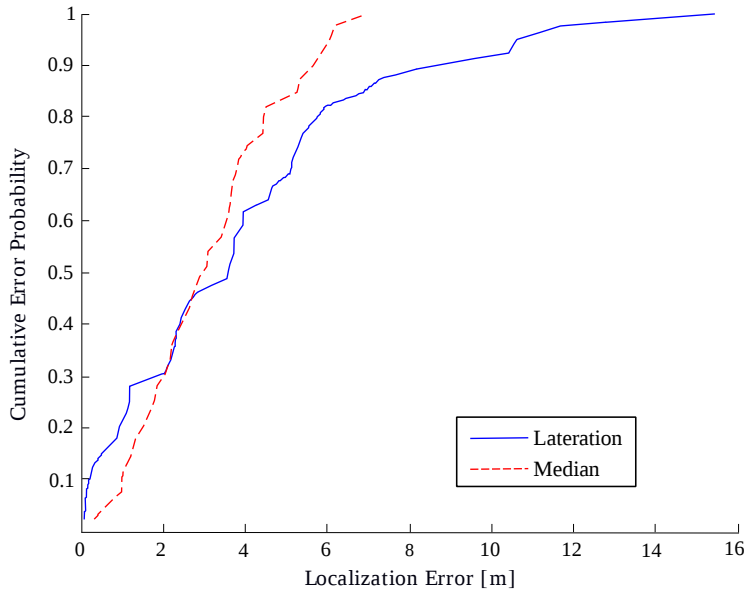


Figure 5.8. Cumulative Density Function (CDF) of the localization error for a stationary vehicle.

Table 5.3. Localization error summary for a stationary vehicle.

	Lateration	Median
Mean Absolute Error [m]	4.06	3.08
Root Mean Square of Error [m]	5.36	3.54
$p(error > 10m)$	0.103	0

of the tunnel can measure distance to the target at the other end, for a tunnel that is 2600 m long.

5.7 Time of Flight Error Compensation for In-Tunnel Vehicle Localization

We further improve the localization algorithm by performing calibration with the aim of compensating the measurement error. The experiments in this section have been performed in the pedestrian tunnel at the train station in Lugano,

having a similar shape to a road tunnel. In this section, we first present the ToF error measurement results obtained in the pedestrian tunnel. Then we propose a localization system model and implement and test both static and dynamic calibration.

5.7.1 Measurements and Error Characterization in the Pedestrian Tunnel

Figure 5.9 shows the measurements between one anchor placed near the tunnel wall and the tag at a fixed position in the middle of the tunnel, at $20\,m$ from the tag. We took $N = 200$ samples, in a short time period. Similar as in the road tunnel, we can notice two error components: a fixed offset, which is time independent, and a time-dependent component. We will refer to these error components as offset and noise. We assume that the offset is due to multipath signal propagation, while the noise component might be partially due to quantization error in ToF measurements. As can be seen in Figure 5.9, the offset component is more significant than the noise component.

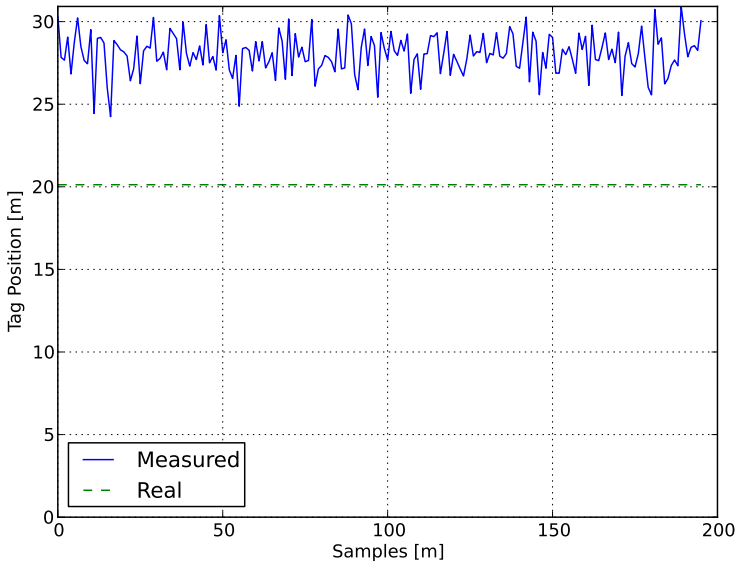


Figure 5.9. Distance measurement samples between the anchor placed near the tunnel wall and the tag at a fixed position in the middle of the tunnel.

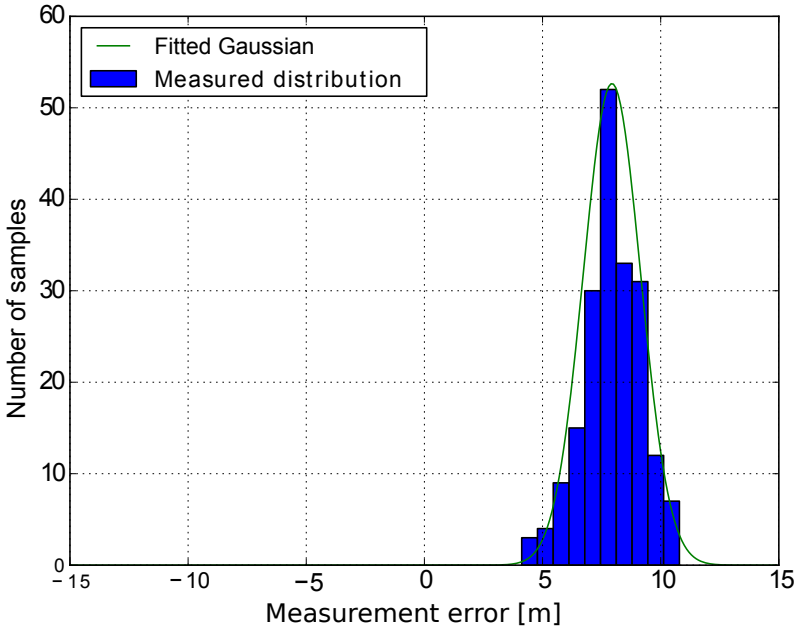


Figure 5.10. Distance measurement error PDF between an anchor placed near the tunnel wall and the tag at a fixed position in the middle of the tunnel.

Figure 5.10 shows the noise distribution as Probability Density Function (PDF) of the error. A Gaussian curve is plotted on the same graph. The offset component is the mean of the Gaussian distribution, and the noise component can be represented as a Gaussian function with mean 0.

In the following experiment, we placed six anchors alternately on both sides of the tunnel. The distance between two anchors at the same side was $20\,m$, interleaved with the anchors at the opposing wall (see Figure 5.11). The tag was moved along the middle of the tunnel at positions $(2.5m : 5m : 47.5m)$. For each pair of anchor-tag position $N = 20$ measurements were taken. The maximum localization error over all experiments was $5.38\,m$, and the mean error was $2.2\,m$.

To quantify the noise we use standard deviation of the measurement error:

$$noise = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\bar{\mathbf{x}} - x_i)^2} \tag{5.1}$$

where $\mathbf{x}$ is the measurement vector, $\bar{\mathbf{x}}$ is the mean of this vector, and N is the number of measurements taken.

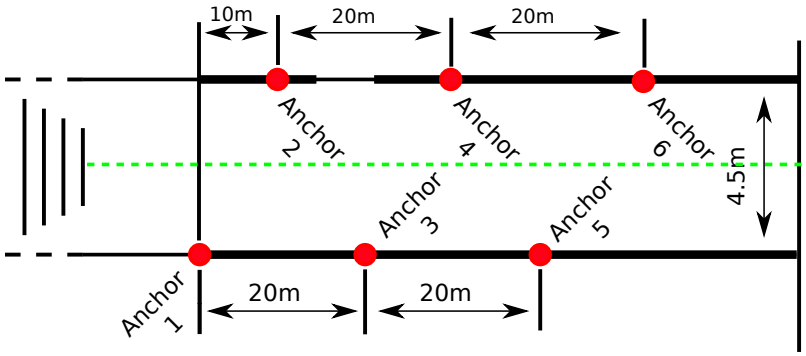


Figure 5.11. Tunnel near Lugano train station - Measurement scenario.

The dependency of the noise on the distance between anchor and tag is shown in Figure 5.12 where different anchors are presented with different shapes. We

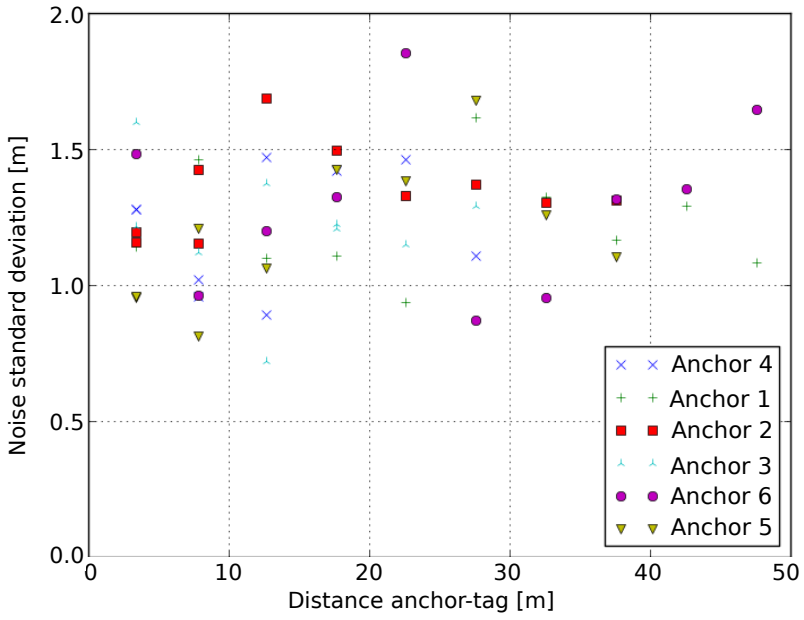


Figure 5.12. Noise standard deviation vs distance tag-anchor.

observe that, similar as for the Vedeggio-Cassarate tunnel, the noise neither depends on the distance between tag and anchor, nor on the anchor position. Such noise can easily be reduced by applying an averaging filter. The offset of a mea-

surements vector of N samples is computed as:

$$offset = \frac{1}{N}\sum_{i=0}^{N}(x_i - d_i), \tag{5.2}$$

where d_i is the real distance from anchor to tag. Figure 5.13 shows the offset values obtained in our experiment. The offset is mostly positive, which confirms our assumption that it comes from multi-path propagation. The error does not correlate with the anchor-tag distance and there are no significant differences depending on anchor position. In Figure 5.14 we compare the offset distribution

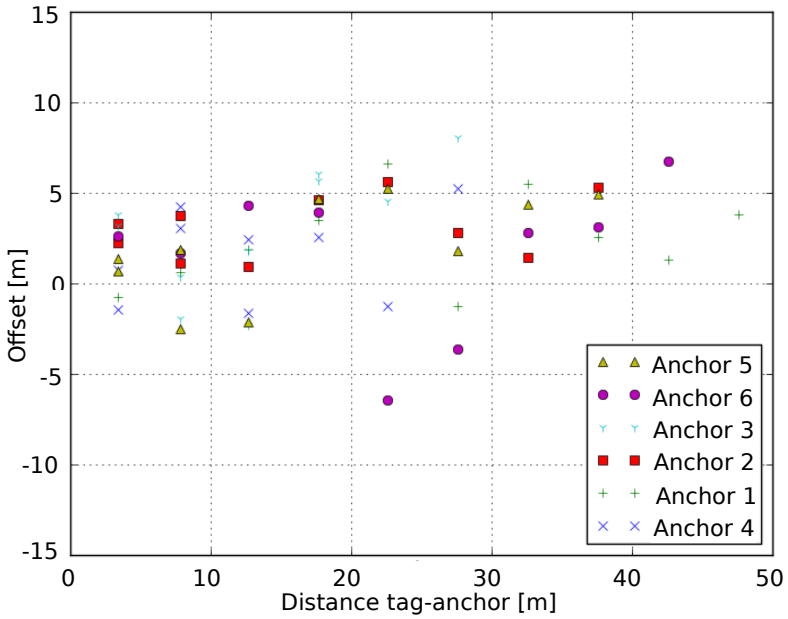


Figure 5.13. Offset vs distance between tag and anchor.

to the Gaussian distribution.

5.7.2 Localization System Model

Based on our error analysis, we propose a model of our localization system. We developed a simulator in Python based on this model, that can be used to quickly prototype and evaluate localization algorithms.

The localization system model is shown in Figure 5.15. It consists of K anchor models and a localization algorithm model. Each anchor model takes as input the

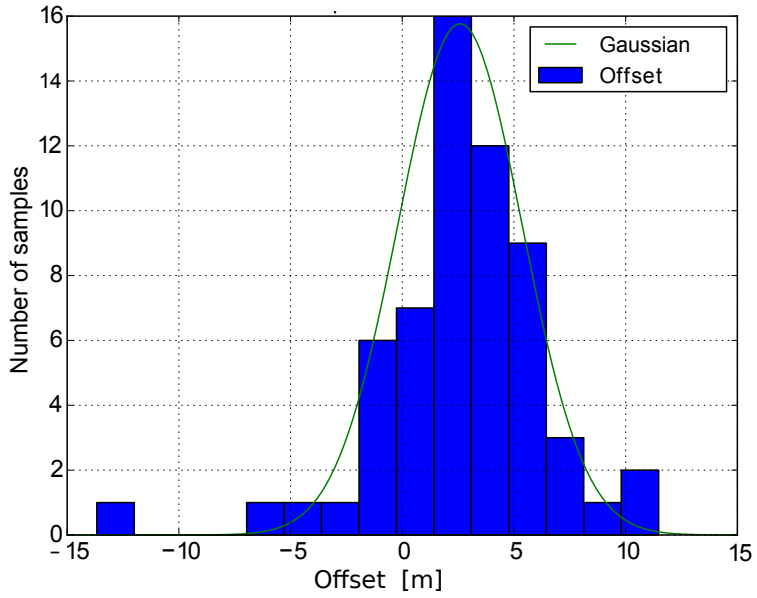


Figure 5.14. Offset PDF.

anchor position (a_x, a_y). The tag position (t_x, t_y), is the same for all anchors, and can change with time. The positions and the number of anchors can be chosen arbitrarily. From these inputs, the real distance d between anchor and tag can be computed. For each anchor, the random Gaussian noise $\sim \mathcal{N}(o, \sigma^2)$ with the mean $(\mu = o)$ and variance (σ^2) is added to d. To represent our tunnel environment, we chose the noise variance as in Figure 5.10, and for each anchor an offset distribution as in Figure 5.14.

A localization algorithm is used to compute the tag position based on the distances measured by the anchors. We used the algorithm presented in Section 5.6.

5.7.3 Compensation Unit

With static calibration, preliminary measurements are collected during the system setup phase. These measurements are used to determine the shape of the corrective function which is later applied to the acquired distance measurements so as to reduce their error. The corrective function can be either linear, polynomial, or wavelet based. A disadvantage of this approach is that it does not consider changes over time of the channel characteristics and it also requires some work to be done in the system setup phase.

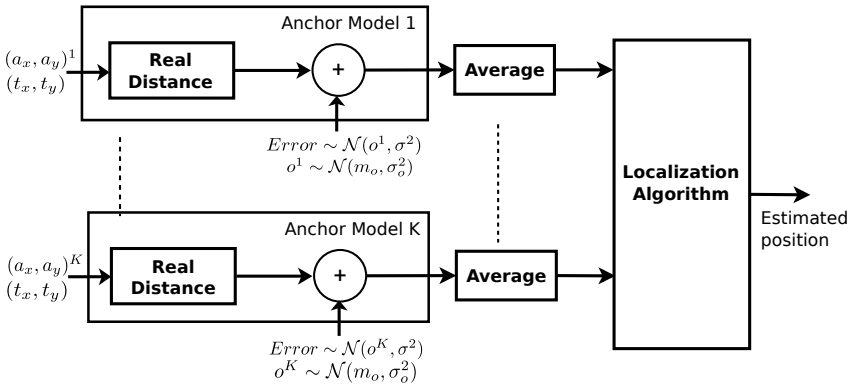


Figure 5.15. Localization System Model.

Automated on-line calibration approaches have been proposed for RSS based systems, where RSS values between anchor nodes with known positions are measured periodically to calibrate the effect of a mutating wireless channel on distance measurements, thus the corrective function is computed at run-time (Gwon and Jain [2004]; Lim et al. [2010]; Redondi et al. [2010]). Inspired by these approaches, we implemented dynamic calibration for our ToF-based distance measurement system.

Static Calibration

We implemented and tested both static calibration and dynamic self-calibration. For static calibration, we train the system by calibrating the distance measurements using a linear regression model. The error compensation unit is located as in Figure 5.16, with the goal of reducing the error between the measured and the real distances. The error compensation process is performed in two phases: the training and the on-line phase, respectively.

The training phase takes place at the system installation time. We use the tag as a probe, moving it along the tunnel at known positions and take distance measurements from each anchor to the tag, as in Figure 5.17. The real distances and the measured distances in the training phase are used to train an individual model for each anchor.

We chose a linear regression model, as the ToF measurement principle is based on a linear relationship between the time the signal flies and the distance traversed. For each anchor k, we determine the coefficients m^k and b^k that we use to compensate the measurements from anchor to tag. The coefficients m^k and b^k are obtained as follows. We measure the distance between the tag and

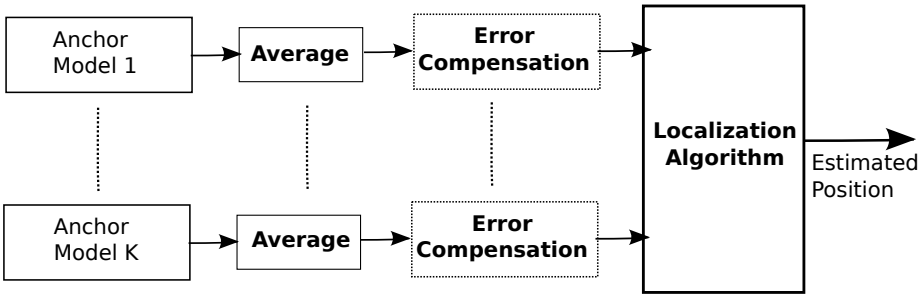


Figure 5.16. Error compensation unit.

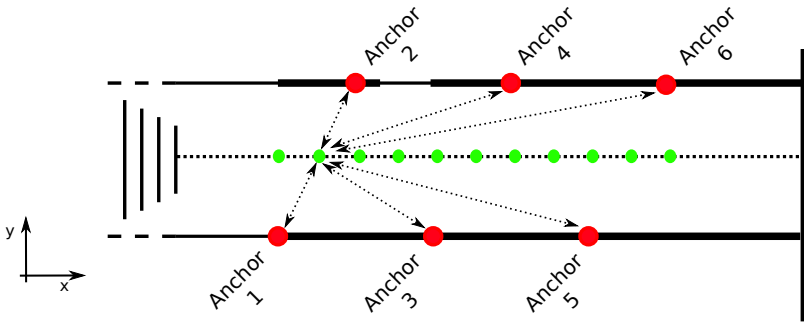


Figure 5.17. Static calibration scenario.

the given anchor, for different tag positions. We denote the obtained vector of measurements as $\{x_n^k\}_n$ and the known vector of real distances as $\{y_n^k\}_n$. The coefficients m^k and b^k are selected so as to minimize the mean square error $\sum_n (y_n^k - g(x_n^k))^2$ where g is a linear predictor function: $g(x_n^k) = m^k x_n^k + b^k$.

The second phase of the error compensation process is the on-line phase, where, for a given anchor, the model parameters computed in the training phase are used to correct the measurements anchor-tag as $d_{corr}(k) = d_k * m^k + b^k$.

The training and the online phase were performed on different days.

Dynamic calibration

Inspired by the RSS-based localization systems in literature, we implemented and evaluated an alternative error compensation method, based on dynamic calibration. This approach avoids calibration in the initial setup phase, but instead calibrates the system periodically, while it is running. The advantage of this approach is that it should mitigate the error variation as the wireless channel changes with time, for example, due to the weather and atmospheric conditions.

To determine the calibration coefficients for each anchor, we used the known

position of the other anchors compared with the measured distance anchor-anchor. Like for static calibration, we used the linear regression model and the mean square error to determine the optimal coefficients. For each anchor k, coefficients m^k and b^k that minimize $\sum_l (y^{kl} - g(x^{kl}))^2$, the mean square error between $\{y^{kl}\}_l$ and the linear predictor function $\{g(x^{kl})\}_l$ are determined, but this time, instead of determining the calibration coefficients in the setup phase and keeping them fixed, they are determined periodically in the on-line phase. The vector $\{y^{kl}\}_l$ is a known vector of distances between the anchor k and all other anchors l, and x_n^l is measured distance between the anchor k and anchor l.

5.7.4 Calibration Algorithm Evaluation

We evaluate the performance of our error compensation unit, both for static and dynamic calibration, by comparing the results of the localization system with and without calibration. The anchors were located as in Figure 5.11.

	Standard deviation [m]	**Mean error [m]**
No Calibration	3.08	3.08
Static Calibration	1.26	1.26
Dynamic Calibration	1.92	1.32

Table 5.4. Static and dynamic calibration.

Table 5.4 compares the performance of static and dynamic calibration. Both methods significantly decrease the localization error. While the mean error is similar for both approaches, static calibration performs better in terms of error standard deviation. As static calibration is also much simpler to implement, it is overall a better choice for our case study. One of the reasons that static calibration exhibits better performance, might be that the tunnel wireless channel characteristics do not change significantly with time, therefore, there is no significant advantage to performing calibration periodically, as in the case with dynamic calibration. Moreover, in the case of dynamic calibration, anchors located next to the tunnel walls are used as a reference for calibrating the system, while the object being localized is in the middle of the tunnel. Wireless propagation characteristics are different near the walls than elsewhere.

Figure 5.18 shows the localization estimation error, with and without static calibration. We can see that static calibration is very effective for increasing localization precision. A reported disadvantage of static calibration in literature is that, with changing environmental conditions, model parameters obtained in

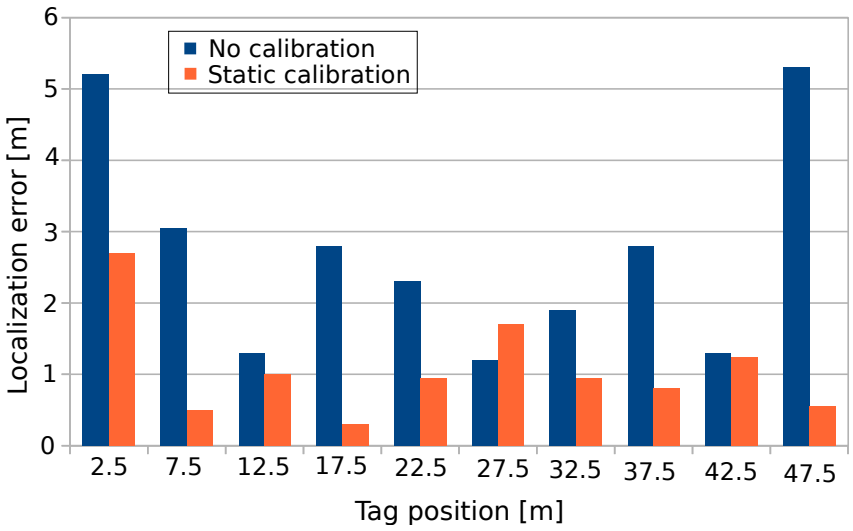


Figure 5.18. Static calibration performance.

the training phase may no longer correspond to the environment and localization precision may decrease. In our opinion, static calibration works well in our case because the environment conditions remain mainly constant with time, as tunnels are semi-closed areas and thus external whether conditions do not significantly affect the signal propagation.

Chapter 6

Two-Dimensional Localization

Our goal in this chapter is to minimize the total number of anchors while guaranteeing that the localization uncertainty is lower than the uncertainty threshold U^* for every target position, or equivalently, that every target position is monitored by at least three non-collinear anchors, in line of sight, not further from it than the node range, while keeping the GDoP lower than $GDoP^*$ (see Section 3.2.4).

Due to the availability of advanced tools for integer linear programs, ILP based formulations allow to obtain exact solutions for reasonably sized instances of NP-hard problems. Availability of optimal solutions for problem instances of limited sizes facilitates the evaluation of any heuristic addressing the same problem by generating different test scenarios. In this chapter, we propose three ILP formulations for the POL problem and compare their performances by simulation experiments. Two of the ILP formulations are general, as they are applicable to any uncertainty function, whenever three anchors are needed to jointly estimate the state of the target. As we will show in Section 6.7.2, which one of them performs better depends on the problem parameters. The remaining formulation is proposed specifically for trilateration based systems. It can be executed in shorter computing time than the other two for the same problem instances. It exploits the results from Section 6.1.2 that allow us to present GDoP in terms of only one angle, rather than all three bearing angles.

We also propose an ILP formulation for the problem of anchor nodes placement optimization for fault-tolerant trilateration-based localization network, where we ensure precise localization at any target position even if any one of the installed anchor nodes fails.

For the case where the problem instance size is too large to handle by using ILP, we propose to use the greedy placement with pruning algorithm. We show by

simulation experiments that this algorithm, in addition to short computing times provides the solutions very close to optimal in terms of the number of anchors placed, for problem instances of sizes relevant for practical applications.

Our results from this section have been published in Balać et al. [2015] Balać et al. [2016].

6.1 Integer Linear Programing Formulations for Placement Optimization for Localization

In this section we propose three different ILP formulations for the anchor placement optimization problem.

6.1.1 The First Integer Linear Programming Formulation for Placement Optimization for Localization

The first ILP formulation for POL can be used for any placement problem where three anchors are needed to jointly estimate the target state. We call this formulation ILP1. It is given by:

$$min : \sum_i x_i \tag{6.1a}$$

$$\sum_i z_i^u = 3, \quad (\forall u), \tag{6.1b}$$

$$z_i^u \leq x_i, \quad (\forall i, u), \tag{6.1c}$$

$$\sum_{j,k} y_{(i,j,k)}^u = z_i^u, \quad (\forall i, u), \tag{6.1d}$$

$$y_{(i,j,k)}^u = 0, \quad (\forall i, j, k, u \mid U(a_i, a_j, a_k, t_u) > U^*) \tag{6.1e}$$

We define one variable x_i for each candidate position $a_i \in C$. The variable x_i will be set to 1 if an anchor is placed at the candidate position a_i. The variable $y_{(i,j,k)}^u$ will be set to 1 if anchors at candidate positions a_i, a_j and a_k are assigned[1] to monitor the target position t_u. The variable z_i^u will be set to 1 if an anchor at candidate position a_i is assigned to monitor the target position t_u.

Equation 6.1a is the objective function, indicating that the total number of anchors in the solution has to be minimized. Equation 6.1b indicates that each

[1]Note that, when the localization system is implemented, the target does not have to know which three anchors are assigned by the placement optimization algorithm to monitor its current position. The target can use more anchors to localize itself, if available. See also Section 3.2.1.

possible target position is assigned three candidates to monitor it. Equation 6.1c indicates that if position a_i is assigned to monitor the target, an anchor has to be placed at position a_i. Equation 6.1d indicates that an anchor a_i is assigned to target t_u if and only if a triple of anchors containing a_i is assigned to t_u. Equation 6.1e ensures that anchors cannot be assigned to a target if they do not meet the quality constraints, imposed by the upper limit on uncertainty function.

Efrat et al. [2005] propose a similar formulation for two nodes that are needed to cover a target. This formulation can be seen as its extension to three nodes. It is well known that high number of variables affects adversely the execution speed of an ILP. Unfortunately, the number of variables in this ILP formulation is very high. Indeed, variables $y^u_{(i,j,k)}$ are defined for combinations of three candidates in the range of each target, therefore, their number is roughly in the order of $N\binom{r}{3}$, where N is the total number of targets and r denotes the average number of candidates in the range of each possible target position. The main motivation behind our second ILP formulation is to reduce the number of variables, by avoiding those variables that are defined for combinations of three candidates. Since three nodes are needed to localize a target jointly, this is not straightforward to achieve.

Our second ILP formulation for POL shows how the number of variables can be reduced. To achieve this, we first derive the alternative condition for the requirement that the GDoP has to be lower than $GDoP^*$, that allows to present the GDoP in terms of only one angle, rather than all three bearing angles, in Section 6.1.2. The second ILP formulation will be given in Section 6.1.3.

6.1.2 Geometric Dilution of Precision for 2D Trilateration

As defined in Section 3.2.4 the localization uncertainty is directly proportional to GDoP, given fixed and known distance measurement standard deviation σ_D. The GDoP for 2D trilateration is given in Equation 3.4 and depends on all three bearing angles in a complex way. In this section, we derive alternative condition for the requirement that the GDoP has to be lower than $GDoP^*$. This alternative condition will allow us to propose an ILP formulation that can tackle problem instances of larger sizes by significantly reducing the number of variables in the ILP formulation.

To limit the localization uncertainty, our goal is to place the anchors in such a way that $GDoP$ is limited from above at every possible target position. Since $\sin 0^0 = \sin 180^0 = 0$, if all three bearing angles are very close to either 0^0 or 180^0, the denominator in Equation 3.4 will be close to 0 and the value of $GDoP$ will be very high. We want to avoid this scenario. Therefore, intuitively, at least

one bearing angle should be far enough from both 0^0 and 180^0. In Proposition 6.1 we express this condition more formally, by proving that if at least one of the angles α, β and γ is separated by θ from both 0 and 180^0, then GDoP is limited from above by a function of θ.

Proposition 6.1. *Let α, β and γ, $\alpha+\beta+\gamma = 360^0$ be the bearing angles and $\theta(\alpha,\beta,\gamma) = \max(\min(|\alpha|, |180^0-\alpha|), \min(|\beta|, |180^0-\beta|), \min(|\gamma|, |180^0-\gamma|))$, then $GDoP < g(\theta) \cdot GDoP_{min}$, where $GDoP_{min} = \frac{2}{\sqrt{3}}$,*
GDoP is defined in Equation 3.4, and $g(\theta) = \frac{3}{2\sqrt{2-cos\theta-cos^2\theta}}$.
If $\theta < 60^0$, then $GDoP > h(\theta) \cdot GDoP_{min}$, where $h(\theta) = \frac{3}{2\sqrt{2-2cos^2\theta}}$.

Proof. We have $0 \leq \theta \leq 90^0$
$0 \leq \alpha,\ \beta,\ \gamma < 360^0$
We can assume without loss of generality that $\alpha \leq \beta$ and
$\gamma \in \{\theta,\ 180^0-\theta,\ 180^0+\theta,\ 360^0-\theta\}$, from which we have
$\alpha+\beta \in \{\theta,\ 180^0-\theta,\ 180^0+\theta,\ 360^0-\theta\}$, and
$\min(|\alpha|,\ |180^0-\alpha|) \leq \theta$,
$\min(|\beta|,\ |180^0-\beta|) \leq \theta$ which give us four different cases:

1. $\alpha+\beta = \theta$, $0 \leq \alpha \leq \theta$, $0 \leq \beta \leq \theta$
   $sin^2\alpha + sin^2\beta + sin^2\gamma = sin^2\alpha + sin^2(\theta-\alpha) + sin^2\theta$.
   By differentiating the right hand expression over α we get that it has its minimum for $\alpha = \frac{\theta}{2}$ which equals $2-cos\theta-cos^2\theta$, and a maximum at the edge of the interval for $\alpha = 0$ which equals $2sin^2(\theta) = 2-2cos^2(\theta)$.

2. only for $\theta > 60^0$
   $\alpha+\beta = 180^0-\theta$, $0 < \alpha \leq \theta$, $0 < \beta \leq \theta$,
   $sin^2\alpha + sin^2\beta + sin^2\gamma = sin^2\alpha + sin^2(\theta+\alpha) + sin^2\theta$.
   This expression has its minimum for $\alpha = 0$ which equals
   $2-2cos^2\theta > 2-cos\theta-cos^2\theta$.

3. $\alpha+\beta = 180^0+\theta$, $0 \leq \alpha \leq \theta$, $180 \leq \beta \leq 180^0+\theta$
   $sin^2\alpha + sin^2\beta + sin^2\gamma = sin^2\alpha + sin^2(\theta-\alpha) + sin^2\theta$.
   By differentiating the right hand expression over α we get that it has its minimum for $\alpha = \frac{\theta}{2}$, $\beta = 180^0 + \frac{\theta}{2}$ which equals $2-cos\theta-cos^2\theta$, and a maximum at the edge of the interval for $\alpha = 0$ which equals $2sin^2(\theta) = 2-2cos^2(\theta)$.

4. $\alpha+\beta = 360^0-\theta$, $180^0-\theta < \alpha,\ \beta < 180^0+\theta$.
   $sin^2\alpha + sin^2\beta + sin^2\gamma = sin^2\alpha + sin^2(\theta+\alpha) + sin^2\theta$.

By differentiating the right hand expression over α we get that it has its minimum for $\alpha = \beta = 180^0 - \frac{\theta}{2}$, which equals $2 - cos\theta - cos^2\theta$, and a maximum at the edge of the interval for $\alpha = 180^0 - \theta$, $\beta = 180^0 - \theta$ which equals $2sin^2(\theta) = 2 - 2cos^2(\theta)$.

From 1, 2, 3 and 4 we get $sin^2\alpha + sin^2\beta + sin^2\gamma > 2 - cos\theta - cos^2\theta$. By substituting in Equation 3.4, we directly prove the first part of our proposition: $GDoP < g(\theta) \cdot GDoP_{min}$.
From 1, 3 and 4 we get $sin^2\alpha + sin^2\beta + sin^2\gamma < 2sin^2\theta$ for $\theta < 60^0$. By substituting in Equation 3.4, we prove the second part of our proposition:
$GDoP > h(\theta) \cdot GDoP_{min}$. □

$GDoP_{min}$ is the minimum possible value for GDoP and is used here for normalization.

Based on this proposition, we will replace the condition $GDoP < GDoP^*$ by a more conservative condition $\theta(\alpha, \beta, \gamma) > \theta^*$, where $GDoP^* = g(\theta^*)GDoP_{min}$.

The functions g and h are close in values. See Figure 6.1. For example, we have $g(37^0) = 2$, $h(37^0) = 1.76$. The cases where $GDoP$ is required to be very close to $GDoP_{min}$ are not of much significance. The ranging error standard deviation in LoS conditions, σ_D, is typically very low (Alavi and Pahlavan [2006]). Therefore, in order to keep the localization error standard deviation σ_0 low, it is not necessary to limit $GDoP$ to very low values and it is unlikely that higher values of θ will be required.

We will also show graphically that if at least one bearing angle is far enough from both 0 and 180^0, then the GDoP is limited from above. Here, we introduce new notation. Each pair of anchors forms two conjugate bearing angles with the target, for example α and $360^0 - \alpha$. We introduce α_0, β_0 and γ_0, such that these angles are always less than 180^0, defined as: $\alpha_0 = \alpha$, if $\alpha < 180^0$, $\alpha_0 = 360^0 - \alpha$, otherwise. Analogously for β_0 and γ_0 (see Figure 6.2(b) - 6.2(c)). Notice that if we substitute, for example, α for α_0 in Equation 3.4, the value of $GDoP$ will not change. We change both α and β from 0 to 360^0 in small increments and calculate $GDoP$, as given by Equation 3.4, for each value pair. note that γ is dependent on α and β. The minimum value of $GDoP$, $GDoP_{min} = \frac{2}{\sqrt{3}}$ is achieved for $\alpha = \beta = \gamma = 120^0$. Figure 6.2(a) shows all of the values where $GDoP < GDoP^* = 2 \cdot GDoP_{min}$ in white, and those where $GDoP > GDoP^*$ in black. The black regions are marked with numbers from 1 to 6.

By observing Figure 6.2(a), we can see that if at least one bearing angle α_0, β_0 or γ_0, is between θ^* and $180^0 - \theta^*$, then the value of $GDoP$ is not in any of the black regions. For example, we can show that in Black Region 2 none of the

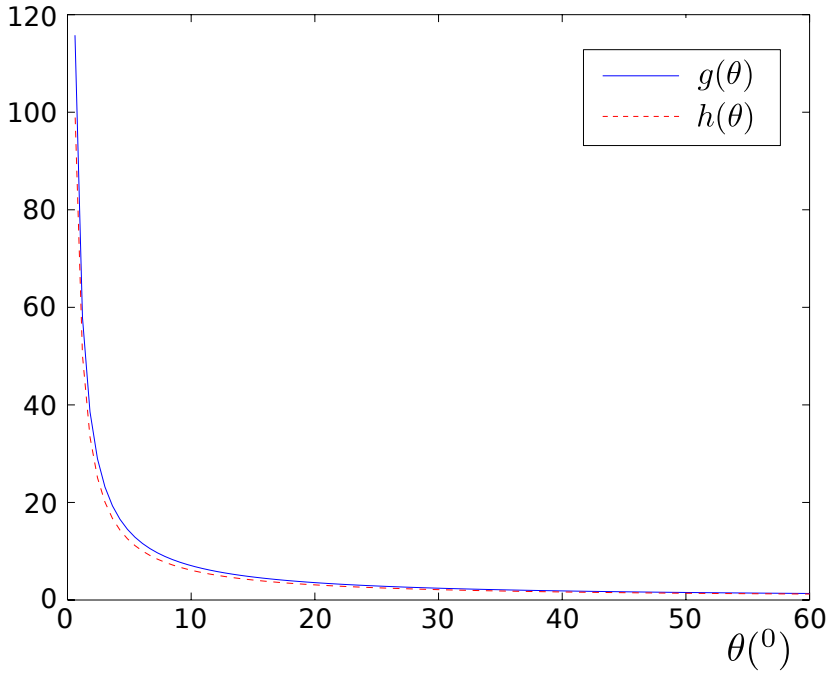


Figure 6.1. The functions $g(\theta)$ and $h(\theta)$.

bearing angles is between θ and $180^0 - \theta$ as follows. We describe Black Region (2) in Figure 6.2(a), and we get:

$$\begin{aligned} 180^0 - \theta^* < \alpha < 180^0 + \theta^* &\longrightarrow \alpha_0 > 180^0 - \theta^*. \\ \beta < \theta^* \text{ (line b)} &\longrightarrow \beta_0 = \beta < \theta^*. \\ \alpha + \beta < 180^0 + \theta^* \text{ (line c)} &\longrightarrow \gamma > 180^0 - \theta^*, \\ \alpha > 180^0 - \theta^* &\longrightarrow \gamma < 360^0 - \alpha < 180^0 + \theta^*, \\ 180^0 - \theta^* < \gamma < 180^0 + \theta^* &\longrightarrow \gamma_0 > 180^0 - \theta^*. \end{aligned} \tag{6.2}$$

In a similar way, we can show that in all six black regions in Figure 6.2(a), all angles $\alpha_0, \beta_0, \gamma_0$ are either less than θ^* or greater than $180^0 - \theta^*$. Also, we can see that the gray dashed line in Figure 6.2(a) approximates the border between the white and black area quite closely.

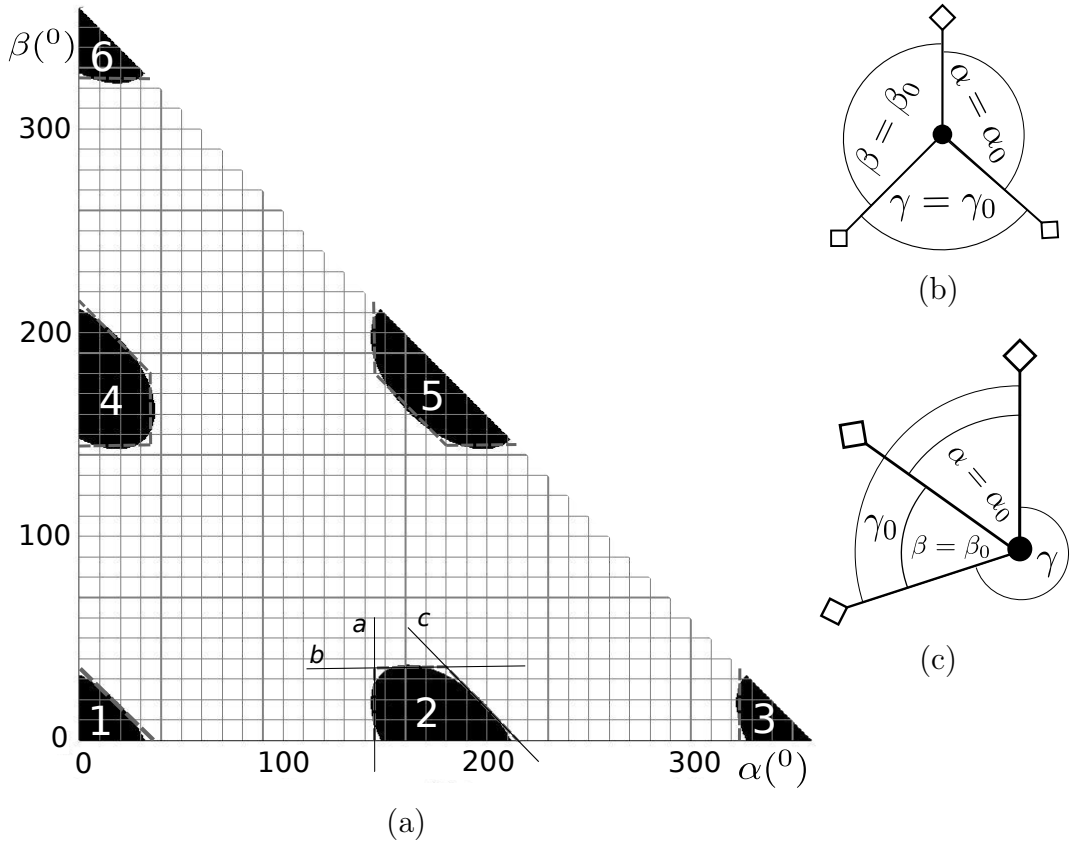


Figure 6.2. (a) Regions where $GDoP > 2 \cdot GDoP_{min}$ are shown in black. (b) - (c) Bearing angles. Three anchor nodes, indicated by diamond shapes, cover the target, indicated by the black dot.

6.1.3 The Second Integer Linear Programming Formulation for Placement Optimization for Localization

Our second ILP formulation for POL shows how the number of variables can be reduced, by exploiting the results from Section 6.1.2 that allow us to present GDoP in terms of only one angle, rather than all three bearing angles. We will show in Section 6.7 that this indeed results in much lower execution times. We call this formulation second ILP formulation for placement optimization for localization (ILP2).

Before proceeding to ILP formulation we introduce some notation. Given U^*, we define θ^* such that $g(\theta^*)GDoP_{min}\sigma_D = U^*$. This will guarantee that if placement of anchors P is such that a target t_u is monitored by tree non-collinear nodes in range, within the LoS and at least one of the bearing angles is between θ^* and

$180^0 - \theta^*$, then $U(P, t_u) < U^*$. For each possible target position t_u we define a set $R(t_u)$ such that the candidate position $a_i \in R(t_u)$ if the distance between a_i and t_u is lower than the node range and there is line of sight between t_u and a_i. We will also say that the candidate a_i reaches t_u if $a_i \in R(t_u)$. Notation $\text{Col}(a_i, a_j, a_k)$ denotes that candidate positions a_i, a_j and a_k are collinear. Notation $\angle(a_i, t_u, a_j)$ denotes the bearing angle formed by candidate positions a_i and a_j with vertex at target t_u. We say that an angle $\angle(a_i, t_u, a_j)$ is *adequate* if it is bigger than θ^* and smaller than $180^0 - \theta^*$, and denote this as a boolean function $\text{A}(\angle(a_i, t_u, a_j), \theta^*)$, which takes value $true$ if $\angle(a_i, t_u, a_j)$ is adequate.

We assign three anchor nodes at non-collinear candidate positions: a_i, a_j and a_k, such that $a_i, a_j, a_k \in R(t_u)$ to monitor each target t_u. Two of these three candidate positions, a_i and a_j have to form an adequate angle with the target (Figure 6.3). We do not place any constraints on the angles formed by a_k with other candidate positions.

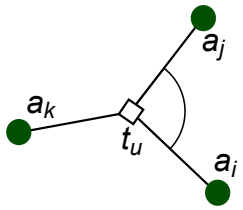


Figure 6.3. The anchor nodes, indicated by the filled circles, monitor the target, indicated by the diamond shape. The angle marked with an arc is guaranteed to be adequate.

Instead of variables $y^u_{(i,j,k)}$ indicating that candidate positions a_i, a_j and a_k are assigned to monitor target position t_u, we have two new sets of variables. The variable $y^u_{(i,j)}$ will be set to 1 if a_i and a_j are those two candidate positions assigned to target t_u that have to form an adequate angle with t_u, and the variable w^u_k will be set to 1 if a_k is the third candidate assigned to monitor target t_u, without any constraints on angles formed with other positions. This is how the reduction of the number of variables is achieved. The variable x_i will be set to 1 if an anchor is placed at the candidate position a_i. The variable z^u_i will be set to 1 if a_i is one of the two candidate positions assigned to target t_u that is guaranteed to form an adequate angle with another candidate position assigned to t_u.

The ILP formulation is given by:

$$min : \sum_i x_i, \tag{6.3a}$$

$$\sum_i z_i^u = 2, \quad (\forall u), \tag{6.3b}$$

$$\sum_i w_i^u = 1, \quad (\forall u), \tag{6.3c}$$

$$\sum_{x_i \in R(t_u)} x_i \geq 3, \quad (\forall u), \tag{6.3d}$$

$$z_i^u \leq x_i, \quad (\forall i, u), \tag{6.3e}$$

$$w_i^u \leq x_i, \quad (\forall i, u), \tag{6.3f}$$

$$z_i^u = 0, \quad (\forall i, u \mid a_i \notin R(t_u)), \tag{6.3g}$$

$$w_i^u = 0, \quad (\forall i, u \mid a_i \notin R(t_u)), \tag{6.3h}$$

$$\sum_j y_{(i,j)}^u = z_i^u, \quad (\forall i, u), \tag{6.3i}$$

$$y_{(i,j)}^u = 0, \quad (\forall i, j, u \mid \neg \mathrm{A}(\angle(a_i, t_u, a_j), \theta^*)) \tag{6.3j}$$

$$y_{(i,j)}^u + w_i^u <= 1, \quad (\forall i, j, u), \tag{6.3k}$$

$$y_{(i,j)}^u + w_k^u <= 1, \quad (\forall i, j, k, u \mid \mathrm{Col}(a_i, a_j, a_k)). \tag{6.3l}$$

Equation 6.3a states that the total number of anchors used has to be minimized. Equations 6.3b and 6.3c ensure respectively that for each target t_u there are exactly two anchors assigned to it that have to form an adequate bearing angle with t_u, and exactly one anchor that does not have to form any adequate angles. Each possible target position has to be reached by at least three anchors (6.3d). Equations 6.3e and 6.3f ensure that a candidate position can be assigned to monitor a target only if an anchor is placed at that candidate position. Equations 6.3g and 6.3h ensure that a candidate position cannot be assigned to target t_u if it does not reach t_u. Equation 6.3i ensures that a candidate a_i is one of the nodes assigned to target t_u that have to form an adequate angle with t_u and another candidate, if at only if a pair of candidates that contains a_i is assigned, forming an adequate angle, to t_u. Equation 6.3j ensures that if candidates a_i and a_j are assigned to t_u such that they have to form an adequate angle, then indeed they do form an adequate angle with t_u. Equation 6.3k, indicates that the two points assigned to a target that have to form an adequate angle with it are distinct from the third point assigned to this target. Finally, Equation 6.3l ensures that the three candidate positions assigned to a target are not collinear.

6.1.4 The Third Integer Linear Programming Formulation for Placement Optimization for Localization

In our third formulation, we reduce the number of variables even more by completely avoiding variables that combine several anchors that cover the target together, and only have one anchor assigned to a target in each variable. The idea behind this formulation is that instead of expressing which nodes can monitor the target together, we express the negative condition - which of the neighboring nodes cannot monitor the target together. As we will also show in Section 6.7.2, this formulation scales better than ILP1 for the small number of obstacles, due to the lower number of variables. However, with the increased number of obstacles, its performance decreases, as the number of equations grows due to more combinations of nodes that cannot cover the target together. The same as ILP1, this ILP can be used wherever three nodes are needed to cover the target together.

The set $R(t_u)$ is the set of all nodes that reach the target t_u, defined in the same way as in 6.1.3. We define following variables:

- $x_i = 1$ if a node is placed at the candidate position i, equivalently, if candidate position i belongs to the solution. There is one variable x_i for each candidate position i.
- $z_i^u = 1$ if the candidate position i is assigned to target u.

All variables can only take values 0 or 1.

The ILP3 formulation is given by:

$$min : \sum_i x_i, \tag{6.4a}$$

$$\sum_i z_i^u = 3, \quad (\forall u), \tag{6.4b}$$

$$z_i^u \leq x_i, \quad (\forall u, i), \tag{6.4c}$$

$$z_i^u + z_j^u + z_k^u < 3, \quad (\forall i, j, k, u \mid U(a_i, a_j, a_k, t_u) > U^*) \tag{6.4d}$$

$$z_i^u = 0, \quad x_i \notin R(t_u). \tag{6.4e}$$

Equation 6.4a states that the total number of anchors used has to be minimized. Equation 6.4b states that each target u has to be assigned three anchors to monitor it. Equation 6.4c ensures that an anchor at candidate position i can be assigned to a target only if that candidate location belongs to a solution, that is, if

an anchor is placed at the position i. Equation 6.4d ensures that anchors cannot be assigned to a target if they do not meet the quality constraints, imposed by the upper limit on uncertainty function. Equation 6.4e ensures that an anchor can only be assigned to a target if it reaches the target.

6.2 Integer Linear Programing Formulation for Fault-tolerant Placement Optimization for Localization

In this section, we propose an ILP formulation for the fault-tolerant placement for UWB trilateration-based localization problem.

Each target has to be 2-monitored by four anchors. We say that a target is 2-monitored by a set of K anchors if, when any of these anchors is removed, the remaining $K-1$ anchors are sufficient to monitor the target. As derived in Section 3.2.4, three anchors monitor a target in 2D if they are not collinear, closer to the target than the node range with no obstacles in between, and at least two of them form an adequate angle with the target. All configurations of four anchors and a target, such that the anchors 2-monitor the target, can be classified into two categories (see Figure 6.4). Either there are two pairs of anchors such that each pair forms an adequate angle with the target (Figure 6.4(a)), or three out of four anchors form three adequate angles with the target (Figure 6.4(b)). In both cases, no three out of the four anchors are collinear.

If a subset of candidate positions is a solution to our problem, then to each target, we can assign four anchors from this solution that 2-monitor it together. These four anchors form one of the two configurations in Figure 6.4. All angles marked by an arc in Figure 6.4 have to be adequate, and this is guaranteed by our ILP formulation. Angles not marked with an arc may or may not be adequate.

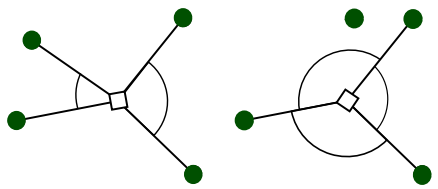


Figure 6.4. Possible configurations that 2-monitor the target. The target is marked by the diamond shape, and anchors by the filled circles. All angles marked by an arc are guaranteed to be adequate.

Before we proceed to the ILP formulation, we introduce the short notation $A((a_i, a_j, a_k), t_u, \theta^*)$ which stands for $A(\angle(a_i, t_u, a_j), \theta^*) \wedge A(\angle(a_i, t_u, a_k), \theta^*) \wedge A(\angle(a_j, t_u, a_k), \theta^*)$, meaning that the anchors a_i, a_j, a_k pairwise form adequate angles with target t_u. If we denote the three anchors in Figure 6.4(b) that form the angles marked by arcs with the target with a_i , a_j and a_k and we denote the target in the same figure with t_u, then we have $A((a_i, a_j, a_k), t_u, \theta^*) = true$.

We define the following variables. The variable x_i is set to 1 if a node is placed at the candidate position a_i. The variable ω_u u is set to 0 if the target t_u is assigned anchor nodes in configuration 6.4(a), and $\omega_u = 1$ if it is assigned anchors in configuration 6.4(b). The variable $y^u_{(i,j)}$ will be set to 1 if target t_u is assigned anchors in configuration 6.4(a), and nodes a_i and a_j have to form an adequate angle with t_u (a_i and a_j are a pair of nodes with an arc between them in Figure 6.4(a)). The variable z^u_i is set to 1 if target t_u is assigned configuration 6.4(a), and the candidate position a_i is assigned to target t_u. The variable v^u_i will be set to 1 if configuration assigned to target u is 6.4(b), node a_i is assigned to target t_u, while a_i has to form three adequate angles with other two nodes assigned to t_u. The variable t^u_i will be set to 1 if configuration assigned to target t_u is 6.4(b), the candidate a_i is assigned to target t_u, and does not have to form an adequate angle with any other node and t_u. All variables can only take values 0 or 1.

The ILP formulation is given by:

$$min : \sum_i x_i \tag{6.5a}$$

$$y^u_{i,j} \leq 1 - \omega_u, \quad (\forall i, j, u), \tag{6.5b}$$

$$z^u_i \leq 1 - \omega_u, \quad (\forall i, u), \tag{6.5c}$$

$$\sum_{(i,j)} y^u_{(i,j)} = 2(1 - \omega_u), \quad (\forall u), \tag{6.5d}$$

$$\sum_i z^u_i = 4(1 - \omega_u), \quad (\forall u), \tag{6.5e}$$

$$z^u_i \leq x_i, \quad (\forall i, u), \tag{6.5f}$$

$$\sum_j y^u_{(i,j)} = z^u_i, \quad (\forall i, u), \tag{6.5g}$$

$$y^u_{(i,j)} + y^u_{(i,k)} \leq 1, \quad (\forall i, u), \tag{6.5h}$$

$$z_i + z_j + z_k \leq 2, \quad (\forall i, j, k, u \mid Col(a_i, a_j, a_k)), \tag{6.5i}$$

$$v^u_i \leq \omega_u, \quad (\forall i, u), \tag{6.5j}$$

$$t^u_i \leq \omega_u, \quad (\forall i, u), \tag{6.5k}$$

$$\sum_i v_i^u = 3\omega_u, \quad (\forall u), \tag{6.5l}$$

$$v_i^u + v_j^u \leq 1, \quad (\forall i, j, u \mid \neg A(\angle(a_i, t_u, a_j), \theta^*)), \tag{6.5m}$$

$$\sum_i t_i^u = \omega_u, \quad (\forall u), \tag{6.5n}$$

$$v_i^u + v_j^u + v_k^u \leq 2, \quad (\forall i, j, k, u \mid Col(a_i, a_j, a_k)), \tag{6.5o}$$

$$v_i^u + v_j^u + t_l^u \leq 2, \quad (\forall i, j, l, u \mid Col(a_i, a_j, a_l)), \tag{6.5p}$$

$$v_i^u \leq x_i, \quad (\forall i, u), \tag{6.5q}$$

$$t_i^u \leq x_i, \quad (\forall i, u), \tag{6.5r}$$

$$y_{(i,j)}^u = 0, \quad (\forall i, j, u \mid \neg A(\angle(a_i, t_u, a_j), \theta^*)), \tag{6.5s}$$

$$v_i^u + t_i^u \leq 1, \quad (\forall i, u), \tag{6.5t}$$

$$z_i^u = 0, \quad (\forall i, u \mid a_i \notin R(t_u)), \tag{6.5u}$$

$$v_i^u = 0, \quad (\forall i, u \mid a_i \notin R(t_u)), \tag{6.5v}$$

$$w_i^u = 0, \quad (\forall i, u \mid a_i \notin R(t_u)), \tag{6.5w}$$

$$z_i^u = 0, \quad (\nexists j \in R(t_u) \mid A(\angle(a_i, t_u, a_j), \theta^*)), \tag{6.5x}$$

$$v_i^u = 0, \quad (\nexists j, k \in R(t_u) \mid A((a_i, a_j, a_k), t_u, \theta^*)), \tag{6.5y}$$

$$x_i, \omega_u, y_{i,j}^u, z_i^u, v_i^u, t_i^u \in \{0, 1\}. \tag{6.5z}$$

Equation 6.5a states that the total number of anchors used has to be minimized. Equations 6.5b and 6.5c state that the variables $y_{(i,j)}^u$ and z_i^u can only be equal to 1 if anchor t_u is assigned configuration as in Figure 6.4(a). Analogously, Equations 6.5j and 6.5k state that the variables v_i^u and t_i^u can only be 1 if node t_u is assigned configuration as in Figure 6.4(b). Equation 6.5d states that, if the node t_u is assigned the first configuration, it has to be assigned exactly two pairs of nodes with adequate angles, and Equation 6.5e, that it has to be assigned four anchors. Equations 6.5f, 6.5q and 6.5r ensure that an anchor at candidate position a_i can be assigned to a target only if that candidate location belongs to a solution, that is, if an anchor is placed at the position a_i. Equation 6.5g ensures that, for the configuration in Figure 6.4(a), a candidate is assigned to a target position, if and only if at least one pair of candidates it belongs to is assigned to this target position. Equation 6.5h ensures that, for the configuration in Figure 6.4(a), two pairs covering the target with adequate angles are disjoint. Equations 6.5i, 6.5o and 6.5p guarantee that no three candidates assigned to the same target are collinear. Equation 6.5l states that, if the node t_u is assigned the second configuration, the number of nodes assigned to it that pairwise form adequate angles with the target is exactly three, and Equation 6.5n, that it has

to be assigned exactly one other node with no guarantee about angles. Equation 6.5m ensures that in second configuration, all angles between the three anchors are adequate. Equation 6.5t ensures that in the second configuration, the three anchors that have to form adequate angles pairwise with the target are not incident with the fourth point, that has no guarantee about angles. Equation 6.5s ensures that if candidates a_i and a_j are assigned to t_u such that they have to form an adequate angle, then indeed they do form an adequate angle with t_u. Equations 6.5u, 6.5v and 6.5w ensure that a candidate position cannot be assigned to target t_u if it does not reach t_u. Equation 6.5x ensures that z_i^u can be set to 1 only if it forms adequate angles with the target u together with some other nodes that reach t_u. Equation 6.5y ensures that v_i^u can be set to 1 only if it forms three adequate angles with the target u together with some other two nodes that reach u.

6.3 Greedy Placement with Pruning

If the size of a problem instance is too large to tackle by using ILP, we propose to use our greedy placement with pruning algorithm. This algorithm can be used both for POL and FPOL. Where necessary, we will make the distinction between the two.

The algorithm consists of two stages. The first stage, greedy placement of anchor nodes, has extremely short computation times and low overhead in terms of the number of anchors placed with respect to the optimum, for the problem instances of reasonable sizes, for which we were able to compare with the ILP based solutions. The second stage, pruning, performs the refinement of a solution to anchor nodes placement optimization problem obtained in the greedy placement stage. The greedy placement algorithm works as follows. We define a score function $S : C \rightarrow \mathbb{R}^+$, that assigns a score to each candidate position. We start from an empty set P. In each iteration, the element with the highest score from C/P is added to P, that is, we place an anchor at the candidate position with the highest score. The scores are updated, and the procedure repeated until P is the solution to POL/FPOL problem, in other words, until all possible target positions are covered. We say that a target position is covered if it is monitored in case of the POL problem, or 2-monitored in case of the FPOL problem. Choosing a good score function is crucial for the algorithm performance. Our score function, for each candidate position is a sum over all target positions of contributions that depend on two factors. The first factor quantifies how much placing an anchor at the candidate position will contribute to covering the target.

It is zero if placing an anchor at the candidate position will not affect covering the target at all, one if it is necessary for covering the target and otherwise, it is in between. The other factor quantifies to which extent the target in question is already covered. Targets that are less covered so far are given advantage.

Pruning consists of any number of independently executed levels. At each pruning level l, $l = 1, 2, 3...$ combinations of l nodes that can be replaced by $l-1$ nodes, so that all possible target positions still remain covered, are identified. In this way, the number of anchors in the solution is reduced. Each subsequent level, as l increases, is more complex and requires more time. The number of pruning levels can be selected to achieve the desired trade-off between the solution quality in terms of the number of nodes placed and execution time.

In Section 6.7, we will show by simulation experiments how the algorithm scales and that by using pruning, it is possible to get the results same or very close to that of the ILP algorithm with shorter execution times, for the same problem instances. For problems of large sizes, run-times can be reduced by decreasing the number of pruning levels, while keeping the quality of results in terms of the number of anchors placed much better than with greedy placement only.

6.3.1 Greedy Placement

To define the score function for greedy placement, we first introduce some definitions. We say that a subset of candidate positions $W \subset C$ covers the target t_u if $U(W, t_u) < U^*$, in case of POL problem, and in case of FPOL, if for all sets W^-, obtained by removing any one element from W, $U(W^-, t_u) < U^*$. We denote this as the binary relation $W \Lambda t_u$. We use the variable K to denote the minimum number of anchors that are needed to cover one target: $K = 3$ for POL, $K = 4$ for FPOL. Given the set P of candidate positions selected for placing anchors, the coverage of target $t_u \in T$, denoted as $\text{cov}(t_u)$ is the maximum cardinality of set $W \subset P$ such that for some candidate positions $a_{i_1}...a_{i_K} \in C$, $\{a_{i_1}...a_{i_K}\}$ covers t_u and W is a subset of the K-element set $\{a_{i_1}...a_{i_K}\}$:

$$\text{cov}(t_u) = \max_{\substack{\exists\{a_{i_1}...a_{i_K}\}\subset C \\ \{a_{i_1}...a_{i_K}\}\Lambda t_u \\ W\subset\{a_{i_1}...a_{i_K}\}}} |W|. \tag{6.6}$$

Such a K-tuple of candidate positions $\{a_{i_1}...a_{i_K}\}$ is then called the maximum K-tuple for the target position t_u. If a candidate position from C/P which is in a maximum K-tuple for a target is added to P, the coverage of this target will increase. Coverage of a target is an integer between 0 and K and it indicates the

maximum number of anchors that are already placed out of any K-element set of anchors that can cover the target.

We define the function $p(a_i, t_u)$ as the number of maximum K-tuples for t_u that a_i belongs to, divided by the total number of maximum K-tuples for t_u. This function quantifies how important a_i is for covering t_u. If $p(a_i, t_u) = 1$, then adding a_i to P is necessary for covering t_u, and if $p(a_i, t_u) = 0$, then adding a_i to P will not contribute to covering t_u at all. The score for each candidate position $S(a_i)$ is then calculated as:

$$S(a_i) = \sum_{\substack{u \\ \text{cov}(t_u)<K}} p(a_i, t_u)^{c_1} * (c_2 - \text{cov}(t_u)). \tag{6.7}$$

There is one element in the sum for each possible target position whose coverage will increase if a_i is added to P. Target positions with lower coverage should be given preference, which is reflected by the factor $(c_2 - \text{cov}(t_u))$.

The parameter c_1 has to be greater than 0 and it determines the relative influence of each of the two factors in the score function. Bigger the value of c_1, more influence on the cost function will $p(a_i, t_u)$ have with respect to current coverage of a target. The parameter c_2 reflects the extent to which the targets with low coverage are favored with respect to targets with high coverage. If c_2 is very close to $K-1$, the target with coverage $K-1$ will account for very little with respect to a target with current coverage 0, while as c_2 increases, the difference becomes less important. The parameter c_2 has to be greater than $K-1$, as for a target t_u that is not already covered the value of $\text{cov}(t_u)$ is between 0 and $K-1$, and all members of the sum in Equation 6.7 have to be positive.

The parameters c_1 and c_2 are determined by simulation experiments. We run the greedy algorithm for different values of c_1 and c_2. We repeat for different randomly generated problem instances, and for each pair of parameter values (c_1, c_2) we look at the average number of anchors placed over all instances. The values that provide the best performance are $c_1 = 2.15$, $c_2 = 2.35$ for localization with no fault tolerance, and $c_1 = 0.85$, $c_2 = 3.55$ for fault-tolerant localization. These values are used in all our simulation experiments in Section 6.7.

The greedy algorithm with pruning is given by Algorithm 2. First we calculate the scores for all the candidate positions (Lines 1-3). In Lines 4-6, we initialize all targets to not covered. In Lines 7-16, we add one candidate at the time until all targets are covered. This proceeds as follows: we select a candidate a_i with the highest score where an anchor is not already placed (Line 8), place an anchor there (Line 9), and recalculate scores for all possibly affected candidates that are not already placed (Lines 10-12). The candidates a_j whose

Algorithm 2: The Greedy Placement with Pruning.

Data: C-candidates, T-targets, $(\forall t_u \in T)$: $\{W \subset C \mid W \Lambda t_u\}$,
L - number of pruning levels
Result: $P \subset C$: candidates with anchors placed

```
1  for (a_i ∈ C) do
2  |  S(a_i) ← calculate;
3  end
4  for (t_u ∈ T) do
5  |  covered(t_u) ← false;
6  end
7  while (∃t_u ∈ T | not covered(t_u)) do
8  |  select a_i ∈ C/P - with highest score;
9  |  add a_i to P ;
10 |  for (t_u ∈ T | (∃W) a_i ∈ W, W Λ t_u) do
11 |  |  (∀a_j | (∃W') a_i ∈ W', W' Λ t_u) S(a_j) ← update;
12 |  end
13 |  for (t_u ∈ T | (∃W), a_i ∈ W, W Λ t_u) do
14 |  |  covered(t_u) ← update;
15 |  end
16 end
17 for l ← 1 to L do
18 |  Prune(P, l);
19 end
```

score functions may be affected by placing the anchor a_i are those that cover any target that anchor a_i also covers. In Lines 13-15 we check if any targets that may be affected by adding anchor a_i are now covered, and update this information. In Lines 17-19, we perform L levels of pruning.

Complexity of Greedy Placement

To calculate the complexity of greedy placement, we denote the total number of candidates $|C| = c$ and the total number of targets $|T| = t$. We denote the maximum number of candidate positions that can reach a target, meaning the distance between them is less than the node range and there are no obstacles in between as d_c. This value depends on maximum density of candidate positions and node range. Similarly, we denote the maximum number of possible target positions that can reach a candidate, depending on the maximum density

of possible target positions, as d_t.

The loop in Lines 1-3 is repeated c times. Calculating $p(a_i, t_u)$ takes $O(d_c^K)$ time, since it requires iterating over all K-tuples of candidate positions that can cover each target, and the number of such K-tuples is in the order of $\binom{d_c}{K}$. The sum in Equation 6.7 has $O(d_t)$ elements which means that the code 1-3 can be executed in $O(cd_c^K d_t)$. The loop in Lines 4-6 is executed t times. In loop in Lines 7-16, one anchor is added in each iteration, therefore, the maximum number of iterations is the number of candidate positions $O(c)$. The loop in Lines 10-12 is executed $O(d_t)$ times, once for each target the added anchor node covers. The pseudo-code in Line 11 can be executed in $O(d_c^K)$ times, as it takes one iteration through all the K-tuples that cover the target t_j. This gives us the execution time of the Greedy placement, without the pruning part $O(cd_t d_c^K + t)$.

6.3.2 Pruning

Pruning consists of any number of independently executed levels. At each pruning level l, $l = 1, 2, 3...$ combinations of l nodes that can be replaced by $l-1$ nodes, so that all possible target positions still remain covered, are identified. Pruning stage at level l can be executed in $O(t\binom{p}{l}\binom{c}{l-1}) = O(tc^{2l-1})$, where p is the number of anchors placed in the greedy placement stage and c is the total number of candidate positions. We used $p = O(c)$. For the total of L pruning stages the complexity is $O(tc^{2L-1})$, which is polynomial in the number of target and candidate positions, although it grows quickly with the number of pruning levels L. In Section 6.4 we will present an improved pruning algorithm that decreases this complexity.

Our implementation of the pruning procedure is given in Procedure Prune. The algorithm takes P, the set of anchor nodes selected by the greedy placement algorithm and lower pruning levels, and identifies all subsets of P of l nodes, that can be replaced by at most $l-1$ nodes, so that all targets remain covered.

We denote by:

- P - set of all selected candidate positions where anchors are placed. P is a solution to POL/FPOL.
- $R(I)$ - a set of $replacements$ for anchors I where $I \subset P$. $R(I)$ contains all sets of at most $l-1$ not selected (not in P) candidate positions that can replace all anchors in I. A set of candidate positions $S \in R(I)$, if after removing all anchors in I and adding to P all anchors in S, all targets are still covered.

Procedure Prune(P, l)

Input: P - set of selected candidates, l - level of pruning
Output: new set of selected candidates

```
1  foreach a_j ∈ P do
2  |  R(a_j) ← calculate;
3  end
4  for i ← 2 to l do
5  |  foreach I ⊂ P, |I| = i do
6  |  |  R_0(I) ← calculate;
7  |  |  R(I) ← R_0(I) ∩ (⋂_{|J|=|I|−1}^{J⊂I} R(J));
8  |  end
9  end
10 if |R(I)| > 0, |I| = l then
11 |  P ← Select and replace sets of l anchors by l − 1 candidates.
12 end
```

- We will also use notation $R(a_i)$ for a set of sets that replaces a single anchor a_i.

- $R_0(I)$ - a set of sets of at most $l-1$ not selected candidate positions that can replace all anchors in $I \subset P$ at positions they all reach. A set of candidate positions $S \in R_0(I)$ if after removing all anchors in I and adding to P all anchors in S, all targets that are reached by all anchors in I are still covered.

The pruning algorithm can be implemented based on the following recursive relation:

$$R(I) = R_0(I) \cap \Big(\bigcap_{\substack{J \subset I \\ |J|=|I|-1}} R(J) \Big). \tag{6.8}$$

A set of candidate positions S can replace a set of anchors I if it can replace all anchors in I at positions they all reach ($R_0(I)$) and it can replace all set of anchors J where J is subset of I. Because sets of candidates that replace J also replace subsets of J it is enough to only consider J of size by 1 smaller than I.

The Pruning Procedure at level l consists of l steps (see Procedure Prune). In each step i, $i = 1...l$, $R(I)$ is identified for all subsets I of P of size i. Replacements for single anchors ($i = 1$) are computed in Lines 1-3. In Lines 4-9 replacements are determined for sets of sizes 2...l by using the relation given by Equation 6.8.

In Lines 10-12, among all subsets of P of l elements that can be replaced, a number of subsets is selected and replaced.

For higher l, most of the sets $R(J)$ (see Equation 6.8) will be empty, and intersection will not have to be computed.

6.4 Greedy Placement with Iterative Pruning

We propose an improved version of greedy placement with pruning algorithm that shortens execution times. We call it greedy placement with iterative pruning algorithm. The greedy placement stage is the same as described in Section 6.3 and the pruning stage is significantly improved. This algorithm will be particularly useful in 3D, due to intrinsically larger problem sizes.

The main improvement compared to the algorithm described in Section 6.3 is that, rather than trying to replace l anchors by $l-1$, where $l = 2,3...L$ we try to replace two anchors by one multiple times, so that the complexity does not increase with the number of attempts to reduce the number of anchors. In between the attempts, we replace some anchors by other anchors, so that all the targets are still covered, while the total number of anchors remains the same. This allows us to attempt the reduction again. This phase is called shuffling. The

Procedure PruneIterative(P)

Input: P - set of selected candidates
Output: new set of selected candidates

```
1 for i ← 1 to L do
2 |   Prune(P, 2);
3 |   Shuffle(P);
4 end
```

algorithm is given in Procedure PruneIterative. L is the number of iterations that can be large, we use 100 in our experiments. Iterative pruning consists of two phases: reduction (Line 2) and shuffling (Line 3). Reduction and shuffling are performed interchangeably until shuffling gives a configuration that has already appeared before, or until the given maximum number of iterations is reached. In the reduction phase, combinations of two nodes that can be replaced by one node, so that all the targets still remain covered, are identified and the replacements are performed. In this way, we attempt to reduce the number of anchors in the solution. This phase is identical to pruning procedure (Procedure Prune in Section 6.3) for $l = 2$.

Shuffling is performed as follows, where C is the set of all candidate positions, and P is the set of candidate positions where anchors are already placed. All pairs $r = (c, p)$ where $c \in C$, $p \in P \setminus C$ are identified, such that if c is replaced by p, all targets remain covered. We will call these pairs possible replacements. Then, possible replacements are selected and performed in random order. Randomization is used to provide different configurations at each shuffling step. Performing some replacements will make others impossible. Therefore, as the result, a random subset of possible replacements will be performed.

If the number of candidate positions is c, the number of target positions is t and L is the number of pruning iterations, then the complexity of the reduction stage is $O(tc^3)$, while the complexity of shuffling is $O(tc^2)$, which gives us total complexity of iterative pruning procedure of $O(Ltc^3)$.

This algorithm is an improvement of the pruning algorithm we proposed in Section 6.3 where we attempted to replace l nodes by $l-1$ nodes at each step, for $l = 2, 3...L$, resulting in complexity of $O(tc^{2L-1})$. In the new algorithm, by attempting to replace two nodes by one at a time, in each iteration, we keep the algorithm complexity low, while the shuffling stage allows to attempt the reduction multiple times.

6.5 Preprocessing

Before running optimization, preprocessing has to be executed. Preprocessing takes as input the floor plan, consisting of room dimensions, coordinates of obstacles, coordinates of target positions and candidate positions. For any of the optimization methods we have described, this block has to produce, for each target position, the set of all triples/quadruples that cover it, and for the optimization methods where this is required, also a set of good pairs: the pairs of anchors that form an adequate angle with the target. The optimization algorithms are then designed to select one of the triples/quadruples that cover each target, while minimizing the total number of anchors selected.

It is important to notice that, if two targets are covered by the exact same sets of triples/quadruples, one of those targets is redundant and can be eliminated. Moreover, if the set of triples/quadruples that cover target t_j is a superset of the set of triples/quadruples that cover target t_i, the target t_j can be eliminated, as one of the triples/quadruples that cover t_j will have to be selected in order to cover t_i. Preprocessing can reduce the number of targets, and thus the problem size significantly, especially if the number of target positions is large compared to the number of candidate positions, as in that case there will be many targets

that are covered by exact same tuples of candidate positions. Intuitively, if two targets are very close to one another, conditions to cover them are likely to be the same. Preprocessing algorithm needs to detect such cases in order to reduce the number of targets, while not reducing coverage in any way. Algorithm 3 gives the preprocessing algorithm for the simpler case where good pairs are not needed.

Algorithm 3: Preprocessing.

Data: C - candidates, $\{(c_x, c_y)\}$-candidate coordinates, T_0 - targets, $\{(t_x, t_y)\}$-target coordinates, $\{(o_x, o_y)\}$ - obstacle coordinates
Result: C-candidates, $T \subset T_0$, $(\forall t_u \in T)$: $K(t_u) = \{W \subset C \mid W \Lambda t_u\}$, $(\nexists t_u, t_v \in T_0) \mid K(t_u) \subseteq K(t_v)$, $(\forall t_u \in T_0 \setminus T), (\exists t_v \in T) \mid K(t_v) \subseteq K(t_u)$

1 **for** $(t_i \in T_0)$ **do**
2 | $R(t_i) \leftarrow calculate$;
3 **end**
4 **while** T_0 **not** *empty* **do**
5 | select $t_i \in T \setminus T_0$ - with lowest $|R(t_i)|$;
6 | $K(t_i) = \{W\} \leftarrow calculate$;
7 | remove all $t_{j \neq i} \in T_0$, such that $(\forall W \in K(t_i) \mid W \Lambda t_j)$;
8 | $T = T \cup \{t_i\}$;
9 **end**

Input is the set of candidates and their coordinates, targets T_0 with their coordinates, and obstacle coordinates. The result is a minimal subset of targets $T \subset T_0$ such that if all targets in T are covered, all targets in T_0 will be covered, as well as for each $t_u \in T_0$ a set of all triples/quadruples that cover t_u: $K(t_u)$. In Lines 1-3 we calculate for each target, all candidate positions that reach it. In Lines 4-10, we select targets from T_0 one by one, in order of the number of candidates that reach them (Line 5). We calculate all tuples that cover the selected target (Line 6). In Line 7 we remove all targets that will also be covered if the selected target is : all targets t_j such that if a tuple W covers the selected target t_i it also covers t_j. In line 8 we add t_i to T.

6.6 The Simulator

For testing our approaches, we designed and implemented a simulator. The simulator block diagram is given in Figure 6.5. Our simulator can work with either square or triangular grid in 2D. It either accepts an existing floor plan as input, or

generates the floor plan randomly with given parameters. Random generation of floor plan is useful for testing and comparing proposed approaches. *Instance Generator* block takes as input the problem parameters: percentage of obstacles, number of candidate positions where anchors can be placed, the room size and the grid resolution. Instance Generator then generates a specific problem instance by placing the given number of obstacles and candidate positions at random points on the grid. The problem instance is then fed into *Preprocessing* and from there into *Optimization* block. Alternatively, the floor plan is fed directly to the preprocessing block, allowing to tackle the real-life problems. The optimization block executes either ILP optimization programs or greedy placement with pruning heuristic, and gives the solution in terms of positions where anchors should be placed. The code is implemented in C++.

6.7 Evaluation

In this subsection we test and compare ILP formulations and greedy placement with pruning for both placement optimization for localization and fault-tolerant placement optimization for localization problem.

6.7.1 The Simulation Setup

Experiments in this section, unless if otherwise stated, are run on Intel(R) Core(TM) i5-3470S 2.90GHz processor. We use the node ranges of 20 m, which is the typical range for ultra wide band system and a triangular grid. We limit $GDoP$ to $2 \cdot GDoP_{min}$ which corresponds to $\theta^* = 37^0$.

In our simulation experiments, all grid points where there are no obstacles are possible target positions. We average all our simulation results in this section, for both times and numbers of anchors, over 100 different test instances. This is done as follows. We run the simulation 100 times, each time with the same parameters: percentage of obstacles and number of candidate positions. However, the problem instance will be different each time, as simulator places the candidate positions and obstacles randomly. Therefore, for each of the 100 test instances, we will get different numbers of anchors placed and execution times.

The problem parameters are chosen in such a way that the problem size is large enough to allow to clearly see the difference between the different approaches, and still small enough to allow executing a large number of simulation experiments (100) in a not too long period of time (less than a week for the

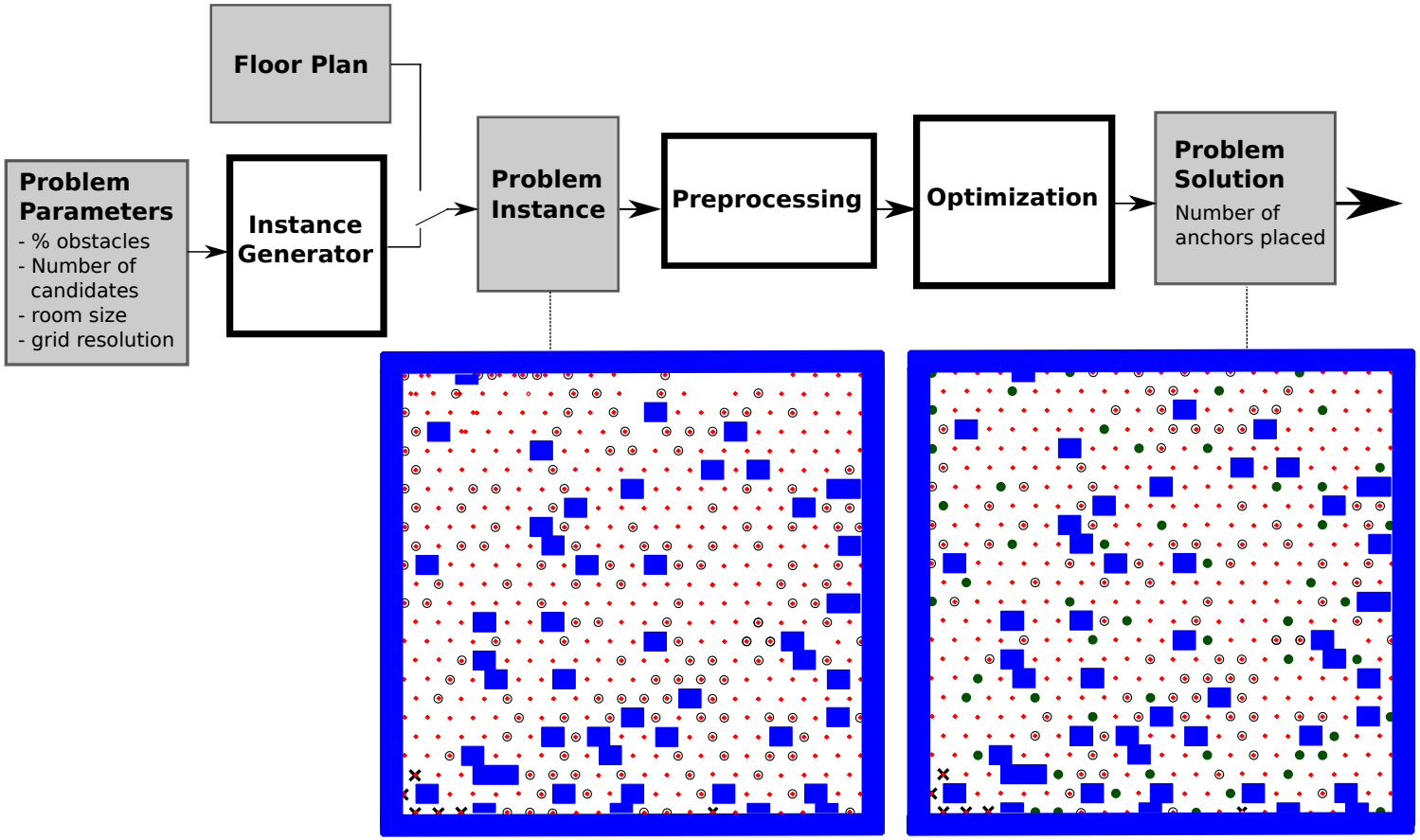


Figure 6.5. The simulation setup. The 2D problem instance contains exact positions of obstacles represented as large squares (in blue), the candidate positions represented as empty circles (in black) and possible target positions represented as small diamond shapes (in red). The positions where anchors should be placed are shown as green filled circles in the solution block. The target positions that cannot be covered even if anchors are placed at all candidate positions are marked by crosses.

slowest solution we have tested). For example, see Table 6.1, where we started with 130 candidate positions.

Since the candidate positions and obstacles are placed randomly, there may be some target positions that cannot be covered, even if anchors are placed at all candidate positions. This happens more frequently as the number of obstacles increases, as those points that cannot be covered are mostly behind obstacles. When this is the case, we simply remove those target positions from the set of target positions, and do not cover them. However, when running simulation experiments, we always choose the input parameters in such a way, that the number of target positions that cannot be covered is below 5 % of the total number of target positions.

6.7.2 Simulation Results for Placement Optimization for Localization

We first test the ILP1 and ILP2 formulation and the greedy placement with pruning heuristic applied to POL problem. In the second simulation experiment, we test how these three approaches behave with increasing problem size. Then we test the behavior of ILP2 and heuristic with changing grid resolution. Next we evaluate how much the more conservative condition for GDoP to be limited from above, derived in Section 6.1.2 and implemented by ILP2, influences the number of nodes in the solution. Finally, we compare the performances of ILP1, ILP2 and ILP3 in terms of execution time and test the improved heuristic: greedy placement with iterative pruning.

Comparison of Integer Linear Programming and Greedy Placement with Different Levels of Pruning

In the first simulation experiment, we fix the size of the area of interest to $100\,m \times 100\,m$ and the resolution of the triangular grid to $5m$. We run the test for 0 %, 10 %, 20 % and 30 % of vertices occupied by obstacles. Increasing the number of obstacles makes the search space smaller, therefore, solutions can be found in much shorter times. Also, increasing the number of obstacles increases the number of target positions that cannot be covered, as discussed in Section 6.7.1. To increase the search space and thus execution times and have a better comparison of the approaches, as well as to keep the number of uncovered target positions low, we increase the number of candidate positions as we increase the percentage of space occupied by obstacles.

Table 6.1. Experimental results - placement optimization for localization.

Obstacles [%]	Candidate positions	Time avg/std [s]							
		ILP 1	ILP 2	Greedy	Greedy +1 prune	Greedy +2 prune	Greedy +3 prune	Greedy +4 prune	Greedy +5 prune
0	130	4864/10045	805/1657	0.02/0.01	0.02/0.003	0.13/0.18	0.35/0.20	1.08/0.41	5.19/2.16
10	149	1112/1579	161/193	0.01/0.001	0.02/0.002	0.08/0.08	0.25/0.14	0.70/0.28	3.19/1.54
20	171	421/619	60/76	0.01/0.001	0.01/0.002	0.06/0.04	0.17/0.08	0.50/0.19	2.62/1.17
30	196	188/391	29/46	0.01/0.001	0.01/0.001	0.05/0.04	0.13/0.05	0.45/0.19	3.47/1.63
		Average number of anchors placed		Anchors overhead [%] avg/std					
0	130	41.89	41.89	9.11/2.98	8.13/2.77	5.39/2.31	3.54/1.89	2.59/1.73	1.75/1.68
10	149	54.64	54.64	6.75/2.27	6.00/2.21	3.97/2.12	2.45/1.58	1.77/1.41	1.21/1.32
20	171	68.55	68.55	6.08/2.07	5.41/2.00	3.45/1.51	2.32/1.20	1.55/1.03	1.01/0.85
30	196	81.63	81.63	5.72/1.85	4.97/1.71	2.87/1.42	1.85/1.17	1.28/1.07	0.86/0.91

We compare ILP1, ILP2 and greedy placement with different levels of pruning. these simulation experiments do not include ILP3,as we only included ILP3 in the later stages of our work. We will present the comparison of ILP3 with other methods later in this section.

For this simulation experiment, we want to use equivalent conditions for the anchors to cover a target for both ILP1 and ILP2, as well as the greedy placement with pruning heuristic, so we can compare their speeds for the same problem complexity. Therefore, for this experiment, we replace the condition $U(a_i, a_j, a_k, t_u) > U^*$ in Equation 6.1e for ILP1, by the condition that at least one of the angles formed by the candidate positions a_i, a_j, a_k with the target t_u has to be adequate, all three anchor positions reach the target and they are not collinear. We do similarly for the heuristic. We will evaluate later in other experiments, in Sections "Influence of the GDoP Condition" and "Comparison of ILP1, ILP2, ILP3 and Greedy Algorithm with Iterative Pruning", how using the more conservative expression for GDoP to be limited, formulated in Section 6.1.2 influences the result in terms of the number of anchors placed.

The comparison of ILP1, ILP2 and the greedy heuristic with different levels of pruning is given in Table 6.1. The upper part of the table gives information on execution times. For each combination of the total percentage of obstacles and the number of candidate positions generated, it gives the average and the standard deviation of the execution times for both ILPs and greedy heuristics with different numbers of pruning levels. The lower part of the table gives information on the number of anchors placed. For each combination of parameters: total percentage of obstacles and the number of candidate positions generated, the average number of anchors placed for ILPs and the average and standard deviation of overhead in terms of the number of anchors placed for the greedy heuristics with different numbers of pruning levels are given. The numbers of candidate positions are chosen in such a way that they are high enough for realistic scenarios in a factory environment.

The results in Table 6.1 show that the ILP2 has significantly lower execution times than ILP1. As discussed in Section 6.1, we believe this is primarily due to the lower number of variables used. For the problem size we used, the ILP2 execution time is about 4 hours at most.

Our simulation experiments show that the greedy heuristic with pruning has very low execution times while the number of anchors placed is within one or two percent from the optimum for five pruning levels. As expected, by increasing the number of pruning levels, the number of placed anchors gets closer to optimal, and the execution time increases. With only three pruning levels, and the average execution time of under $0.5\,s$, we have the overhead in the number of anchors of less than 4% on the average.

Behavior with Increasing Problem Size

In the second simulation experiment, to test the scalability of our solutions, we look at how the execution times increases when increasing the size of the area of interest. The area is of square shape, with 20 % of grid points occupied by obstacles, the grid resolution is $5m$ and the number of candidate positions increases proportionally to the area size. Results are again averaged over 100 randomly generated instances for each area size.

Figure 6.6 shows the results for ILP1, ILP2 and greedy placement with different levels of pruning on one graph for comparison. On the scale used in this graph, all five lines for the greedy heuristic with 0 to 4 levels of pruning overlap, therefore, they are shown as one line only. ILP1 and ILP2 formulations have only been tested up to the area of $14900\,m^2$ and $19900\,m^2$ respectively, because for larger sizes, the time required to execute the simulations would be prohibitive. As we can see, the difference between ILP1 and ILP2 execution times becomes much larger with larger problem sizes. For the area of size $120\,m \times 120\,m$, 230 candidate positions and 20 % of obstacles, ILP2, on the average, executes in 24 times shorter time than ILP1. We can also see that greedy placement with pruning has much lower computing time, and scales better than ILP based solutions.

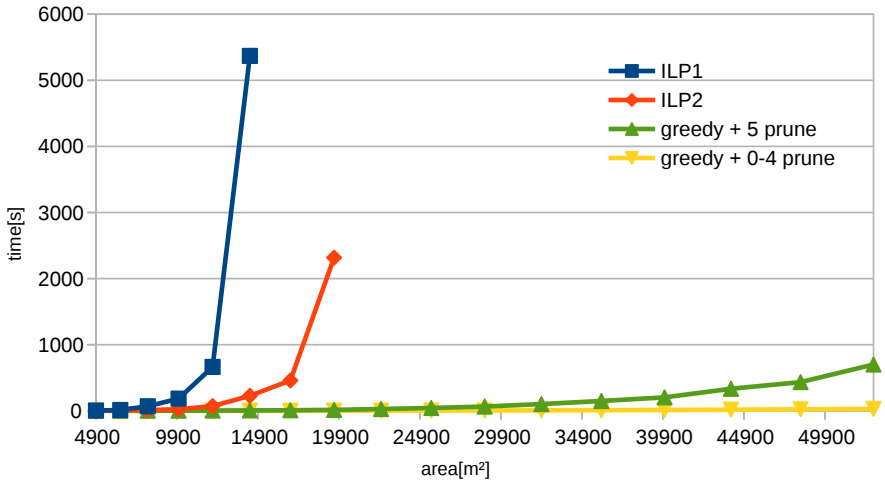


Figure 6.6. Execution time of two integer linear programs and the greedy placement with pruning algorithm depending on the size of the area of interest, with 20 % of area occupied by obstacles. Results are averaged over 100 instances. The number of candidate positions increases proportionally to the area.

Changing Grid Resolution

Distance measurement precision and localization precision do not directly depend on grid resolution and are not limited by the grid edge. Our solutions guarantee that the target will be localizable at grid vertices and we implicitly assume that if a target is localizable at grid vertices, it is also localizable in between them. This will be the case if the grid resolution is high enough with respect to node range, so that the bearing angles do not significantly change within a single grid cell. If this condition is not met, the localization precision far from the grid vertices could be decreased. Therefore, low grid resolution may still affect localization precision indirectly.

In the previous simulation experiment in this section, we set the grid edge to $5\,m$. Figure 6.7 shows how the ILP2 and greedy heuristic with 5 pruning levels behave with increase of the grid resolution. We tested for 130 candidate positions and no obstacles. ILP2 algorithm takes around an hour and a half for a grid edge of $2.5\,m$, while the greedy heuristic scales well and takes around three minutes for the grid edge of $0.5\,m$. The number of anchors placed also increases with increasing grid resolution: from the average of 42.41 at $5\,m$ resolution, to the average of 63.11 at $0.5\,m$ resolution for greedy placement with 5 pruning levels.

Here we can already see some limitations of the greedy placement with pruning algorithm. It slows down as the grid resolution increases. This is one of the reasons that we proposed another heuristic algorithm, greedy placement with iterative pruning, that scales much better.

Influence of the GDoP Condition

For the rest of Section 6.7.2, our simulations are run on a different processor: Intel(R) Core(TM) i7-7500U 2.70GHz.

Now we evaluate by simulation, how much using the more conservative condition for GDoP to be limited from above, derived in Section 6.1.2, influences the number of anchors placed by the ILP. We run a simulation experiment where we compare ILP1 and ILP2 as the number of obstacles increases. ILP1 uses the uncertainty function U given by Equation 3.5, where $GDoP$ is given by Equation 3.4. ILP2 uses the more conservative condition from Section 6.1.2, where at least one bearing angle has to be adequate.

We compare the resulting number of anchors placed by ILP1 and ILP2 for the same problem instances as well as the number of target positions that cannot be covered. In the first experiment, the grid resolution is $5\,m$, the room size is $100 \times 100\,m$, the number of candidate positions respectively 130, 149, 171 and

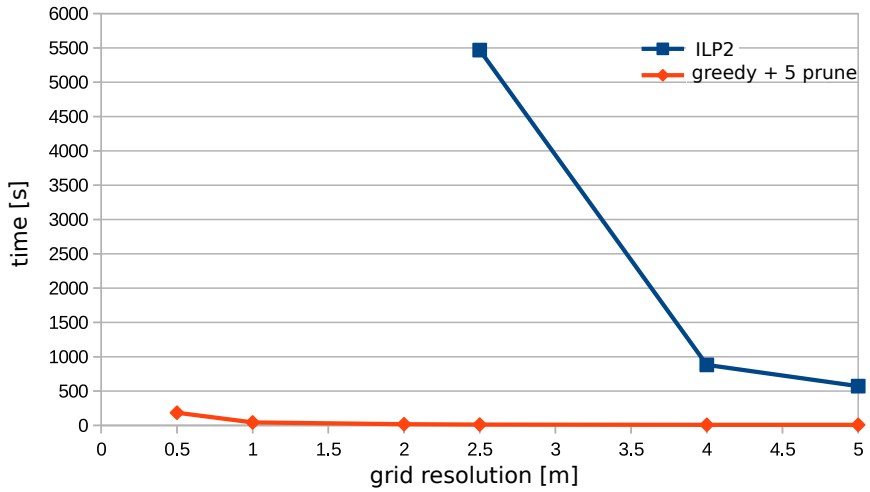


Figure 6.7. Execution time of ILP2 and the greedy placement with 5 pruning levels depending on the grid resolution, in a $100\,m \times 100\,m$ room with no obstacles and 130 candidate positions. Results are averaged over 100 instances.

196, for 0, 0.1, 0.2 and 0.3 percent of the area covered by obstacles. We generate 100 different problem instances for each number of obstacles. Both the resulting number of nodes placed and the number of uncovered positions were the same for each execution, for ILP1 and ILP2. Thus, at $5\,m$ resolution, using a more conservative condition for GDoP to be limited, did not influence the results. This means that the approximate GDoP from Section 6.1.2 is quite close to the exact GDoP.

We run another experiment to compare the number of anchors placed and the target positions uncovered for ILP1 and ILP2 with $2\,m$ resolution and no obstacles. The room size is fixed to $100\,m \times 100\,m$, while the number of candidate positions increases. The results are given in Table 6.2. We can observe that at this resolution there are very small differences in the number of anchors placed and the number of positions uncovered: The difference in the average number of anchors placed is below 1 % and the difference in the number of positions that cannot be covered below 0.1 % of total area. This confirms that the GDoP representation from Section 6.1.2 is close to the exact one, although, as the grid resolution increases, we start to see some difference in the results.

Table 6.2. Experimental results - Comparison of Exact and Approximate GDoP.

Candidate positions	Average number of anchors placed		
	ILP 1	ILP 2	Difference [%]
100	52.32	52.65	0.631
105	51.97	52.43	0.885
110	51.92	52.42	0.963
115	51.43	51.83	0.778
	Average percentage of uncovered positions [%]		
	ILP 1	ILP 2	Difference
100	2.035	2.122	0.087
105	1.473	1.553	0.080
110	1.285	1.359	0.074
115	1.129	1.177	0.048

Comparison of ILP1, ILP2, ILP3 and Greedy Algorithm with Iterative Pruning

As discussed in Section 6.1.4, the advantage of ILP3 is that it has fewer variables than both ILP1 and ILP2 formulation. However, the number of equations is larger and increases with the increasing number of obstacles. In the following simulation experiment we compare ILP1, ILP2 and ILP3 formulation as the number of candidate positions, and thus the complexity, increases. So far we only evaluated the greedy placement with different levels of pruning. In this simulation experiment we evaluate the improved version, the greedy placement with iterative pruning (GIP) algorithm.

We compare the three ILP formulations for 2D placement for localization and the GIP heuristic by randomly generating test instances. We use the triangular grid with $2\,m$ resolution and the room size of $100 \times 100\,m$. For the different percentage of the total area covered by obstacles, ranging from 0 to 20 %, we run experiments for the increasing number of candidate positions. For each combination of parameters, number of obstacles and candidate positions, we average

all our simulation results in this section, for both times and numbers of anchors, over 100 different test instances. All grid points where there are no obstacles are possible target positions.

In Figure 6.8 we show the average execution times as the number of candidate position increases, for an area with no obstacles. Figure 6.9 and Figure 6.10 respectively show the average execution times for 10 % and 20 % of total area covered by obstacles, for the increasing number of candidate positions. Table 6.3 shows the execution times and the number of anchors placed for ILP1, ILP2, ILP3 and the greedy heuristic with iterative pruning. For each percentage of obstacles, the table shows results for the largest number of candidate positions that was generated for all four of the solutions. The left part of the table gives the average and the standard deviation of the execution times for all three ILPs and the GIP heuristic. The right part of the table gives information on the number of anchors placed. Since ILP1 and ILP3 provide optimal solutions in terms of the number of anchors placed, the average number of anchors placed is given for these two approaches. The average and standard deviation of overhead in terms of the number of anchors placed are given for the ILP2 and for the GIP algorithm.

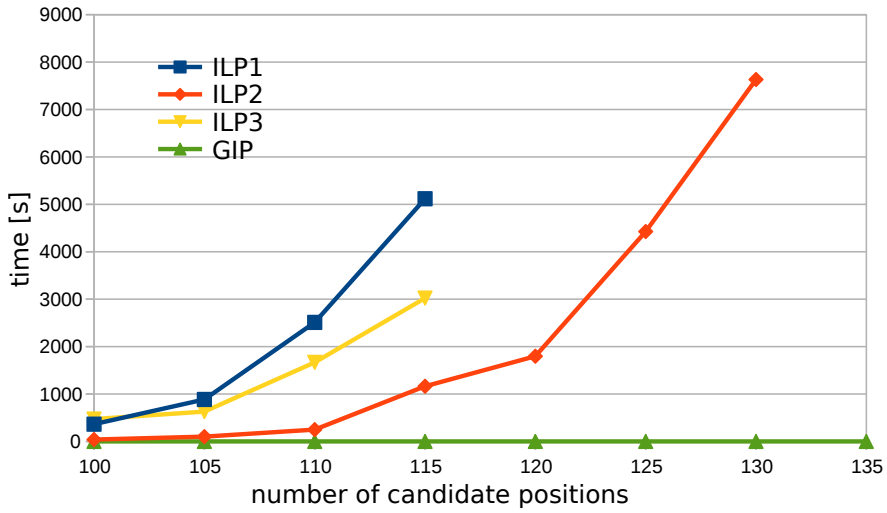


Figure 6.8. Execution times depending on the number of candidates for GIP and ILP for a room without obstacles.

Both ILP1 and LP3 provide optimal results in terms of the number of anchors placed. In Figures 6.8, 6.9 and 6.10, we can see that which one of these formulations has lower execution times will depend on the number of obstacles. For

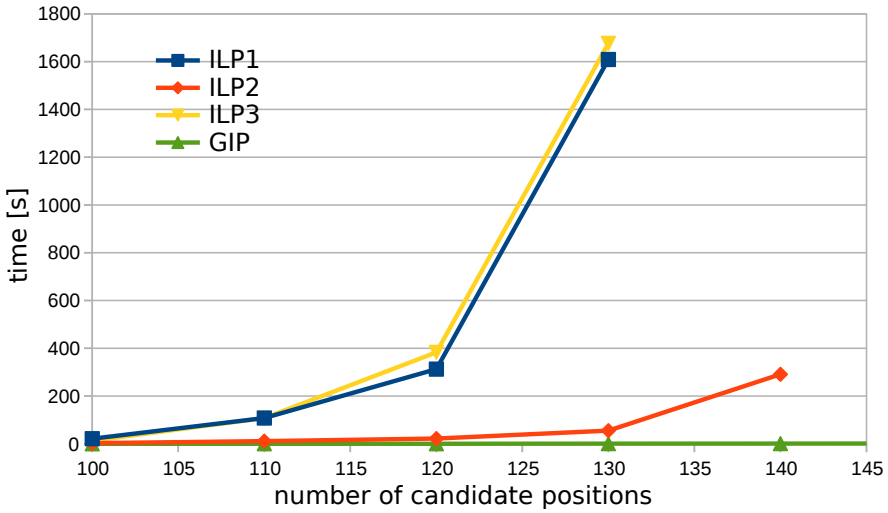


Figure 6.9. Execution times depending on the number of candidates for GIP and ILP for 10 % of area covered by obstacles.

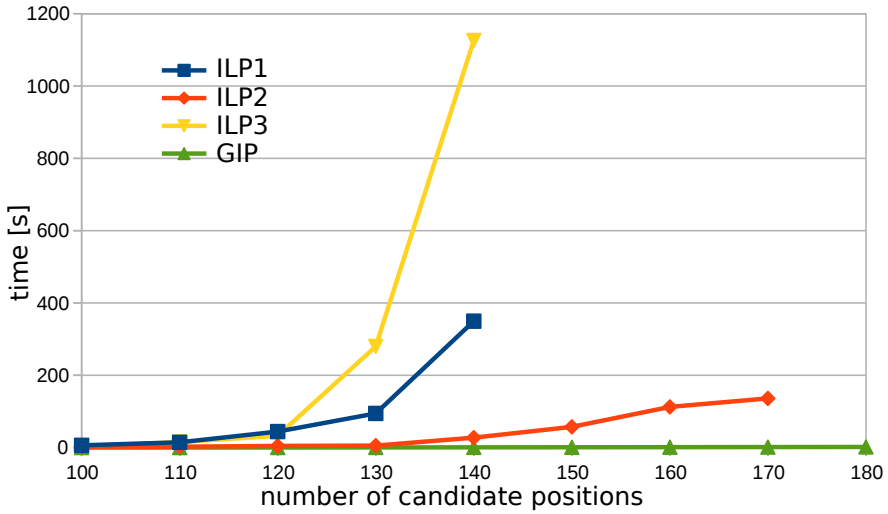


Figure 6.10. Execution times depending on the number of candidates for GIP and ILP for 20 % of area covered by obstacles.

the area with no obstacles (Figure 6.8) ILP3 is faster than ILP1. We assume this is because the number of variables for ILP3 is lower. However, as the number of obstacles increases, the number of equations for ILP3 increases, so for 10 % of

Table 6.3. Experimental results - placement optimization for localization.

Obstacles [%]	Candidate positions	Time avg/std [s]				Average number of anchors placed		Anchors overhead [%] avg/std	
		ILP 1	ILP 3	ILP2	GIP	ILP 1	ILP 3	ILP2	GIP
0	115	5118/9805	3022/3967	1163/4416	0.57/0.37	51.43	51.43	0.79/1.24	0.94/1.02
10	130	1609/7342	1676/6734	55/125	0.74/0.43	61.3	61.3	0.75/1.79	0.53/0.81
20	140	350/444	1126/3998	27/158	0.66/0.39	69.17	69.17	0.47/1.79	0.27/0.61

area covered by obstacles, these two formulations are close in execution times and for 20 % of area covered by obstacles, ILP1 is faster than ILP3.

ILP2 is significantly faster than both formulations in all cases, which may be a result of the good trade-off between the number of variables and equations. The formulation we use for ILP2 is a little more conservative. We can see in Table 6.3 that average overhead in terms of the number of anchors placed for ILP2 is less than 1 %.

The execution time of the GIP algorithm is very low, as can be seen both in the figures and in the table. The overhead in terms of the number of anchors for GIP algorithm is lower than 1 % on the average for our experiment and similar to that of ILP2. As can also be seen in Table 6.3, standard deviation of execution times for ILP algorithms is rather high. Indeed, depending on a different randomly generated instance, ILP solver may take much longer to solve the problem instance. This is not the case with the GIP algorithm which has low variance of execution times.

6.7.3 Simulation Results for Fault-tolerant Placement Optimization for Localization

Next, we test the ILP formulation and greedy placement with pruning heuristic for fault-tolerant placement optimization for localization and evaluate the execution times for both approaches, as well as the quality of the solutions in terms of the number of anchors placed. Since we only propose one ILP formulation for fault-tolerant placement (see Section 6.2), we simply refer to it as ILP in this section.

Comparison of Integer Linear Programming and Heuristic

We compare the results obtained by ILP to our greedy placement with pruning heuristic with different numbers of pruning levels. We run simulation experi-

ments in the $70\,m \times 70\,m$ square area with no obstacles, where 65 candidate positions are randomly selected, and average results over 100 different test cases, with different randomly selected candidate positions. Table 6.4 shows the results for ILP and greedy heuristic with up to 8 pruning levels. It shows the average number of anchors placed, average and standard deviation of execution time, and average and standard deviation of overhead in the number of anchors placed with respect to the exact solution obtained by ILP. We see that the greedy heuristic with pruning has the number of anchors placed within several percent of the optimum while executing in much shorter time than ILP. We also observe that the standard deviation for ILP is very high compared to the average. Indeed, the optimization time for the ILP solver can be dramatically different for different randomly generated instances, even for the same number of candidate positions.

Figure 6.11 shows the effect of adding more pruning levels in terms of the number of anchors overhead and the execution time. Although the execution times increase when adding pruning levels, this is not a serious limitation as the number of anchors overhead is very low after a small number of levels, and a large number of pruning levels is not necessary. We see that after only four pruning levels the number of anchors overhead drops to 1 % only, and the execution time for this number of pruning levels is about $0.4\,s$ on the average.

Table 6.4. Experimental results for fault-tolerant placement optimization for localization.

	Anchors avg	Time [s] avg/std	Overhead [%] avg/std
ILP	33.8	354/600	0/0
Greedy	35.8	0.01/0.002	5.84/3.49
Greedy + 1 prune	35.3	0.02/0.01	4.53/3.09
Greedy + 2 prune	34.6	0.09/0.20	2.50/2.21
Greedy + 3 prune	34.4	0.18/0.20	1.66/1.75
Greedy + 4 prune	34.2	0.38/0.27	1.08/1.64
Greedy + 5 prune	34.1	0.79/0.44	0.84/1.49
Greedy + 6 prune	34.0	1.91/1.07	0.61/1.36
Greedy + 7 prune	34.0	7.15/6.50	0.46/1.25
Greedy + 8 prune	33.9	40.28/43.98	0.40/1.19

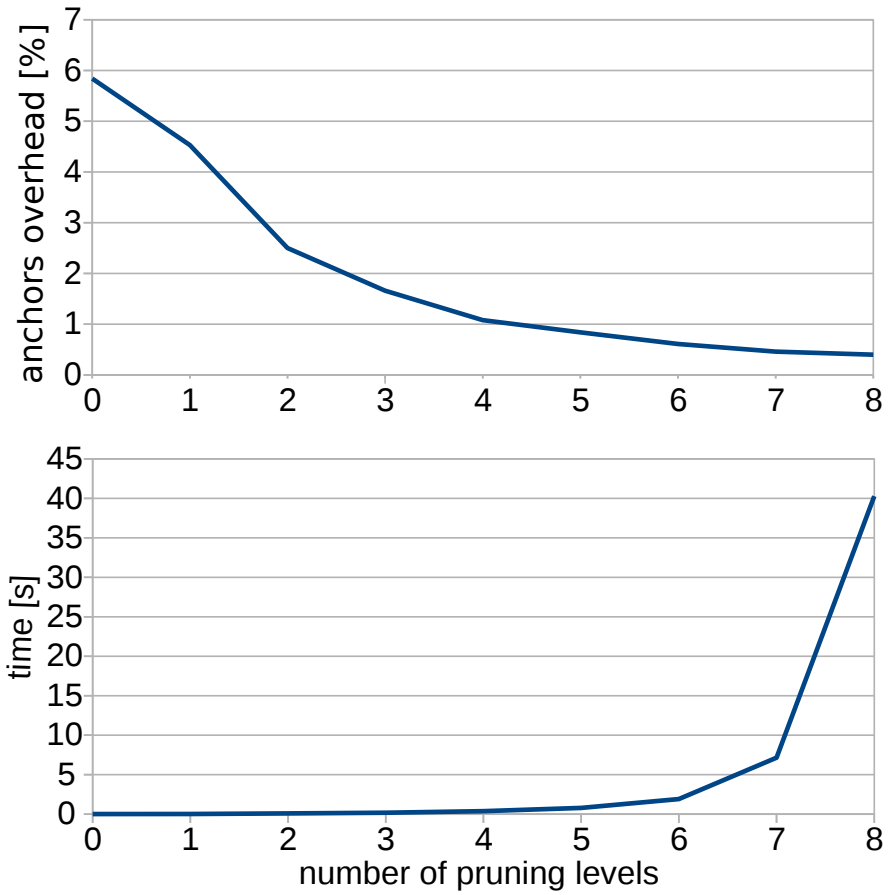


Figure 6.11. Fault-tolerant placement optimization. Overhead in terms of the number of anchors and execution time of the greedy algorithm depending on the number of pruning levels.

Scalability of Heuristic and ILP for Fault-tolerant Placement Optimization

In the next simulation experiment, we show how the execution time of both the greedy heuristic with pruning and ILP increase with the problem size for FLPO problem. We increase the size of the area of interest, with no obstacles, while the number of randomly selected candidate positions increases proportionally. We average the result for each size over 100 randomly generated problem instances. The Figure 6.12 shows the execution times as the problem size increases, for greedy heuristic with different numbers of pruning levels, and for ILP. In Figure 6.13 results for ILP are omitted for better visualization. We can see that in case of heuristic, the execution time is dominated by pruning stage, and increases with

the number of pruning levels. The line for no pruning (0 pruning levels) is not shown on the graph, because it is coincident with the line for 1 pruning level. In Figure 6.12 it is clearly visible that the heuristic, even with the high number of pruning levels, is much more scalable than the ILP, which confirms the usefulness of the greedy placement with pruning heuristic.

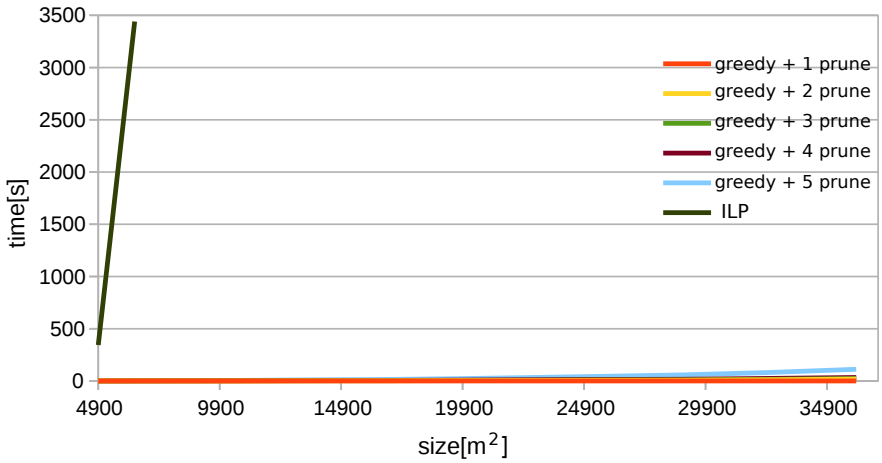


Figure 6.12. Fault-tolerant placement optimization execution times depending on the problem size for ILP and the proposed heuristic.

Evaluating Quality of Greedy Placement

Our heuristic consists of two stages. The greedy placement and multiple pruning levels. It is possible to perform only greedy placement without pruning, while pruning can be performed after any other algorithm that will ensure all targets are 2-covered. The goal of this simulation experiment is to evaluate the quality of our greedy placement algorithm, by comparing it to another, simple placement algorithm, as well as to evaluate the quality of pruning alone, by performing it after this simple algorithm.

We call this simple placement algorithm quasi-random placement. With quasi-random placement, we place one anchor after another, placing a new anchor at a random candidate position, with the only condition being that placing an anchor at this position should increase the coverage of at least one target. We run experiments in the $100\,m \times 100\,m$ square area, with 130 randomly selected candidate positions. We compare our greedy placement algorithm to quasi-random placement, both followed by 6 pruning stages, and average the results over 50 dif-

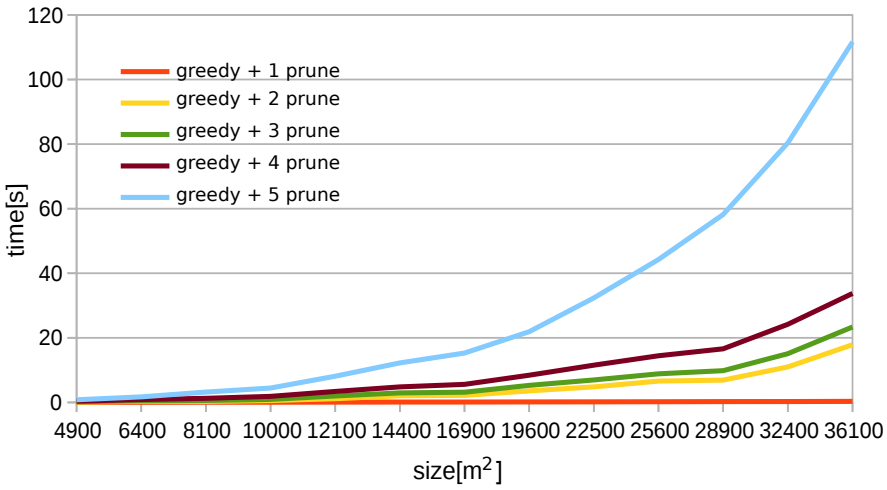


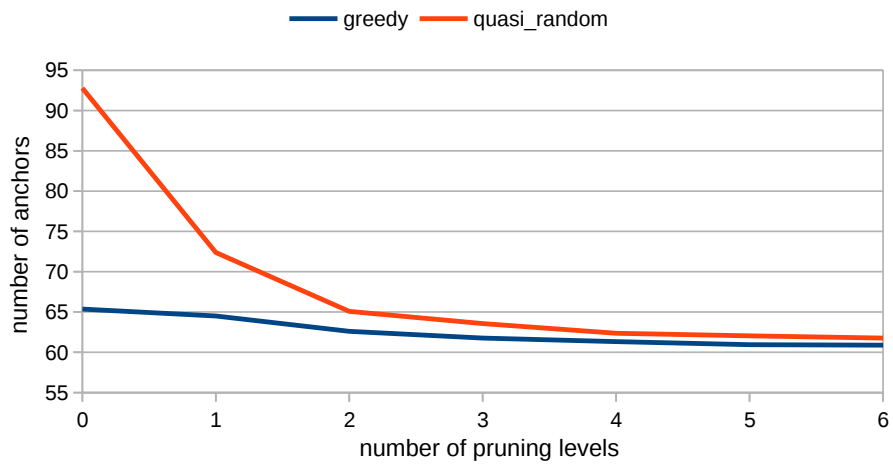
Figure 6.13. Fault-tolerant placement optimization execution times depending on the problem size for greedy heuristic with different numbers of pruning levels.

ferent problem instances, with different randomly selected candidate positions. The results are shown in Figure 6.14(a), where the average number of nodes is shown after each pruning step, and Figure 6.14(b), showing the average number of nodes over time for both algorithms.

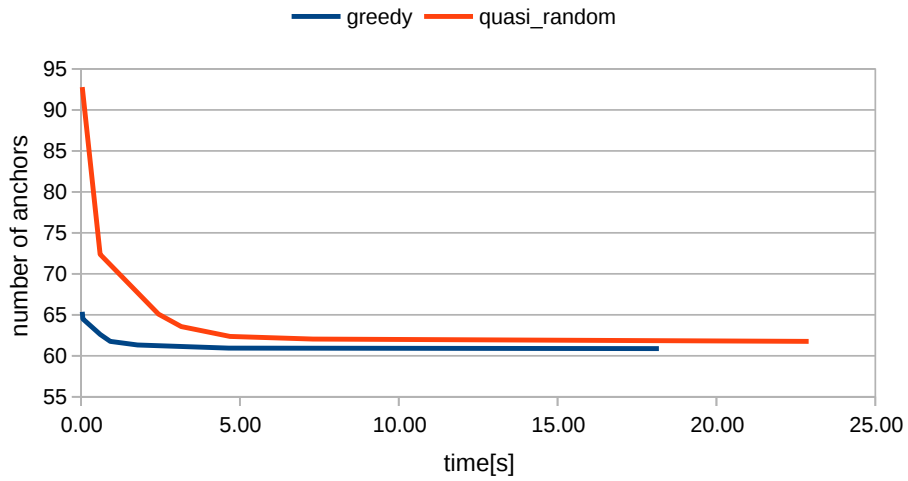
We can see from the initial values in both graphs that our greedy placement is much superior to quasi-random placement. The quasi-random algorithm places over 40 % more nodes. However, after 4 pruning stages, the difference between these two algorithms is about 1 extra anchor, on the average. Pruning is very effective, and can work with any placement algorithm, even random, but it is much more time consuming than the greedy placement. Our greedy placement algorithm has very short execution times, and performs very well also on its own, which can be significant for very large problem instances. We also see that after about 4 stages the number of nodes decreases very little, and judging from our previous experiment, this is probably because the optimum is nearly reached. We can also notice that the slope of decrease in the number of anchors is much steeper at the beginning in Figure 6.14(b), with respect to Figure 6.14(a). This is because lower levels of pruning have very short execution times, while providing excellent benefits in terms of the number of anchors. Pruning allows an easy trade-off between speed and quality of the solution.

For the problem size we use in this simulation experiment, the time require-

ment of ILP would already be prohibitive. This is the reason we do not include the optimal solution in our analysis.



(a)



(b)

Figure 6.14. Comparison of greedy placement and quasi-random placement.

Coverage After Node Failure

Our fault-tolerant approach guarantees that after any one anchor fails all targets will remain covered, in a sense that they can be reliably localized. In the

worst-case scenario, failure of two nodes will already leave at least one target not covered. However, this worst-case scenario is unlikely.

The goal of this simulation experiment is to understand what percentage of targets that will remain not covered, on the average, after a certain percentage of anchors, placed according to our heuristic, fails. In the square area of size $100\,m \times 100\,m$, 130 candidate positions are randomly placed, and our greedy heuristic is used to place the anchor nodes. After that, we remove anchors in random order, and record the percentage of targets that remain covered. The results are averaged over 100 executions. The results are shown in Figure 6.15.

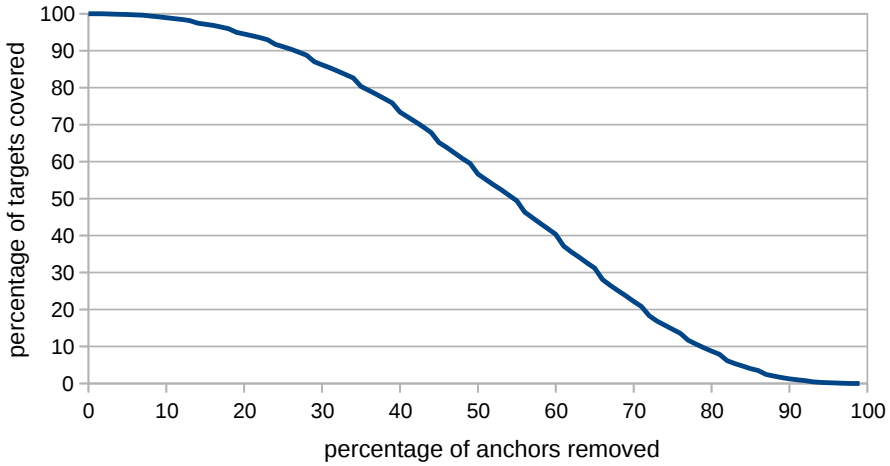


Figure 6.15. Percentage of targets that remain covered after a given percentage of anchors fail.

As expected, when only few anchors are removed the percentage of targets covered decreases very slowly. After 10 % of anchors fail, 99 % of targets are still covered, and after 30 % of anchors are removed more then 85 % of targets are covered. At that point, many targets are already covered without redundancy, therefore, the coverage starts to decrease more quickly as more anchors are removed.

The experiment shows that although 1-fault tolerance only guarantees that one node can fail while leaving all targets covered, on an average case, this percentage is much higher.

Chapter 7

Three-Dimensional Localization

In this chapter we improve on the solution we proposed in Chapter 6 and extend it to 3D scenario. When using the 2D model we assume that the differences in heights of positions at which the anchors can be placed and where the target is moving are small in comparison to the measured distances from anchors to the target and to the maximum localization error allowed. If the target is a person walking on the ground level, and anchors are placed on the walls, this would be the case. Moreover, if the ceiling is not very high and the localization precision of several meters is sufficient for the given application, the precision will still be in the desired range even if the anchors are placed on the ceiling. Such model is sufficient for many applications. However, the application scenario we focus on is safety at work in industrial environments such as factory halls. For those industrial environments where the ceiling is very high and the worker might be climbing or be lifted from the ground, or if very high localization precision is needed, 3D localization may be required. Therefore, in this chapter we focus on 3D localization.

The main challenge when moving from 2D to 3D is increased problem complexity, due to the increased problem size: the number of candidate and target positions. Moreover, four anchors are needed to localize the target instead of three. If distances of target from the anchors are $\{r_i\}_{i=1,2,3,4}$ the target will be located at the intersection of the spheres with the centers at the anchors and the radii $\{r_i\}_{i=1,2,3,4}$. As three spheres intersect at two different points, the fourth anchor is required to distinguish between these two and determine the target position. Therefore, each target position needs to be reached by at least four anchor nodes, to be covered. The four anchors that cover a target must not be coplanar. This is because, in the ideal case where no distance measurement errors exist, the four spheres around coplanar nodes will intersect at two points,

and it will not be possible to distinguish at which of the two the target is.

In this chapter, we propose an ILP formulation which enables to obtain exact solutions for problem instances of reasonable sizes in 3D scenario. We will show that the greedy placement with iterative pruning algorithm we proposed in Section 6.4 is able to handle problems of large sizes that appear in 3D. While GDoP expression for 2D multilateration can be found in literature, in this chapter we derive the expression for GDoP for 3D multilateration system.

7.1 Geometric Dilution of Precision

As for the 2D case, we use GDoP to express precision of localization in 3D. However, the expression from Equation 3.4, that was derived in Spirito [2001], is only valid in 2D. We were not able to find a similar expression for GDoP for 3D multilateration. Therefore, we derive it in this section. We also give a geometric interpretation of the result.

Let (x, y, z) be the target position, (x_i, y_i, z_i) the position of anchor i, and d_i the distance from target to anchor i. Then we have:

$$\sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2} = d_i \tag{7.1}$$

The target coordinates are estimated by using the linear weighted least squares (WLS) algorithm. We assume that an a priory estimate of the target position $(x^{(0)}, y^{(0)}, z^{(0)})$ is available. This estimate is typically the target position at the previous WLS algorithm iteration. Then we can linearize Equation 7.1 as:

$$d_i \simeq d_i^{(0)} - u_{i,x}\Delta x - u_{i,y}\Delta y - u_{i,z}\Delta z, \tag{7.2}$$

where:

$$\Delta x = x - x^{(0)}; \qquad \Delta y = y - y^{(0)}; \qquad \Delta z = z - z^{(0)}; \tag{7.3}$$

$$u_{i,x} = \frac{x_i - x^{(0)}}{d_i^{(0)}}, \qquad u_{i,y} = \frac{y_i - y^{(0)}}{d_i^{(0)}}, \qquad u_{i,z} = \frac{z_i - z^{(0)}}{d_i^{(0)}}, \tag{7.4}$$

and $\hat{u}_i = u_{i,x}\hat{e}_x + u_{i,y}\hat{e}_y + u_{i,z}\hat{e}_z$ is the unit vector originating at the target, directed towards the anchor i. By representing Equations 7.2 in matrix form we get the system $\mathbf{Ax} = \mathbf{b}$, where:

$$\mathbf{A} = [u_{i,x}\ u_{i,y}\ u_{i,z}]_{i\in\mathsf{A}}; \qquad \mathbf{x} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}; \qquad \mathbf{b} = [d_i^0 - d_i]_{i\in\mathsf{A}}, \tag{7.5}$$

where A is the set of anchors.

The WLS solution of the linear problem $\mathbf{Ax} = \mathbf{b}$ that minimizes the scalar cost function $(\mathbf{Ax}-\mathbf{b})^{\mathbf{T}}\mathbf{Q_b}^{-1}(\mathbf{Ax}-\mathbf{b})$ is:

$$\hat{x} = \mathbf{G}^{-1}\mathbf{g} \quad \text{where} \quad \mathbf{G} = \mathbf{A}^T\mathbf{Q_b}^{-1}\mathbf{A}, \quad \mathbf{g} = \mathbf{A}^T\mathbf{Q_b}^{-1}\mathbf{b} \tag{7.6}$$

and $\mathbf{Q_b}$ is the covariance matrix of $\mathbf{b}$. Since we assume that the distance measurement errors are independent and identically distributed with standard deviation σ_D, $\mathbf{Q_b}$ is a diagonal matrix $\mathbf{Q_b} = \text{diag}(\sigma_D^2)$.

GDoP is defined as the square root of the trace of covariance matrix $\mathbf{Q}_{\hat{x}}$ of $\hat{x}$: $GDoP = \sqrt{Tr\{\mathbf{Q}_{\hat{x}}\}}$, where $\mathbf{Q}_{\hat{x}} = E\{\hat{x}\hat{x}^T\}$. From 7.6 we get $\mathbf{Q}_{\hat{x}} = \mathbf{G}^{-1}$. From 7.6 and 7.5 we get:

$$\mathbf{G} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{12} & G_{22} & G_{23} \\ G_{13} & G_{23} & G_{33} \end{bmatrix}, \tag{7.7}$$

with:

$$\begin{aligned} G_{11} &= \sigma_D^{-2}\sum_{i\in\mathsf{A}} u_{ix}^2, & G_{12} &= \sigma_D^{-2}\sum_{i\in\mathsf{A}} u_{ix}u_{iy}, \\ G_{22} &= \sigma_D^{-2}\sum_{i\in\mathsf{A}} u_{iy}^2, & G_{13} &= \sigma_D^{-2}\sum_{i\in\mathsf{A}} u_{ix}u_{iz}, \\ G_{33} &= \sigma_D^{-2}\sum_{i\in\mathsf{A}} u_{iz}^2, & G_{23} &= \sigma_D^{-2}\sum_{i\in\mathsf{A}} u_{iy}u_{iz}. \end{aligned} \tag{7.8}$$

By inverting the matrix $\mathbf{G}$ we get:

$$\mathbf{G}^{-1} = \frac{1}{\det\mathbf{G}} \begin{bmatrix} G_{22}G_{33}-G_{23}^2 & G_{13}G_{23}-G_{12}G_{33} & G_{12}G_{23}-G_{13}G_{22} \\ G_{13}G_{23}-G_{12}G_{33} & G_{11}G_{33}-G_{13}^2 & G_{12}G_{13}-G_{11}G_{23} \\ G_{12}G_{23}-G_{13}G_{22} & G_{12}G_{13}-G_{11}G_{23} & G_{11}G_{22}-G_{12}^2 \end{bmatrix}, \tag{7.9}$$

$$GDoP = \frac{1}{\sigma_D}\sqrt{\frac{G_{22}G_{33}-G_{23}^2+G_{11}G_{33}-G_{13}^2+G_{11}G_{22}-G_{12}^2}{\det\mathbf{G}}}. \tag{7.10}$$

The numerator in Equation 7.10 equals:

$$\begin{aligned}
&G_{22}G_{33} - G_{23}^2 + G_{11}G_{33} - G_{13}^2 + G_{11}G_{22} - G_{12}^2 \\
&= \sigma_D^{-4} \sum_{i,j \in \mathsf{A}} u_{ix}^2 u_{jy}^2 + u_{ix}^2 u_{jz}^2 + u_{iy}^2 u_{jz}^2 - \\
&\qquad - u_{ix}u_{iy}u_{jx}u_{jy} - u_{ix}u_{iz}u_{jx}u_{jz} - u_{iy}u_{iz}u_{jy}u_{jz} \qquad (7.11) \\
&= \sigma_D^{-4} \sum_{\substack{i,j \in \mathsf{A} \\ i>j}} u_{ix}^2 u_{jy}^2 + u_{jx}^2 u_{iy}^2 + u_{ix}^2 u_{jz}^2 + u_{jx}^2 u_{iz}^2 + u_{iy}^2 u_{jz}^2 + u_{jy}^2 u_{iz}^2 - \\
&\qquad - 2u_{ix}u_{iy}u_{jx}u_{jy} - 2u_{ix}u_{iz}u_{jx}u_{jz} - 2u_{iy}u_{iz}u_{jy}u_{jz} \qquad (7.12) \\
&= \sigma_D^{-4} \sum_{\substack{i,j \in \mathsf{A} \\ i>j}} \left|\hat{u}_i \times \hat{u}_j\right|^2 = \sigma_D^{-4} \sum_{\substack{i,j \in \mathsf{A} \\ i>j}} sin^2\alpha_{ij}, \qquad (7.13)
\end{aligned}$$

where α_{ij} is the angle between unit vectors $\hat{u}_i$ and $\hat{u}_j$, which is equal to the absolute value of the cross product between these vectors $\left|\hat{u}_i \times \hat{u}_j\right|$. We have used:

$$\left|\hat{u}_i \times \hat{u}_j\right| = \begin{vmatrix} \hat{e}_x & \hat{e}_y & \hat{e}_z \\ u_{ix} & u_{iy} & u_{iz} \\ u_{jx} & u_{jy} & u_{jz} \end{vmatrix}. \qquad (7.14)$$

For the denominator in Equation 7.10 we have:

$$\begin{aligned}
\det \mathbf{G} =& G_{11}G_{22}G_{33} - G_{11}G_{23}^2 - G_{22}G_{13}^2 - G_{33}G_{12}^2 + G_{12}G_{13}G_{23} \qquad (7.15) \\
=& \sigma_D^{-6} \sum_{i,j,k \in \mathsf{A}} u_{ix}^2 u_{jy}^2 u_{kz}^2 - u_{ix}^2 u_{jy}u_{jz}u_{ky}u_{kz} - u_{iy}^2 u_{jx}u_{jz}u_{kx}u_{kz} \\
&\qquad - u_{iz}^2 u_{jx}u_{jy}u_{kx}u_{ky} + 2u_{ix}u_{iy}u_{jx}u_{jz}u_{ky}u_{kz} \qquad (7.16) \\
=& \sigma_D^{-6} \sum_{\substack{i,j,k \in \mathsf{A} \\ i>j>k}} u_{ix}^2 u_{jy}^2 u_{kz}^2 + u_{ix}^2 u_{ky}^2 u_{jz}^2 + u_{jx}^2 u_{iy}^2 u_{kz}^2 + u_{jx}^2 u_{ky}^2 u_{iz}^2 + u_{kx}^2 u_{iy}^2 u_{jz}^2 + \\
&\qquad + u_{kx}^2 u_{jy}^2 u_{iz}^2 - 2u_{ix}^2 u_{jy}u_{jz}u_{ky}u_{kz} - 2u_{jx}^2 u_{iy}u_{iz}u_{ky}u_{kz} \\
&\qquad - 2u_{kx}^2 u_{iy}u_{iz}u_{jy}u_{jz} - 2u_{iy}^2 u_{jx}u_{jz}u_{kx}u_{kz} - 2u_{jy}^2 u_{ix}u_{iz}u_{kx}u_{kz} \\
&\qquad - 2u_{ky}^2 u_{ix}u_{iz}u_{jx}u_{jz} - 2u_{iz}^2 u_{jx}u_{jy}u_{kx}u_{ky} - 2u_{jz}^2 u_{ix}u_{iy}u_{kx}u_{ky} \\
&\qquad - 2u_{kz}^2 u_{ix}u_{iy}u_{jx}u_{jy} + 2u_{ix}u_{iy}u_{jx}u_{jz}u_{ky}u_{kz} \\
&\qquad + 2u_{ix}u_{iy}u_{kx}u_{kz}u_{jy}u_{jz} + 2u_{jx}u_{jy}u_{ix}u_{iz}u_{ky}u_{kz} \\
&\qquad + 2u_{jx}u_{jy}u_{kx}u_{kz}u_{iy}u_{iz} + 2u_{kx}u_{ky}u_{ix}u_{iz}u_{jy}u_{jz} \\
&\qquad + 2u_{kx}u_{ky}u_{jx}u_{jz}u_{iy}u_{iz} \qquad (7.17) \\
=& \sigma_D^{-6} \sum_{\substack{i,j,k \in \mathsf{A} \\ i>j>k}} \left(\hat{u}_k \cdot \left|\hat{u}_i \times \hat{u}_j\right|\right)^2 = 36\sigma_D^{-6} \sum_{\substack{i,j,k \in \mathsf{A} \\ i>j>k}} V_{ijk}^2, \qquad (7.18)
\end{aligned}$$

where V_{ijk} is the volume of tetrahedron formed by the target, and the endpoints of unit vectors $\hat{u}_i$, $\hat{u}_j$, and $\hat{u}_k$. Here we used:

$$V_{ijk} = \frac{1}{6}\hat{u}_k \cdot \left|\hat{u}_i \times \hat{u}_j\right| = \frac{1}{6}\begin{vmatrix} u_{kx} & u_{ky} & u_{kz} \\ u_{ix} & u_{iy} & u_{iz} \\ u_{jx} & u_{jy} & u_{jz} \end{vmatrix}. \tag{7.19}$$

Finally, from Equation 7.10, Equation 7.13 and Equation 7.18 we have:

$$GDoP = \frac{1}{6}\sqrt{\frac{\sum_{\substack{i,j\in A \\ i>j}} sin^2\alpha_{ij}}{\sum_{\substack{i,j,k\in A \\ i>j>k}} V_{ijk}^2}}. \tag{7.20}$$

We also give the geometric interpretation of the expression. V_{ijk} in the denominator is the volume of tetrahedron formed by the target and the endpoints of unit vectors directed from the target to the anchors. $sin\alpha_{ij}$ we have in the numerator is proportional to the surface of the triangle formed by the target and the endpoints of unit vectors directed from the target to the anchors. We can compare this with the 2D GDoP expression in Equation 3.4 where we have surfaces of the triangles formed by the target and the endpoints of unit vectors directed from the target to the anchors ($\sin\alpha_{ij}$) in the denominator. Volumes of the tetrahedron in the denominator of Equation 7.20 are the 3D equivalent of the triangle surfaces in the denominator of Equation 3.4.

7.2 Integer Linear Programming Formulation

The main challenge for proposing an ILP formulation for 3D compared to 2D is increased problem complexity due to the increased number of target and candidate positions, as well as the fact that four anchors are needed to cover each target. Our goal is to propose a formulation which allows to obtain exact solutions to 3D problems of realistic sizes in moderate computation times. Each target position has to be assigned four anchors, which must not be coplanar and have to meet the GDoP requirement.

We could define one ILP variable for each combination of four candidates in the range of each target, similarly as our ILP1 formulation (see Section 6.1.1). However, if using such an approach with four nodes, the number of variables would be very high, in the order of $N\binom{r}{4}$, where N is the total number of targets and r denotes the average number of candidates in the range of each possible target position. It is well known that high number of variables affects adversely

the execution time of an ILP. Therefore, we completely avoid variables that combine several anchors that cover the target together, and only have one anchor assigned to a target in each variable. Thus we obtain an ILP system that allows to tackle moderate problem sizes in 3D. This formulation is similar to the ILP3 formulation we had for 2D (Section 6.1.4).

For each possible target position t_u, we define the set $R(t_u)$ such that the candidate position $a_i \in R(t_u)$ if the distance between a_i and t_u is lower than the node range and there is line of sight between t_u and a_i.

We define following variables:

- $x_i = 1$ if a node is placed at the candidate position i, equivalently, if candidate position i belongs to the solution. There is one variable x_i for each candidate position i.

- $z_i^u = 1$ if the candidate position i is assigned to monitor target u.

All variables can only take values 0 or 1.

The ILP formulation is given by:

$$min : \sum_i x_i, \tag{7.21a}$$

$$\sum_i z_i^u = 4, \quad (\forall u), \tag{7.21b}$$

$$z_i^u \leq x_i, \quad (\forall u, i), \tag{7.21c}$$

$$z_i^u + z_j^u + z_k^u + z_l^u < 4, \quad (\forall i, j, k, l, u \mid U(a_i, a_j, a_k, a_l, t_u) > U^*) \tag{7.21d}$$

$$z_i^u = 0, \quad x_i \notin R(t_u). \tag{7.21e}$$

Equation 7.21a states that the total number of anchors used has to be minimized. Equation 7.21b states that each target u has to be assigned four anchors to monitor it. Equation 7.21c ensures that an anchor at candidate position i can be assigned to a target only if that candidate position belongs to the solution, that is, if an anchor is placed at the position i. Equation 7.21d ensures that four anchors cannot be assigned to monitor a target if they do not meet the quality constraints, imposed by the upper limit on uncertainty function (see Section 3.2.4 for the definition of uncertainty function). Equation 7.21e ensures that an anchor can only be assigned to a target if it reaches the target.

7.3 Evaluation

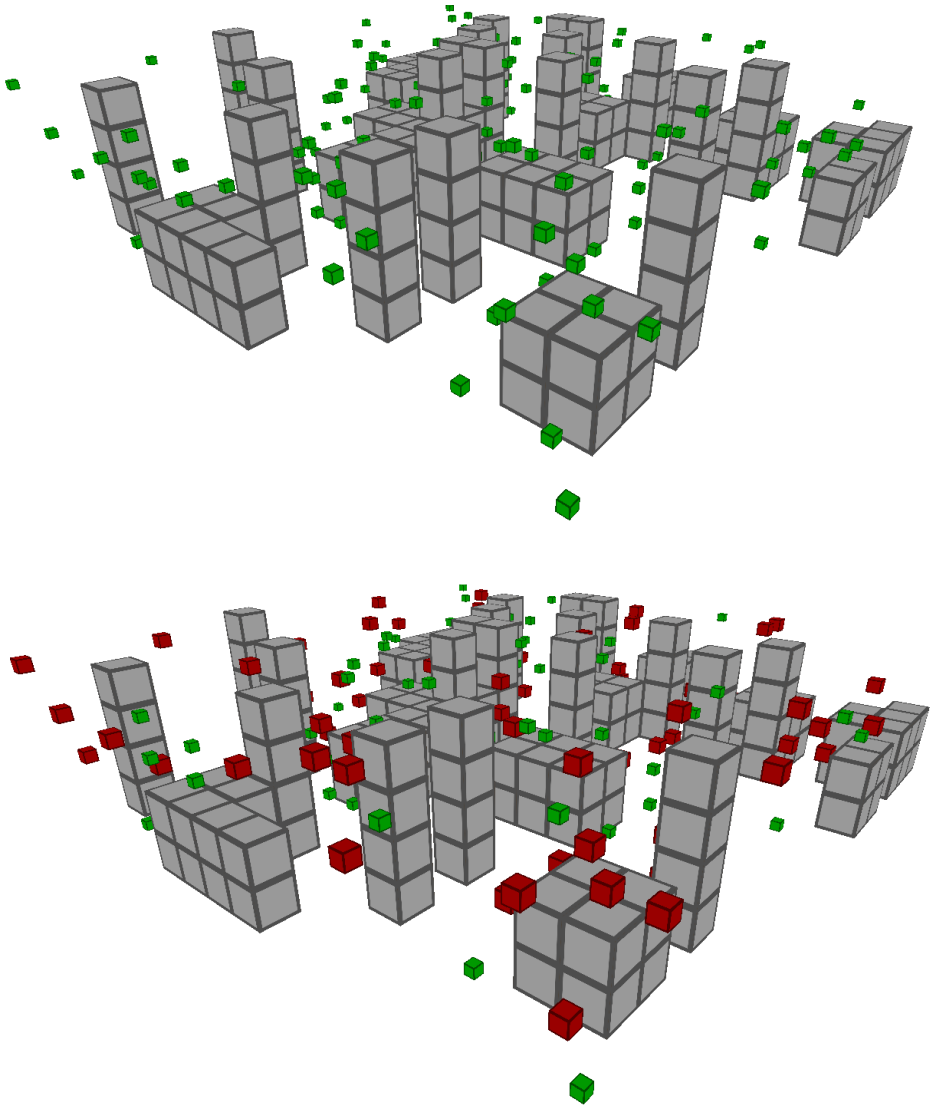


Figure 7.1. A problem instance (above) and a solution (below) for 3D placement for localization. Obstacles are shown as large gray cubes. Candidate positions are shown as small green cubes, and the positions where anchors should be placed are shown by red cubes.

In this section we test both our solutions for 3D placement for localization and give an insight into the speed of both approaches, as well as the quality of the solutions in terms of the number of anchors placed. The code is implemented

in C++ and the simulation experiments are run on Intel(R) Core(TM) i5-3470S 2.90GHz processor. For all experiments, unless if otherwise stated, we use a square grid of resolution (square side) $2\,m$, and the node ranges of $20\,m$. An example of a problem instance and a solution for the 3D problem is shown in Figure 7.1.

7.3.1 Comparison of ILP and Heuristic

We compare the results obtained by ILP to our greedy placement with iterative pruning heuristic. We run simulation experiments in the $100\,m \times 100\,m \times 10\,m$ cubic area. Obstacles are randomly placed so that 20 % of floor area is covered by obstacles. Obstacles can be of two types: pillars or cubes. The dimensions of cubes are $5\,m \times 5\,m \times 5\,m$, and the dimensions of pillars are $2\,m \times 2\,m \times 10\,m$. One of the two types is randomly selected with equal probability for each obstacle. Then 250 candidate positions are randomly selected, so that they can only be placed next to the obstacles, floor, ceiling and walls. Hereby we aim to achieve a realistic scenario, where anchor nodes have to be either placed on the wall, or mounted on the ceiling, or on obstacles such as pillars or large machines. We average results over 100 different test instances, with randomly generated obstacles and candidate positions. Results are as follows. Overhead in terms of the number of anchors placed is 0.048 % on the average with 0.15 % standard deviation. Average time for ILP is $134\,s$ and for heuristic $6.02\,s$ which constitutes a speedup of 22 times. We will show in the next experiment that the speedup increases with increasing problem size.

7.3.2 Scalability of Heuristic and ILP

In the next simulation experiment, we show how the execution time of both greedy placement with iterative pruning (GIP) heuristic and ILP increase with the problem size. We increase the size of the area of the room, with the number of randomly selected candidate positions increasing proportionally, while the ceiling height is kept at $10\,m$ and 20 % of floor area is covered by obstacles. The obstacle shapes are the same as in the previous experiment. We average the result for each size over 100 randomly generated problem instances. Figure 7.2 shows the execution times as the problem size increases, for the greedy placement with iterative pruning heuristic and for the ILP. It is clearly visible that the heuristic is much more scalable then the ILP, which confirms the usefulness of the heuristic.

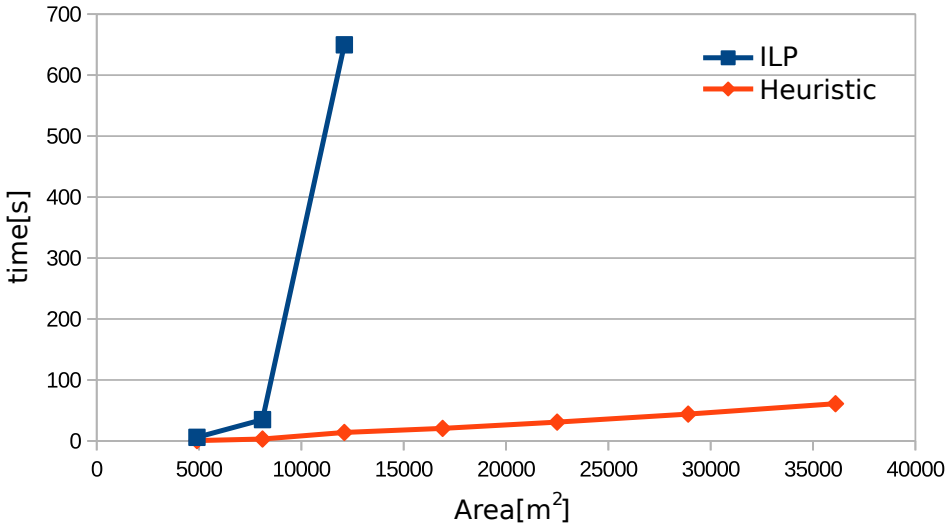


Figure 7.2. Execution times depending on the problem size for greedy placement with iterative pruning and ILP.

7.3.3 Heuristic Parameters

The number of iterations of the heuristic can be changed to achieve the trade-off between the results in terms of the number of nodes placed and the execution time. In this simulation experiment we run the greedy placement with iterative pruning and record the number of anchors and the execution time after each iteration. The room size is 200 $m \times 200\ m \times 20\ m$ with 20 % of floor area covered by obstacles. We average the results over 100 problem instances. The results are shown in Figure 7.3. We can see that the execution time increases linearly with the number of iterations, while most nodes are removed in the initial iterations.

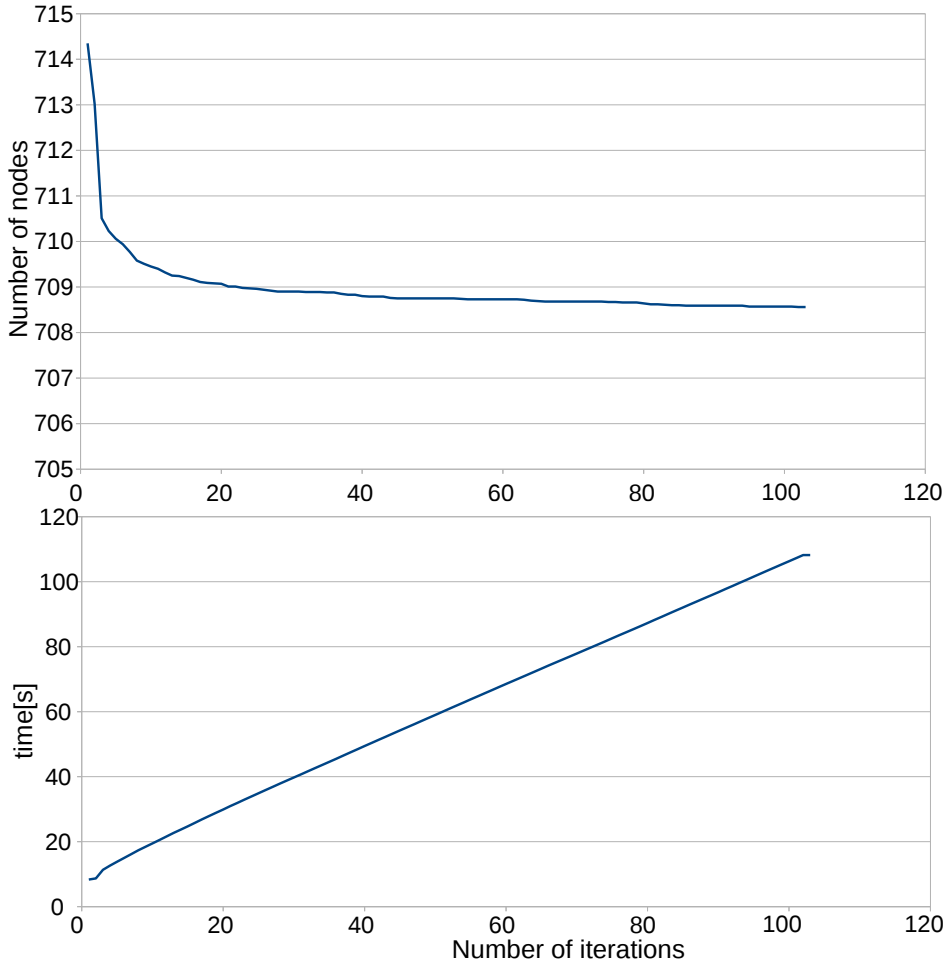


Figure 7.3. Number of anchors in solution (above) and execution time (below) depending on the number of iterations in greedy placement with iterative pruning heuristic.

Chapter 8

Conclusions and Future Work

My thesis focuses on node placement for time-of-flight-based wireless localization networks. We address 1D, 2D and 3D scenarios. Our main motivation are critical safety applications.

We address 1D environments through the experimental study on in-tunnel vehicle localization. In-tunnel localization of vehicles is crucial for emergency management, especially for large trucks transporting dangerous goods such as inflammable chemicals. Compared to outdoor roads, evacuation in tunnels is much more difficult in case of fire or other accidents. We compared characteristics and performance of RSS and ToF ranging inside the road tunnel between Vedeggio and Cassarate in Lugano, Switzerland. We provided detailed distance measurement characterization inside road tunnels concentrating on ToF measurements. We analyzed the dependency of ToF ranging error on node positions and possibility to reduce the error by combining measurements from different channels. We provided statistical characterization of ToF ranging error, based on which a simulation model to use for design of localization algorithms and systems can be made. We designed a complete system for in-tunnel-based localization and showed that such a system is feasible, accurate, and cheap to realize as only few nodes are necessary for high localization accuracy.

Main conclusions of this study are that the radio frequency time of flight can provide precise in-tunnel vehicle localization and that few nodes are sufficient to cover the entire tunnel. Also, the ToF distance measurement precision does not significantly change with the distance in tunnel environments. The NLoS condition affects the distance measurement precision negatively.

We also designed an error compensation unit which used either static or dynamic calibration. We showed that the use of such compensation unit significantly improves the performance of the localization system.

The second part of our work addresses anchor nodes placement optimization for ToF-based localization networks where multilateration is used to obtain the target position based on its distances from fixed and known anchors. The goal is to optimize the placement of anchor nodes, so as to minimize their number, thus minimizing the cost and power consumption of the network, while ensuring that the target can be reliably localized at each point in the area of interest. Our propagation model accounts for the presence of line of sight between nodes, while geometric dilution of precision is used to express the localization error introduced by multilateration. We address this problem in a comprehensive way, including localization in 2D and 3D. We also address the fault-tolerant localization in 2D, that ensures the target can be localized in the whole area of interest, even after any one of the nodes fails.

Our main motivation are critical safety at work applications. Most approaches in literature dealing with optimal configuration for localization in indoor environments, only address 2D scenarios. The extension to 3D is useful when the target is moving in 3D space, for example, if a person is lifted off the floor or climbing. It also allows to fully exploit possibilities of UWB systems by increasing the localization precision. For this purpose, we also derived the GDoP expression for 3D multilateration and gave its geometric interpretation.

We used ILP formulations to obtain exact solution for instances of our placement problems of moderate sizes. We proposed and compared three different ILP formulations for the placement optimization for multilateration-based localization system in 2D. We proposed ILP formulations for the fault-tolerant extension of the same problem, as well as for the placement optimization for multilateration-based localization system in 3D. We also proposed an alternative way to represent GDoP in 2D in terms of only one angle, which allowed to design an ILP formulation in 2D providing shorter computation times.

For the problem instances of large sizes we proposed a greedy placement heuristic with pruning, and its improvement, greedy placement with iterative pruning. The heuristic consists of the greedy node placement stage with execution times in the order of 0.01-0.02 seconds for several hundreds of candidate positions, followed by the iterative pruning which further increases the solution quality.

In addition, we created a simulator that either accepts an existing floor plan as input, or generates the floor plan randomly with given parameters. Random generation of floor plan is useful for testing and comparing proposed approaches, while the existing floor plan feature can be used to tackle the real-life problems. We tested all our approaches through simulation.

For the anchor placement for multilateration-based localization problem, we

obtained solutions of below 2 % overhead in terms of the number of anchors with respect to the optimum on average, with around 5 s average execution time for 130 candidate positions. For the fault-tolerant version of the problem, we obtained solutions of around 1 % anchors overhead with respect to the optimum on average, with 0.4 s execution time for 65 candidate positions, by using greedy heuristic with pruning. For the 3D placement, the greedy heuristic with iterative pruning produced results within 0.05 % of optimum on average, with average execution time of around 6 s for 250 candidate positions, for the problem instances we tested.

8.1 Main Contributions

The main contributions of this thesis are:

- An experimental study of in-tunnel vehicle localization which shows that RF ToF-based in-tunnel vehicle localization is feasible and provides recommendations for node placement in such systems.
- Time of flight distance measurement error characterization for road tunnels.
- An error model and an uncertainty function for a UWB-based localization system that takes into account both Line of Sight (LoS) and GDoP effect on localization uncertainty, for both 2D and 3D localization.
- Three different ILP formulations for 2D multilateration and their comparison by simulation experiments.
- ILP formulation for fault-tolerant placement for multilateration-based localization in 2D that allows to solve problem instances of sizes that are relevant for practical applications.
- ILP formulation for placement for multilateration-based localization in 3D that allows to solve problem instances of sizes that are relevant for practical applications.
- An approximate expression for GDoP for 2D multilateration in terms of only one angle that allows to design ILP formulation for 2D multilateration as well as for fault-tolerant 2D multilateration with shorter execution times.

- A novel greedy placement heuristic with iterative pruning that can tackle instances of placement for multilateration-based localization problem of large sizes. For the 3D placement, the greedy heuristic with iterative pruning produced results within 0.05 % of optimum on average, for the problem instances we tested.
- A GDoP expression for 3D multilateration and its geometric interpretation.
- A simulator for testing and comparing proposed approaches as well as for handling instances of real-life problems.

8.2 Comparison With State-of-the-Art Solutions

The main novel aspects and the differences of our approach from the state of the art are:

- Other works that analyze electromagnetic propagation properties in tunnels mostly focus on signal power (see Section 5.1). We analyze the time of flight measurement error in road tunnels, which is important for designing localization systems. We design a system for in-tunnel vehicle localization. Most existing localization systems involving tunnels focus on mine tunnels, which have significantly different propagation properties than road tunnels.
- Few of the existing works propose exact solutions for node placement problems (see Section 4.1.1 for an overview). We propose several new and different ILP formulations to tackle placement for multilateration and compare them by simulation experiments. We also derive a new approximate expression for GDoP for 2D multilateration that results in an ILP formulation that provides shorter execution times.
- Our solutions for 3D scenarios are based on the GDoP expression we derived for for 3D multilateration. The GDoP expression usually found in literature is only valid for 2D problems.
- It is difficult to compare performances of different optimization approaches found in literature. This is in part because many authors do not provide quantitative comparison of heuristics and metaheuristics with exact algorithms (see Section 4.1.2 and Section 4.1.3). We have designed a simulator implementing all of our solutions and we tested and compared them.

Our non-exact solutions are tested by comparing with exact solutions for different randomly generated problem instances.

- Kirchhof [2013] reports the experimental results where the number of anchors placed is within 1.5 times of the optimal solutions, while our greedy placement with pruning achieves overhead in terms of the number of anchors placed with respect to the optimal one of below 2 % of the average. Also, our 2D solution requires each position to be covered by three nodes while the solution of Kirchhof [2013] requires each point to be covered by two nodes. As explained in Section 2.4 at least three non-collinear nodes are needed to localize a target in 2D space.

- We consider 3D localization networks for indoor environments. Majority of works on indoor localization are restricted to 2D, while the majority of works on 3D anchor placement focus on underwater networks, where the space is free, with no obstacles (see Section 4.2).

Our results on in-tunnel vehicle localization have been published in Balać et al. [2014] and Widmann et al. [2013]. The results on anchor placement for 2D localization have been published in Balać et al. [2015] and Balać et al. [2016].

8.3 Limitation of the Proposed Approach

In Section 3.2.2 we adopted somewhat simplifying assumptions regarding signal propagation properties that we used throughout the work. In particular, we assume that standard deviation of distance measurement error does not depend on distance from the transmitter. We also assume that the standard deviation does not depend on the environment as long as the line of sight between transmitter and receiver is present. These assumptions are needed in order to define the localization uncertainty by means of the GDoP metrics as given by Equation 3.4 for 2D and Equation 7.20 for 3D.

However, most of our solutions do not require the localization uncertainty to be defined by using GDoP. The only exception are ILP2 we proposed for 2D placement in Section 6.1.3 and the ILP formulation we proposed for fault-tolerant placement optimization in Section 6.2. All other ILP formulations, as well as both heuristic solutions we proposed for 2D and 3D placement work directly with the localization uncertainty function U, which can be defined in any way, the only requirement being that U is known for given anchor and target positions. For example, the localization uncertainty function could be modified to incorporate

the distance from the anchor, or it could be obtained experimentally by measuring directly in a known environment. If different ranging errors have to be introduced, a similar ILP formulation as for 3D could be used for fault-tolerant 2D placement instead of the one proposed in Section 6.2. All other methods, with the exception of ILP2, could be used directly as they are.

8.4 Future Work

Related to our work on in-tunnel vehicle localization, future work may include in-tunnel ToF measurement characterization and evaluating localization system performance in the presence of traffic.

For our placement for localization in 2D and 3D, incorporating connectivity will be useful for scenarios requiring wireless data transfer via the network of anchors. For the network to achieve its functionality, it is necessary for the localization data, as well as possibly other data collected by the network, to be transmitted to the central server. As in our proposed system with TWToF (See Section 3.2.1) the target can collect all the information needed to calculate its own location, the target node can communicate its location to the central server, by means of a GSM signal, for example. Therefore, for localization purposes only, it may not be necessary for the anchor nodes to be able to communicate with one another. However, in some cases, communication among anchor nodes may be needed. For example, the anchors may have other functionalities besides localization. Since our work focuses on safety applications, they may have sensors to detect smoke or dangerous chemicals. Also, the target node may send its location data via the network of anchors, to a designated node known as sink node, which then sends it to the central server. In this case, all anchor nodes have to form a connected network. Since TWToF-based systems, unlike TDoA, do not require cables for synchronization (see also Section 2.1) if anchor nodes need to form a connected network, this should be achieved wirelessly.

The communication graph of a wireless network is a graph where each anchor is represented by a graph node, and there is an edge between those nodes that can directly communicate with each other wirelessly. The network is connected if its communication graph is connected, alternatively, if there is a path between any two nodes in the network. We say that a network is k-node-connected if the network will remain connected after removing any k-1 nodes, equivalently, if there are k node disjoint paths between any two nodes in the network. k-connectivity makes the network (k-1)-fault-tolerant.

The future work may include optimizing node placement, such that the net-

work, in addition to providing precise localization in the entire area of interest, is also either connected or 2-connected by means of wireless links.

The ILP formulation can be obtained by adding equations that will guarantee connectivity or 2-connectivity to the existing ILP system of equations providing precise localization. While the execution times for 2-connected network repair using ILP may be too long even for problem instances of moderate sizes, we expect problem instances of realistic sizes of localization with 2-connectivity problem to be manageable in limited times. This is because optimization with multiple criteria, in this case localization and connectivity, reduces the search space with respect to a problem where only one criterion needs to be optimized.

List of Acronyms

CDF	Cumulative Density Function
CRLB	Cramer Rao Lower Bound
FIM	Fisher Information Matrix
FPOL	Fault-tolerant Placement Optimization for Localization
GDoP	Geometric Dilution of Precision
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRASP	Greedy Randomized Adaptive Search Procedure
ILP	Integer Linear Program
LoS	Line of Sight
NLoS	Non Line of Sight
POL	Placement Optimization for Localization
PDF	Probability Density Function
PTA	Piattaforma Technologica Alpina
RF	Radio Frequency
RSSI	Received Signal Strength Indication
TDoA	Time Difference of Arrival
ToF	Time of Flight
TWToF	Two Way Time of Flight
UASN	Underwater Acoustic Sensor Networks
UWB	Ultra Wide Band
WLS	Weighted Least Squares
WSN	Wireless Sensor Networks

Bibliography

Alam, S. M. N. and Haas, Z. J. [2006]. Coverage and connectivity in three-dimensional networks, *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, MobiCom '06, ACM, New York, NY, USA, pp. 346–357.

Alavi, B. and Pahlavan, K. [2006]. Modeling of the toa-based distance measurement error using uwb indoor radio measurements, *Communications Letters, IEEE* **10**(4): 275–277.

Almasaeid, H. M. and Kamal, A. E. [2009]. On the minimum k-connectivity repair in wireless sensor networks, *Communications, 2009. ICC'09. IEEE International Conference on*, IEEE, pp. 1–5.

Amaldi, E., Capone, A., Cesana, M. and Filippini, I. [2012]. Design of wireless sensor networks for mobile target detection, *Networking, IEEE/ACM Transactions on* **20**(3): 784–797.

Ammari, H. M. [2017]. Convex polyhedral space-fillers based connected k-coverage in three-dimensional wireless sensor networks, *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 318–322.

Ammari, H. M. and Das, S. [2010]. A study of k-coverage and measures of connectivity in 3d wireless sensor networks, *IEEE Transactions on Computers* **59**(2): 243–257.

Ash, J. N. and Moses, R. L. [2008]. On optimal anchor node placement in sensor localization by optimization of subspace principal angles, *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, IEEE, pp. 2289–2292.

Bai, X., Kumar, S., Xuan, D., Yun, Z. and Lai, T. H. [2006]. Deploying wireless sensors to achieve both coverage and connectivity, *Proceedings of the 7th ACM*

International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '06, ACM, New York, NY, USA, pp. 131–142.

Bai, X., Xuan, D., Yun, Z., Lai, T. H. and Jia, W. [2008]. Complete optimal deployment patterns for full-coverage and k-connectivity ($k \leq 6$) wireless sensor networks, *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, ACM, pp. 401–410.

Balać, K., Akhmedov, M., Prevostini, M. and Malek, M. [2016]. Topology optimization of wireless localization networks, *European Wireless 2016 (EW2016)*, Oulu, Finland.

Balać, K., Di Giulio, P. A., Taddeo, A. V. and Prevostini, M. [2014]. Time of flight error compensation for in-tunnel vehicle localization, *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, IEEE, pp. 326–331.

Balać, K., Prevostini, M. and Malek, M. [2015]. Optimizing sensor nodes placement for fault-tolerant trilateration-based localization, *IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)*, Zhangjiajie, China.

Bishop, A. N., Fidan, B., Anderson, B. D., Doğançay, K. and Pathirana, P. N. [2010]. Optimality analysis of sensor-target localization geometries, *Automatica* **46**(3): 479–492.

Boukerche, A., Oliveira, H. A. B. F., Nakamura, E. F. and Loureiro, A. A. F. [2008]. Vehicular ad hoc networks: A new challenge for localization-based systems, *Comput. Commun.* **31**: 2838–2849. Important paper, state of art on vehicular networks, nice overview of applications and techniques. Emphasis on data fusion.

Bredin, J., Demaine, E., Hajiaghayi, M. and Rus, D. [2010]. Deploying sensor networks with guaranteed fault tolerance, *Networking, IEEE/ACM Transactions on* **18**(1): 216–228.

Bulusu, N., Heidemann, J. and Estrin, D. [2001]. Adaptive beacon placement, *Proceedings 21st International Conference on Distributed Computing Systems*, pp. 489–498.

Chakrabarty, K., Iyengar, S. S., Qi, H. and Cho, E. [2002]. Grid coverage for surveillance and target location in distributed sensor networks, *IEEE Transactions on Computers* **51**(12): 1448–1453.

Chen, H., Wang, G., Wang, Z., So, H. and Poor, H. [2012]. Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks, *Wireless Communications, IEEE Transactions on* **11**(1): 108–116.

Chen, Y., Francisco, J., Trappe, W. and Martin, R. [2006]. A practical approach to landmark deployment for indoor localization, *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, Vol. 1, pp. 365–373.

Cheriyan, J., Vempala, S. and Vetta, A. [2002]. Approximation algorithms for minimum-cost k-vertex connected subgraphs, *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, ACM, New York, NY, USA, pp. 306–312.

Dardari, D., Conti, A., Ferner, U., Giorgetti, A. and Win, M. Z. [2009]. Ranging with ultrawide bandwidth signals in multipath environments, *Proceedings of the IEEE* **97**(2): 404–426.

Dayekh, S., Affes, S., Kandil, N. and Nerguizian, C. [2010]. Cooperative localization in mines using fingerprinting and neural networks, *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pp. 1 –6.

Dhillon, S. S. and Chakrabarty, K. [2003]. Sensor placement for effective coverage and surveillance in distributed sensor networks, *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, Vol. 3, pp. 1609–1614 vol.3.

Didascalou, D., Maurer, J. and Wiesbeck, W. [2001]. Subway tunnel guided electromagnetic wave propagation at mobile communications frequencies, *Antennas and Propagation, IEEE Transactions on* **49**(11): 1590 –1596.

Dudley, D., Lienard, M., Mahmoud, S. and Degauque, P. [2007]. Wireless propagation in tunnels, *Antennas and Propagation Magazine, IEEE* **49**(2): 11 –26.

Efrat, A., Har-Peled, S. and Mitchell, J. S. B. [2005]. Approximation algorithms for two optimal location problems in sensor networks, *2nd International Conference on Broadband Networks, 2005.*, pp. 714–723 Vol. 1.

Exslie, A. G., Lagace, R. L. and Strong, P. F. [1974]. Theory of the propagation o f uhf radio waves in coal mine tunnels.

Fleischer, L. [2001]. A 2-approximation for minimum cost 0, 1, 2 vertex connectivity, *In Integer Programming and Combinatorial Optimization, number 2081 in the Lecture Notes in Computer Science*, Springer, pp. 115–129.

Gezici, S., Tian, Z., Giannakis, G. B., Kobayashi, H., Molisch, A. F., Poor, H. V. and Sahinoglu, Z. [2005]. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks, *IEEE Signal Processing Magazine* **22**(4): 70–84.

Glotzbach, T., Moreno-Salinas, D., Pascoal, A. and Aranda, J. [2013]. Optimal sensor placement for acoustic range-based underwater robot positioning, *IFAC Proceedings Volumes* **46**(33): 215–220.

Gustafsson, F. and Gunnarsson, F. [2003]. Positioning using time-difference of arrival measurements, *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, Vol. 6, pp. VI–553–6 vol.6.

Gwon, Y. and Jain, R. [2004]. Error characteristics and calibration-free techniques for wireless lan-based location estimation, *Proceedings of the second international workshop on Mobility management & wireless access protocols*, MobiWac '04, ACM, New York, NY, USA, pp. 2–9.

Hao, B., Tang, J. and Xue, G. [2004]. Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation, *High Performance Switching and Routing, 2004. HPSR. 2004 Workshop on*, pp. 246–250.

Howard, A., Matarić, M. J. and Sukhatme, G. S. [2002]. An incremental self-deployment algorithm for mobile sensor networks, *Autonomous Robots* **13**(2): 113–126.

Huang, C.-F. and Tseng, Y.-C. [2005]. The coverage problem in a wireless sensor network, *Mobile Networks and Applications* **10**(4): 519–528.

Jiang, Y. and Leung, V. [2007]. An asymmetric double sided two-way ranging for crystal offset, *Signals, Systems and Electronics, 2007. ISSSE '07. International Symposium on*, pp. 525 –528.

Jin, M., Rong, G., Wu, H., Shuai, L. and Guo, X. [2012]. Optimal surface deployment problem in wireless sensor networks, *2012 Proceedings IEEE INFOCOM*, pp. 2345–2353.

Kay, S. M. [1993]. Fundamentals of statistical signal processing, volume i: estimation theory.

Ke, W.-C., Liu, B.-H. and Tsai, M.-J. [2007]. Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points is np-complete, *IEEE Transactions on Computers* **56**(5).

Kelly, A. [2003]. Precision dilution in triangulation based mobile robot position estimation, *Intelligent Autonomous Systems*, Vol. 8, pp. 1046–1053.

Kershner, R. [1939]. The number of circles covering a set, *American Journal of Mathematics* **61**(3): 665–671.

Khodjaev, J., Park, Y. and Malik, A. S. [2010]. Survey of nlos identification and error mitigation problems in uwb-based positioning algorithms for dense environments, *annals of telecommunications-annales des télécommunications* **65**(5-6): 301–311.

Kirchhof, N. [2013]. Optimal placement of multiple sensors for localization applications, *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, pp. 1–10.

Kong, L., Zhao, M., Liu, X.-Y., Lu, J., Liu, Y., Wu, M.-Y. and Shu, W. [2014]. Surface coverage in sensor networks, *IEEE Transactions on Parallel and Distributed Systems* **25**(1): 234–243.

Lanzisera, S., Zats, D. and Pister, K. [2011]. Radio frequency time-of-flight distance measurement for low-cost wireless sensor localization, *Sensors Journal, IEEE* **11**(3): 837 –845.

Lee, H. B. [1975]. Accuracy of range-range and range-sum multilateration systems, *IEEE Transactions on Aerospace and Electronic Systems* **AES-11**(6): 1346–1361.

Li, D., Liu, W. and Cui, L. [2010]. Easidesign: An improved ant colony algorithm for sensor deployment in real sensor network system, *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pp. 1–5.

Liao, L., Chen, W., Zhang, C., Zhang, L., Xuan, D. and Jia, W. [2011]. Two birds with one stone: Wireless access point deployment for both coverage and localization, *Vehicular Technology, IEEE Transactions on* **60**(5): 2239–2252.

Liénard, M., Bétrencourt, S. and Degauque, P. [2000]. Propagation in road tunnels: a statistical analysis of the field distribution and impact of the traffic, *Annals of Telecommunications* **55**: 623–631. 10.1007/BF02999833.

Lienard, M. and Degauque, P. [2000]. Natural wave propagation in mine environments, *Antennas and Propagation, IEEE Transactions on* **48**(9): 1326 –1339.

Lim, H., Kung, L., Hou, J. and Luo, H. [2010]. Zero-configuration indoor localization over ieee 802.11 wireless infrastructure, *Wireless Networks* **16**(2): 405–420.

Lin, F. Y. S. and Chiu, P. L. [2005]. A near-optimal sensor placement algorithm to achieve complete coverage-discrimination in sensor networks, *IEEE Communications Letters* **9**(1): 43–45.

Liu, H., Wan, P.-J. and Jia, X. [2005]. Fault-tolerant relay node placement in wireless sensor networks, *Computing and Combinatorics*, Springer, pp. 230–239.

Liu, L. and Ma, H. [2012]. On coverage of wireless sensor networks for rolling terrains, *IEEE Transactions on Parallel and Distributed Systems* **23**(1): 118–125.

Liu, L., Ma, M., Liu, C. and Shu, Y. [2017]. Optimal relay node placement and flow allocation in underwater acoustic sensor networks, *IEEE Transactions on Communications* **65**(5): 2141–2152.

Liu, R., Wassell, I. J. and Soga, K. [2010]. Relay node placement for wireless sensor networks deployed in tunnels, *2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 144–150.

Mansoor, U. and Ammari, H. M. [2014]. *Coverage and Connectivity in 3D Wireless Sensor Networks*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 273–324.

Marano, S., Gifford, W. M., Wymeersch, H. and Win, M. Z. [2010]. Nlos identification and mitigation for localization based on uwb experimental data, *Selected Areas in Communications, IEEE Journal on* **28**(7): 1026–1035.

Misra, S., Hong, S. D., Xue, G. and Tang, J. [2010]. Constrained relay node placement in wireless sensor networks: Formulation and approximations, *Networking, IEEE/ACM Transactions on* **18**(2): 434–447.

NXP [2011]. *Data Sheet: JN5148-001 IEEE802.15.4 Wireless Microcontroller*.

Perez-Ramirez, J., Borah, D. K. and Voelz, D. G. [2013]. Optimal 3-d landmark placement for vehicle localization using heterogeneous sensors, *IEEE Transactions on Vehicular Technology* **62**(7): 2987–2999.

Plotkin, S. A., Shmoys, D. B. and Tardos, E. [1991]. Fast approximation algorithms for fractional packing and covering problems, *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pp. 495–504.

Polzin, T. [2003]. *Algorithms for the Steiner problem in networks*, PhD thesis, Universitätsbibliothek.

PTA [2010]. Pta project, `http://www.progettopta.net/`.

PTA-DESTNATION [2015]. Pta-destination project, `http://www.ptadestination.net`.

Pu, J., Xiong, Z. and Lu, X. [2009]. Fault-tolerant deployment with k-connectivity and partial k-connectivity in sensor networks, *Wireless Communications and Mobile Computing* **9**(7): 909–919.

Qu, Y. and Georgakopoulos, S. V. [2011]. Relocation of wireless sensor network nodes using a genetic algorithm, *WAMICON 2011 Conference Proceedings*, pp. 1–5.

Rappaport, T. [1996]. *Wireless communications: principles and practice*, Prentice Hall communications engineering and emerging technologies series, Prentice Hall PTR.

Redondi, A., Tagliasacchi, M., Cesana, M., Borsani, L., Tarrío, P. and Salice, F. [2010]. Laura - localization and ubiquitous monitoring of patients for health care support, *Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), 2010 IEEE 21st International Symposium on*, pp. 218 – 222.

Schroeder, J., Galler, S., Kyamakya, K. and Jobmann, K. [2007]. Nlos detection algorithms for ultra-wideband localization, *2007 4th Workshop on Positioning, Navigation and Communication*, pp. 159–166.

Sheng, X. and Hu, Y.-H. [2005]. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks, *IEEE Transactions on Signal Processing* **53**(1): 44–53.

Silva, B., Pang, Z., Åkerberg, J., Neander, J. and Hancke, G. [2014]. Experimental study of uwb-based high precision localization for industrial applications, *2014 IEEE International Conference on Ultra-WideBand (ICUWB)*, pp. 280–285.

Sitanayah, L., Brown, K. N. and Sreenan, C. J. [2011]. Fault-tolerant relay deployment for k node-disjoint paths in wireless sensor networks, *2011 IFIP Wireless Days (WD)*, pp. 1–6.

Spirito, M. A. [2001]. On the accuracy of cellular mobile station location estimation, *IEEE Transactions on Vehicular Technology* **50**(3): 674–685.

Stutzle, T. and Hoos, H. [1997]. Max-min ant system and local search for the traveling salesman problem, *Evolutionary Computation, 1997., IEEE International Conference on*, pp. 309–314.

Sun, Z. and Akyildiz, I. [2010]. Channel modeling and analysis for wireless networks in underground mines and road tunnels, *Communications, IEEE Transactions on* **58**(6): 1758 –1768.

Tekdas, O. and Isler, V. [2010]. Sensor placement for triangulation-based localization, *Automation Science and Engineering, IEEE Transactions on* **7**(3): 681–685.

Thorbjornsen, B., White, N. M., Brown, A. D. and Reeve, J. S. [2010]. Radio frequency (rf) time-of-flight ranging for wireless sensor networks, *Measurement Science and Technology* **21**(3): 035202.

Vossiek, M., Wiebking, L., Gulden, P., Wieghardt, J., Hoffmann, C. and Heide, P. [2003]. Wireless local positioning, *Microwave Magazine, IEEE* **4**(4): 77 – 86.

Wang, B. [2011]. Coverage problems in sensor networks: A survey, *ACM Comput. Surv.* **43**(4): 32:1–32:53.

Wang, J. and Zhong, N. [2006]. Efficient point coverage in wireless sensor networks, *Journal of Combinatorial Optimization* **11**(3): 291–304.

Widmann, D., Balac, K., Taddeo, A., Prevostini, M. and Puiatti, A. [2013]. Characterization of in-tunnel distance measurements for vehicle localization, *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, pp. 2311–2316.

Wu, Q., Rao, N. S. V., Du, X., Iyengar, S. S. and Vaishnavi, V. K. [2007]. On efficient deployment of sensors on planar grid, *Comput. Commun.* **30**(14-15): 2721–2734.

Xu, X. and Sahni, S. [2007]. Approximation algorithms for sensor deployment, *Computers, IEEE Transactions on* **56**(12): 1681–1695.

Yamaguchi, Y., Abe, T. and Sekiguchi, T. [1989]. Radio wave propagation loss in the vhf to microwave region due to vehicles in tunnels, *Electromagnetic Compatibility, IEEE Transactions on* **31**(1): 87 –91.

Yang, B. [2007]. Different sensor placement strategies for tdoa based localization, *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Vol. 2, pp. II–1093–II–1096.

Yang, B. and Scheuing, J. [2005]. Cramer-rao bound and optimum sensor array for source localization from time differences of arrival, *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, Vol. 4, pp. iv/961–iv/964 Vol. 4.

Yarlagadda, R., Ali, I., Al-Dhahir, N. and Hershey, J. [2000]. Gps gdop metric, *Radar, Sonar and Navigation, IEE Proceedings -* **147**(5): 259–264.

Yun, Z., Bai, X., Xuan, D., Lai, T. H. and Jia, W. [2010]. Optimal deployment patterns for full coverage and k- connectivity (k$\leq$ 6) wireless sensor networks, *IEEE/ACM Trans. Netw.* **18**(3): 934–947.

Zhang, C., Bai, X., Teng, J., Xuan, D. and Jia, W. [2010]. Constructing low-connectivity and full-coverage three dimensional sensor networks, *IEEE Journal on Selected Areas in Communications* **28**(7): 984–993.

Zhao, S., Chen, B. M. and Lee, T. H. [2013]. Optimal sensor placement for target localisation and tracking in 2d and 3d, *International Journal of Control* **86**(10): 1687–1704.

Zhao, S., Chen, B. M. and Lee, T. H. [2014]. Optimal deployment of mobile sensors for target tracking in 2d and 3d spaces, *IEEE/CAA Journal of Automatica Sinica* **1**(1): 24–30.

Zhou, J., Shi, J. and Qu, X. [2010]. Landmark placement for wireless localization in rectangular-shaped industrial facilities, *IEEE Transactions on Vehicular Technology* **59**(6): 3081–3090.

Zou, Y. and Chakrabarty, K. [2004]. Uncertainty-aware and coverage-oriented deployment for sensor networks, *Journal of Parallel and Distributed Computing* **64**(7): 788–798.