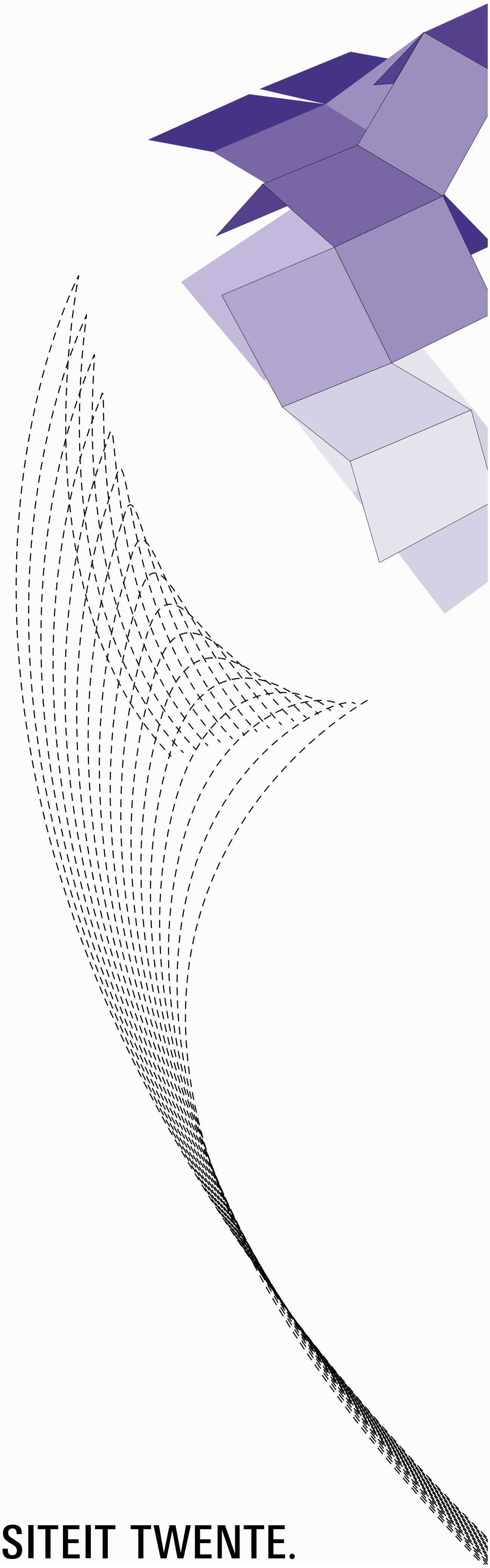
MASTER THESIS
PUBLIC



RANGING AND LOCALISATION ERROR MITIGATION IN INDOOR OBSTRUCTED DIRECT PATH CONDITIONS

Sjoerd Op 't Land

FACULTY OF EEMCS
TELECOMMUNICATION ENGINEERING
SHORT RANGE RADIO

COMMITTEE
prof. dr. ing. Frank Leferink
dr. ir. Mark Bentum MBA
dr. ir. Jac Romme
dr. ir. Arjan Meijerink
Dr. Ing. Dries Neirynck
Yakup Kılıç MSc

27 November 2010 (confidential release)
14 March 2011 (public release)

UNIVERSITEIT TWENTE.

Summary

This master thesis is part of the 'Localisation in Smart Dust Sensor Networks' project. Smart dust is the future vision of having many small, light, cheap, dependable, long-lasting, biodegradable network nodes that can even be carried by the wind. The ability of these network nodes to localise themselves is crucial to many applications. Lateration with Ultra-Wideband (UWB) Time of Flight (ToF) *range* (distance) measurements is widely regarded as the method of choice for localisation in smart dust networks.

In practice, the performance of this localisation technique is impaired by Obstructed Direct Paths (ODPs). An obstruction delays or removes the detectable radio path, causing the real distance to be overestimated. These positively biased range measurements, in turn, cause localisation errors. In this thesis, we perform a survey of known ODP detection techniques, some of which are chiefly tested in simulation. All reviewed techniques consist in evaluating *features*: functions of one measured channel impulse response. Then we design a measurement set-up with a state-of-the-art UWB transceiver and physical obstacles, to test the known ODP detection techniques.

By combining the features from each technique, we are able to estimate both the bias and the precision of each range measurement. Using this information, we can discard distance measurements that appear to be imprecise. This generally improves the localisation accuracy if the *geometry* (the spatial arrangement of nodes) is reasonable; if the geometry is bad, the localisation accuracy worsens slightly.

Collaterally, we propose a new improvement on the existing leading edge detection, yielding a ranging accuracy in line-of-sight (LOS) conditions of 6 cm mean absolute error, where the existing leading edge detection yields 8 cm accuracy.

Definitions

accuracy Inverse of the mean absolute distance error [1, sec. IIIa].

air channel The causal relation between any electromagnetic wave departing from the transmitter and its arrival at the receiver antenna. *This includes scattering and attenuation by obstructions.*

anchor Node of which the location is known or estimated a priori. *Prior to localisation.*

channel The causal relation between an excitation at the baseband input of the transmitter and the baseband output of the receiver. *That is, the channel consist of the chain modulator, transmitter antenna, air channel, receiver antenna and demodulator, including sensitivity.*

chip Baseband pulse of finite duration.

component Part of a system. *For example: a capacitor.*

direct path The shortest path, according to the Euclidian model of space. *Consequently, there is always a direct path, which might or might not be obstructed.*

emulation System X is said to emulate another system Y when the behaviour of X mimics the behaviour of Y by means of a mechanism analogous to the mechanism of Y.

frame The bits from the MAC layer entity, transported in a packet by the PHY layer entity.

geometry Spatial constellation of anchors. The inverse RGDoP is a measure for the quality of the geometry.

line-of-sight A dominant direct path in a channel for visible light.

localisation Estimation of a location.

location Position of a node in Euclidian space.

multilateration Localisation using distance differences to the target, between the anchors.

packet The bits that are transported together in time across the physical medium (after [2, 3.31]).

path A possible route of an electromagnetic wave through the air channel.

precision Inverse of the GDoP (standard deviation of the distance error) [1, sec. IIIb]. *Precision is a measure of robustness.*

protocol Convention on syntax and semantics.

range Euclidian distance between two nodes.

ranging Estimation of a range.

representative A representative set of a whole is a strict subset with finite members of the whole, where the elements of the representative set have the same relevance.

requirement Demand on the behaviour of a system.

scene Measurements that are taken in the same room with the same obstacle positions and types are taken in the same scene.

simulation System X is said to simulate another system Y when the behaviour of X mimics the behaviour of Y by means of a (mathematical) model of the behaviour of Y.

specification Demand the behaviour of a component.

system Physical whole.

target Node of which the location is to be estimated.

trilateration Localisation using distances from the target to the anchors.

ultra-wideband An ultra-wideband signal has a bandwidth of the lesser of 500 MHz and 20% of the center frequency [3].

Contents

1 Introduction

In this chapter, we introduce the context of this thesis project: location-aware smart dust sensor networks. Next, we propose to create such a network as a demonstrator. Then we describe how localisation in this demonstrator works. Subsequently, we describe its weakest point, being ranging accuracy in Obstructed Direct Path (ODP) conditions. Then we define the scope of our research project as solving this problem. Finally, we outline the structure of the remainder of this thesis.

1.1 Smart Dust Sensor Networks

Smart dust (or, less utopian, Wireless Sensor Networks (WSNs)) is the future vision of having many small, light, cheap, dependable, long-lasting, biodegradable network nodes that can even be carried by the wind. These nodes can collaborate in some task by using their sensors and/or actuators. The ability of the nodes to automatically estimate their locations (i.e. to localise themselves) is crucial to many applications and some applications even *consist* in such localisation. Smart dust could be applied in health, agriculture, geology, retail, military, home and incident management [4]. This thesis is part of a research project that is titled "Localisation in Smart Dust Sensor Networks". The project is performed in the Short Range Radio (SRR) chair, part of the Telecommunication Engineering (TE) group at the University of Twente [5].

For example, smart dust can by applied in precision agriculture [6]: the nodes can be distributed over the land, together with the seeds. All nodes are able to sense the advent of a plague and are also able to localise themselves. If a plague strikes the field, the sensors communicate their measurements with the farmer, including their respective positions. Pesticide can then be applied just there, mitigating environmental impact.

Localisation for smart dust deserves research, because smart dust has some particular properties that disqualify existing localisation technologies. For example, smart dust nodes should last long without maintenance, in the order of years. An ordinary IEC PR44 zinc-air cell (1.65 V, 600 mAh) contains about 1 Wh or 3600 J. For a three-year lifetime, the node should consume about 40 μW continuous power. A modern Global Positioning System (GPS) receiver consumes 90-720 μW[1] [7], excluding a tranceiver and sensor electronics that would be needed for smart dust applications. Therefore, the total power consumption will be significantly higher than specified, resulting in a lifetime much shorter then required. For another example, indoor precision localisation is vital to many smart home applications. However, GPS does not allow for indoor localisation and has a typical error of below 10 m in line-of-sight (LOS) conditions [8]. Therefore, existing localisation technology like GPS is not suited for smart dust applications.

1.2 Proposed Smart Dust Demonstrator

The design of localising smart dust for a real application encompasses different subjects. Both network-level decisions (such as the medium and protocol) and node-level decisions (such as the transducer, transceiver and power supply) need to be taken, see Figure 1.1. It is not easy to know in advance what subjects will form a 'bottleneck' and what subjects are relatively easy to design for. There is a substantial risk to perform research on one subject, while it is not the most important. A common way of avoiding this pitfall, is to design a *demonstrator*: a complete system, made with the simplest means possible, to find out where the problems are, if any at all.

[1] It is possible to cut on the power consumption of a GPS receiver by duty-cycling, but the time necessary to lock on the satellites limits the achievable gain.

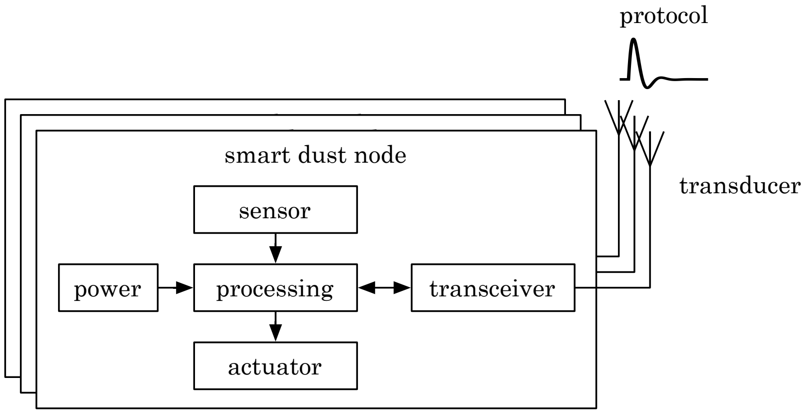


Figure 1.1: Break-down of a smart dust system (as designed in Appendix A).

We drew up requirements for a demonstrator that is a firm step forward in terms of energy consumption, size and localisation accuracy (Section A.1). The network nodes shall last one year without maintenance, measure 2×2 cm and localise with 1 cm mean absolute error.

The following design decisions have been taken (for underpinning, see Appendix A). To lower the cost, the nodes will localise themselves by means of distance (range) measurements, instead of angle measurements. For high accuracy, these range measurements will be time-based (Time of Arrival (ToA) or Time Difference of Arrival (TDoA)), instead of power-based (Received Signal Strength (RSS)). To be able to penetrate objects, the medium is Ultra-Wideband (UWB) radio, instead of acoustic waves [9]. Lacking commercially available transceivers that comply with the specifications, the transceiver will be the Ultra Low Power (ULP) IC that is currently developed at IMEC.

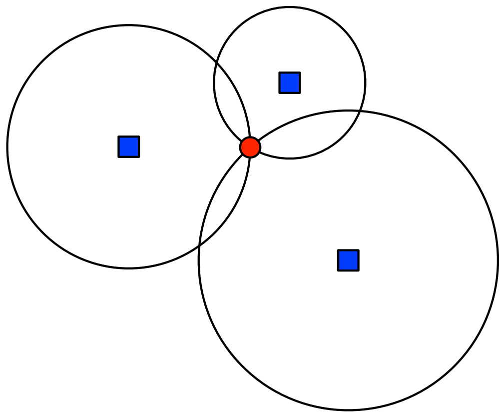
1.3 Localisation in the Proposed Demonstrator

To calculate the node's respective locations, the demonstrator will use range (distance) measurements between the nodes. We first discuss the way locations are calculated. Then we review the way distances are measured. (Both are discussed and analysed in more detail in Chapter 2.)

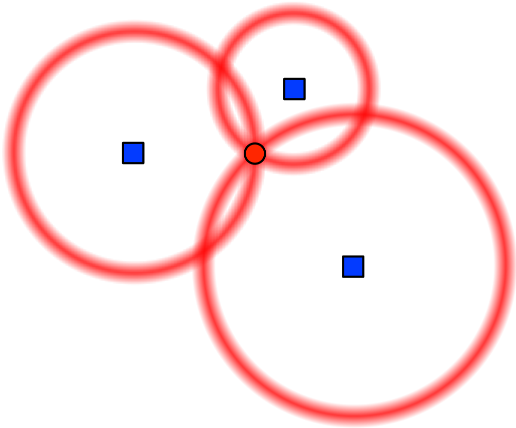
1.3.1 Lateration

Let us look at a typical localisation based on range measurements (lateration). Let there be anchors, being nodes with an *a priori* known location in the horizontal plane. Let there be one target, with an *a priori* unknown location on this same horizontal plane, being the node that we want to localise. If we know that the range between the target and the first anchor is r_1, the target must be on a circle with radius r_1, centred at the first anchor (Figure 1.2a). Adding the range measurement r_2 to the second anchor means that the target must also be on another circle with radius r_2, centred at the second anchor. This means the anchor must be at one of the in general two intersections between both circles. A third range measurement r_3 will generate a circle that in general disambiguates between both intersections. This way, the target can be unambiguously localised by using only three range measurements.

In reality, range measurements are impaired by errors. Therefore, they should be treated as range *estimates*. We assume that these range estimates are unbiased; that is, the additive error has zero mean. The quality of range measurements can, for instance, be characterised by two metrics: *accuracy* and *precision* as defined by [1]. Accuracy then is the mean of the absolute error. Precision then is the standard deviation of the error; this is a metric for robustness of the estimator. These definitions will be used throughout this report.



(a) Localisation with perfect range measurements.



(b) Localisation with range estimates.

Figure 1.2: Localisation using range measurements. The blue squares are anchors, the red circles are targets.

The localisation now becomes finding the most *probable* location, given the range estimates and their statistical properties (Figure 1.2b). Statistically, range errors cause localisation errors, depending on the *geometry* (the position of the anchors relative to the target; examples in Figure 2.4). The quality of the resulting localisation can also be characterised by accuracy and precision. If the geometry is unknown and there are few anchors, the localisation error is in the same order of magnitude as the causing range errors. With an increasing number of anchors, the localisation error decreases.

1.3.2 Time Based Ranging

The demonstrator measures distances between nodes by time based ranging. All time based ranging techniques measure the time it takes an electromagnetic wave to travel the distance to be measured. Knowing the propagation speed of the wave, the distance can be calculated from the time. There are different time based ranging techniques that differ in the required availability of time reference. We now give a brief overview of available techniques, a more detailed description can be found in Section A.3.

The simplest technique is called ToA and requires that two nodes have a shared notion of time [10, Ch. 8]. The transmitter emits a signal that contains a timestamp of the transmission instant. The receiver determines the time of arrival and subtracts the timestamp. The resulting difference is the time of flight of the signal; multiplied with the speed of light, this is the range estimate. Although a simple technique, it is not easy to establish a common notion of time among the nodes of a wireless network. TDoA only requires the anchors to have a shared notion of time and Return Time of Arrival (RToA) only requires the nodes to have a shared notion of frequency, i.e. have equal clock rates. Practically, the time the nodes need to respond in Two-Way Ranging (TWR) needs to be communicated or agreed upon beforehand.

In indoor situations, there are multiple paths between two nodes; i.e. electromagnetic waves can propagate from transmitter to receiver via multiple routes with different lengths. Only the route that coincides with the distance to be measured should be used. As we want to measure the distance between two points in Euclidian space, this is the shortest route or *direct path*.

For example, the signal from Tx to Rx1 in Figure 1.3 travels along the direct path and a reflection against a wall. Rx1 should take care to register the time of the first arriving signal. In the case of Rx2, the direct path is attenuated and also delayed by the wall, because the velocity of propagation through concrete is lower than through air. So although Rx2 registers the first path, it will overestimate the distance because Rx2 assumes that the wave propagated only through air. The direct path that reaches Rx3 is attenuated so severely, that it goes undetected. The first detected path is mistakenly interpreted as the direct path. Consequently, the distance is overestimated.

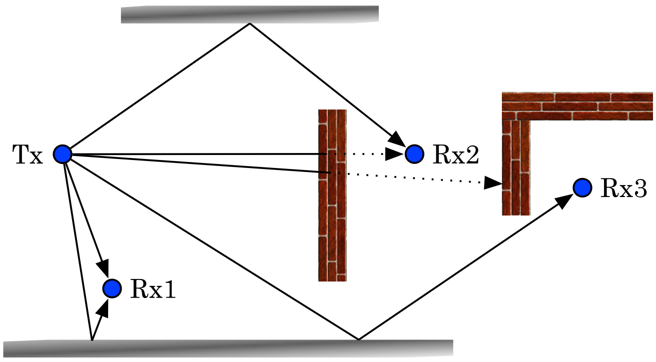


Figure 1.3: Possible multipath propagation scenario, after [11, Fig. 1].

We can conclude that only if the direct path is unobstructed, the propagation delay of the first signal leads to an unbiased estimate of the distance. In all other cases, the distance estimate will be positively biased. The former is called a Dominant Direct Path (DDP) channel condition, the latter is called an Obstructed Direct Path (ODP) channel condition.

1.4 Bottleneck: Errors in Obstructed Direct Path Conditions

To what extent does the designed demonstrator comply with its requirements? The demonstrator measures 2.5×2.5 cm and consumes 650 μW during transmission. The nodes can range with an accuracy of 9 cm in LOS conditions. (These requirements were derived during this thesis project, but are considered out of scope. The interested reader is referred to Section A.5.)

The size of the node is close to the required 2×2 cm. If the transmitter is on during 20% of the time, the node can last one year on its battery. Probably, the transmitters can be on much shorter[2]. In the demonstrator, 200 nodes will be deployed, so if they have a range accuracy of 9 cm each, the resulting localisation accuracy may come close to the required 1 cm, assuming reasonable geometry.[3] We conclude that the designed demonstrator approximates the requirements reasonably well.

However, as soon an obstruction is placed between the nodes, the (positive) range error becomes in the order of metres (this is measured in more detail later, see Figure 5.13). This large a range error will certainly cause large localisation errors with respect to 1 cm. As this is an important bottleneck in the localisation performance of the demonstrator, we decide to focus on mitigating localisation errors in ODP conditions.

[2]This assumes that the power consumption of the transmitter is dominant for the total power budget. Depending on the protocol, this might or might not be a valid assumption.

[3]There is no simple relation between the number of nodes and the Relative Geometric Dilution of Precision (RGDoP), because it depends on the geometry.

1.5 Scope of this Thesis

We can think of two complimentary methods of mitigating the effect of ODP conditions on the localisation error. First, we could try to mitigate the *individual* ranging errors that together constitute the localisation error. Second, we could try to improve the location estimate by *combining* the (erroneous) range estimates in a smarter way, for example by using ODP detection information of each range estimate. Therefore, the main question of this thesis is:

what localisation accuracy and precision can be achieved under ODP conditions, by means of (1) mitigation of the effect of ODP conditions on the individual ranging errors and (2) ODP-aware combination of the range estimates?

1.6 Structure of this Thesis

This thesis is outlined as follows. We will first analyse more in detail what is causing localisation errors in Chapter 2. Then, we list published methods of mitigating these errors in Chapter 3. IMEC's IC is described in Chapter 4. A measurement set-up to answer the abovementioned research question is designed in Chapter 5, that tries to reproduce ODP conditions as analysed in Chapter 2 and using IMEC's IC described in Chapter 4. Chapter 6 continues by presenting the measurement results and the effectiveness of mitigation with the techniques reviewed in Chapter 3. Conclusions and recommendations for further research and development are given in Chapter 7.

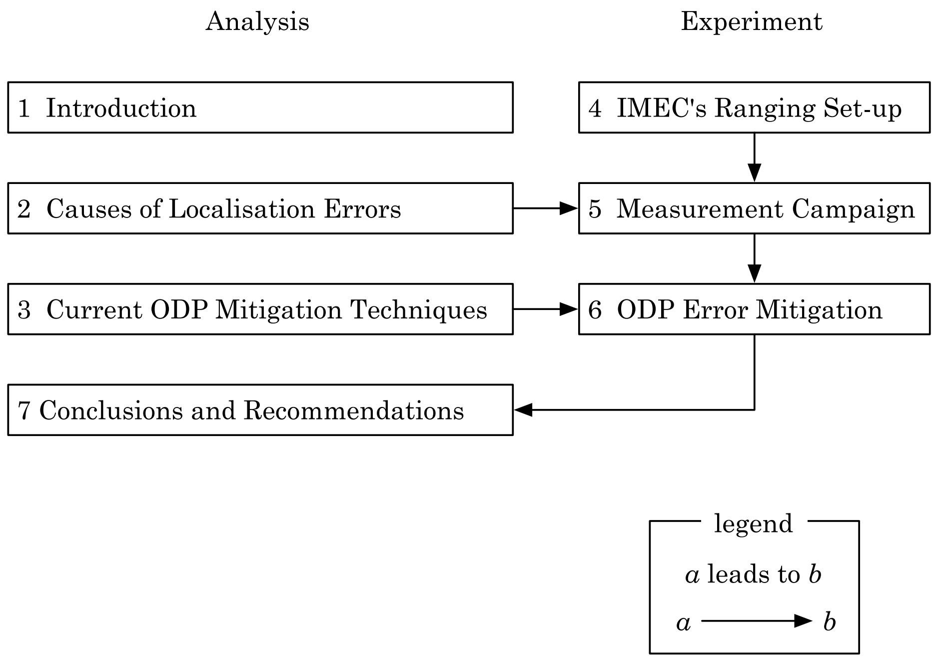


Figure 1.4: Schematic view of the structure of this thesis.

2 Causes of Localisation Errors

In this chapter, we will analyse the relation between localisation errors and Obstructed Direct Path (ODP) conditions. This is a more detailed analysis than what was outlined in Chapter 1.

We start by showing the typical process of localisation based on distance measurements, ignorant of possible ODP conditions. Next, we briefly mention how distance measurements can be obtained using Ultra-Wideband (UWB) signals. Then, we will show how ODP conditions cause a localisation error. Finally, we analyse how ODP conditions come to be in practical situations.

2.1 Lateration

How is localisation performed, if several absolute distance measurements between nodes are given (trilateration)? Let us analyse the classical case of n anchors (of which the position is known) and one target (of which the position needs to be estimated). We start by a geometrical approach with known ranges and illustrate its shortcomings, then we incorporate the fact that only range *estimates* are known, by a probabilistic approach.

2.1.1 Geometric Lateration

Let $\hat{r}_i$ be the distance measurement or *range* between the target and anchor i at location $\vec{x}_i$. If we consider the measurement to be exactly true (i.e. $r_i = \hat{r}_i$), the target must be on the sphere with origin $\vec{x}_i$ and radius $\hat{r}_i$, denoted sphere $(\vec{x}_i, \hat{r}_i)$. In the case of one available range measurement, the location of the target is undetermined, because it can be everywhere on the sphere $(\vec{x}_1, \hat{r}_1)$. In the case of two measurements, the locus of the target is the intersection of the spheres $(\vec{x}_1, \hat{r}_1)$ and $(\vec{x}_2, \hat{r}_2)$, which is, in general, a circle. In the case of three measurements, the locus of the target is the intersection of three spheres, which is, in general, two points. A fourth measurement is necessary to disambiguate between the two points. We conclude that for unambiguous localisation in three dimensional space, four range measurements are needed.

To facilitate visualisation, let us analyse planar, or two dimensional localisation (Figure 1.2a). That is, all nodes are in same plane. If we have one range measurement, we know that the target must be on the intersection of the sphere $(\vec{x}_1, \hat{r}_1)$ and the plane. The resulting locus is a circle. A second measurement adds another circle $(\vec{x}_2, \hat{r}_2)$, resulting in a locus of two points. A third measurement can disambiguate between the two points. We conclude that for unambiguous localisation in two dimensional space, three range measurements are needed.

Note that the last measurement makes the problem overdetermined; that is, there are only two possible range measurements. Still assuming $r_i = \hat{r}_i$, more measurements are neither necessary nor useful. In reality, all measurements are impeded by an error, so $r_i \neq \hat{r}_i$, in general.

2.1.2 Probabilistic Lateration

Therefore, a realistic localisation problem is finding the most probable location of the target, given the measurements:

$$\hat{\vec{x}}_t = \arg\max_{\vec{x}_t} \Pr\left\{\vec{x}_t | \hat{r}_1, \hat{r}_2, \ldots, \hat{r}_n\right\}, \tag{2.1}$$

where $\vec{x}_t$ is the real position of the target and $\hat{\vec{x}}_t$ is the position estimate of the target. We cannot evaluate the direct probability density $\Pr\left\{\vec{x}_t | \hat{r}_1, \hat{r}_2, \ldots, \hat{r}_n\right\}$, but only the inverse probability densities $\Pr\left\{\hat{r}_i | \vec{x}_t\right\}$. To express the direct probability density in these terms, we first apply Bayes' theorem to obtain the inverse probability density:

$$\Pr\left\{\vec{x}_t | \hat{r}_1, \hat{r}_2, \ldots, \hat{r}_n\right\} = \frac{\Pr\left\{\vec{x}_t, \hat{r}_1, \hat{r}_2, \ldots, \hat{r}_n\right\}}{\Pr\left\{\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_n\right\}} = \frac{\Pr\left\{\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_n | \vec{x}_t\right\} \cdot \Pr\left\{\vec{x}_t\right\}}{\Pr\left\{\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_n\right\}}. \tag{2.2}$$

For a given target location $\vec{x}_t$, the range measurements are independent, so we can split the likelihood:

$$\Pr\left\{\vec{x}_t|\hat{r}_1,\hat{r}_2,\ldots,\hat{r}_n\right\} = \Pr\left\{\hat{r}_1|\vec{x}_t\right\}\cdot\Pr\left\{\hat{r}_2|\vec{x}_t\right\}\cdot\ldots\cdot\Pr\left\{\hat{r}_n|\vec{x}_t\right\}\cdot\frac{\Pr\left\{\vec{x}_t\right\}}{\Pr\left\{\hat{r}_1,\hat{r}_2,\ldots,\hat{r}_n\right\}}. \tag{2.3}$$

In lack of a specific model, we assume the target equally likely to be anywhere, i.e. $\vec{x}_t$ is uniformly distributed over all space. Therefore, the probability density $\Pr\left\{\vec{x}_t\right\}$ is constant and infinitesimal. Whatever the value of $\Pr\{\hat{r}_1,\hat{r}_2,\ldots,\hat{r}_n\}$, it is constant in each search for $\hat{\vec{x}}_t$. Consequently,

$$\Pr\left\{\vec{x}_t|\hat{r}_1,\ldots,\hat{r}_n\right\} \propto \Pr\left\{\hat{r}_1|\vec{x}_t\right\}\cdot\Pr\left\{\hat{r}_2|\vec{x}_t\right\}\cdot\ldots\cdot\Pr\left\{\hat{r}_n|\vec{x}_t\right\}. \tag{2.4}$$

As we search the maximum of the probability density (2.1), this proportional product suffices. We must now find the likelihood function $\Pr\left\{\hat{r}_i|\vec{x}_t\right\}$. If we have no model of the range error, we could start by assuming that all errors are independent and equally distributed, with a Gaussian distribution of zero mean and variance σ^2:

$$\begin{aligned} \hat{r}_i &= r_i + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0,\sigma^2), && (2.5)\\ \hat{r}_i &\sim N(r_i,\sigma^2). && (2.6) \end{aligned}$$

In that case, we could find the probability density of the target being somewhere, given the measurements, as follows:

$$\begin{aligned} \Pr\left\{\vec{x}_t|\hat{r}_1,\hat{r}_2,\ldots,\hat{r}_n\right\} &\propto \prod_{i=1}^{n}\Pr\left\{\hat{r}_i = ||\vec{x}_i-\vec{x}_t|||r_i\right\}\\ &\propto \prod_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(\hat{r}_i-||\vec{x}_i-\vec{x}_t||)^2}{2\sigma^2}\right) && (2.7) \end{aligned}$$

Recall that we are only interested in the maximum of this probability density, so any metric that is strict-monotonically increasing with the probability density suffices. We can convert this product of probability densities into a sum by taking the natural logarithm, which is a strict-monotonically increasing function:

$$\Pr\left\{\vec{x}_t|\hat{r}_1,\hat{r}_2,\ldots,\hat{r}_n\right\} \risingdotseq \sum_{i=1}^{n}-\frac{(\hat{r}_i-||\vec{x}_i-\vec{x}_t||)^2}{2\sigma^2} = -\frac{1}{\sigma^2}\sum_{i=1}^{n}(\hat{r}_i-||\vec{x}_i-\vec{x}_t||)^2, \tag{2.8}$$

where $\risingdotseq$ signifies 'is strict-monotonically increasing with'. Instead of trying to find the maximum of this probability density metric, we conventionally try to find the minimum:

$$\begin{aligned} \Pr\left\{\vec{x}_t|\hat{r}_1,\hat{r}_2,\ldots,\hat{r}_n\right\} &\fallingdotseq \sum_{i=1}^{n}(\hat{r}_i-||\vec{x}_i-\vec{x}_t||)^2 && (2.9)\\ \hat{\vec{x}}_t &= \arg\min_{\vec{x}_t}\sum_{i=1}^{n}(\hat{r}_i-||\vec{x}_i-\vec{x}_t||)^2, && (2.10) \end{aligned}$$

where $\fallingdotseq$ signifies 'is strict-monotonically decreasing with'. Note that the unknown σ^2 could be successfully eliminated from the problem. This finding of the location where the sum of the squared errors is minimum is called Least Mean Squared Error (LMSE) optimisation. An example of LMSE localisation using four range measurements is given in Figure 2.1.

As an alternative to using absolute distances, distance *differences* can be used for localisation (multilateration). Let $\hat{d}_i \equiv \hat{r}_{i+1} - \hat{r}_i$. One distance difference $\hat{d}_1$ generates a hyperboloid target locus. In two dimensional localisation, this is a hyperbola on the plane. Another difference $\hat{d}_2$ generates another hyperbola, which should intersect the first in exactly one point. We conclude that two independent distance differences are necessary (so, three anchors) to perform planar multilateration.

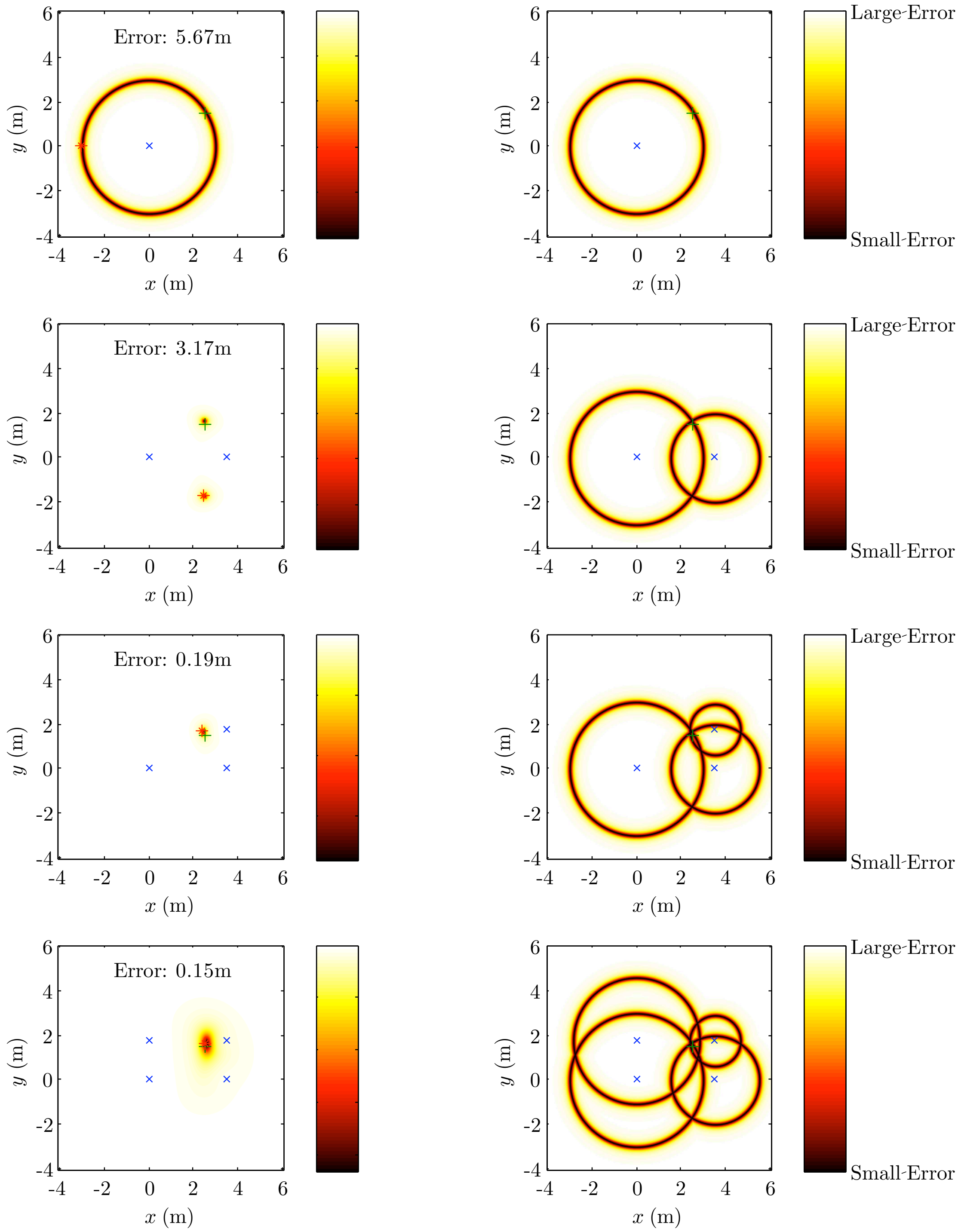


Figure 2.1: Example trilateration steps with four anchors (blue ×s). At the right, the superimposed contributions of each range measurement to the error function are shown. At the left, the cumulative error function is plotted, together with the location of the least error (red star). The real target position is indicated by a green plus. (The measurements were taken with IMEC's transmitter and receiver modules, using leading edge detection ranging.)

If we assume that all distance differences $\hat{d}_i$ are distributed equal and Gaussian, one can follow the same reasoning as above to find the LMSE criterion:

$$\hat{\vec{x}}_t = \arg\min_{\vec{x}_t} \sum_{i=1}^{n-1} \left(\hat{d}_i - (|\vec{x}_{i+1} - \vec{x}_t| - |\vec{x}_i - \vec{x}_t|)\right)^2. \quad (2.11)$$

An example localisation that uses this criterion is shown in Figure 2.2.

2.2 Time Based Ranging

The distance measurements mentioned above could be obtained by one node transmitting a signal, which is answered by the other node (Figure 4.6). The time between sending the signal and receiving the answer is a measure for absolute distance, because the propagation speed is known a priori:

$$t_\text{p} = \frac{r}{c} \quad (2.12)$$

$$\hat{r} = \hat{t}_\text{p} \cdot c, \quad (2.13)$$

where c is the propagation speed of the radio wave[1], $\hat{t}_\text{p}$ is the Time of Flight (ToF), r is the range between the two nodes and hats ($\hat{\cdot}$) indicate estimates. The measured quantity is called Return Time of Arrival (RToA) or Round Trip Time (RTT) and this procedure is called Two-Way Ranging (TWR). Alternatively, all nodes could share a common notion of time. One node sends a signal, together with the current time. The other node receives the signal and subtracts the attached timestamp from the current time. The measured quantity is called ToA, which is a direct measure of the time between sending and receiving the signal (the ToF) and, consequently, a measure of absolute distance.

Alternatively, one node (i.e. the target) transmits a signal, and all the other nodes (i.e. the anchors) register the absolute time of arrival of this signal. The anchors have a shared notion of time amongst themselves, but not with the target. Therefore, only the differences between the arrival times at the anchors contain information. Conversely, the anchors can transmit their signals and the time differences are recorded by the target. Both ways around, these time differences correspond with distance differences; the measured quantities are TDoA.

2.3 Localisation Error Sources

Lateral localisation (as described above) is based only on range or range difference measurements. If all range measurements are error-free ($\hat{r}_i = r_i$), three measurements are enough for planar localisation and LMSE localisation will then yield a perfect location estimate ($\hat{\vec{x}}_t = \vec{x}_t$), see Figure 2.4a. This means that localisation errors must be caused by ranging errors. Conversely, however, ranging errors do not always introduce localisation errors, see Figure 2.4b. Depending on the geometry of the anchors, ranging errors may introduce localisation errors smaller or larger than the range error, see Figure 2.4c-2.4d.

Apparently (Figure 2.4), the geometry determines how large the effect of ranging errors is. The quality of the localisation is conventionally measured using the Geometric Dilution of Precision (GDoP), which is the standard deviation of the localisation error, a metric for precision. The quality of the geometry can be measured with the Relative Geometric Dilution of Precision (RGDoP), which is the ratio between the resulting GDoP and the (equal) standard deviation of all the given ranges [14]. To the author's knowledge RGDoP is always called GDoP in the literature. To disambiguate between the two meanings above, this report uses the term RGDoP. In practical applications, the geometry is given by the user, so we do not consider it a designable

[1] We use the propagation speed of light in vacuum c_0, corrected by the relative permittivity of air: $c = c_0/\sqrt{\varepsilon_r} = 2.99792458 \times 10^8/\sqrt{1.00058986} = 2.99704079 \times 10^8\,\text{ms}^{-1}$. Note that we could only find ε_r for air at 0.9 MHz [12].

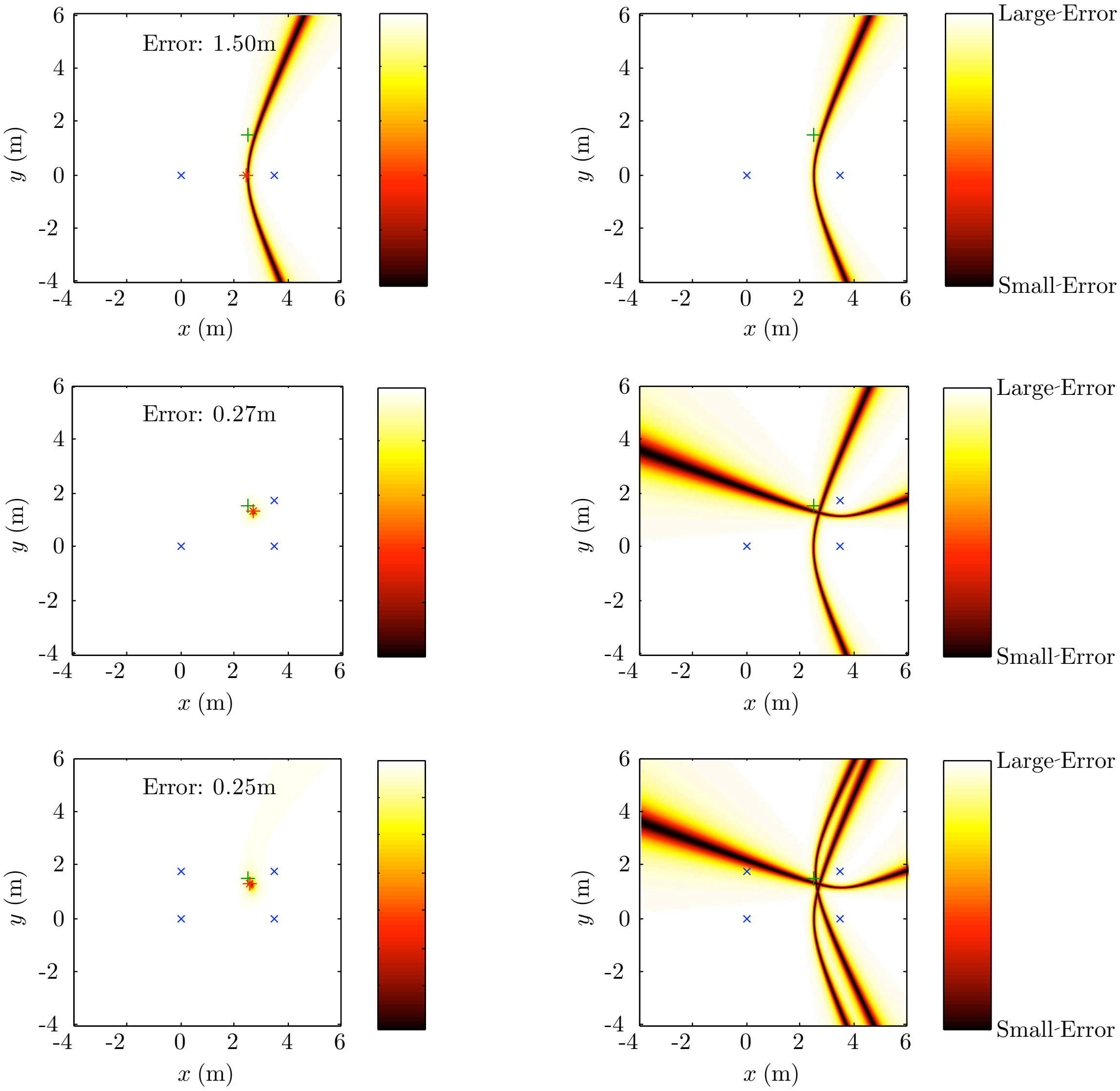


Figure 2.2: Example synthetic multilateration steps with four anchors (blue ×s). At the right, the superimposed contributions of each range measurement to the error function are shown. At the left, the cumulative error function is plotted, together with the location of the least error (red star). The real target position is indicated by a green plus. (The measurements were taken with IMEC's transmitter and receiver modules, using leading edge detection ranging. The ToA range estimates were subtracted to get synthetic TDoA measurements.)

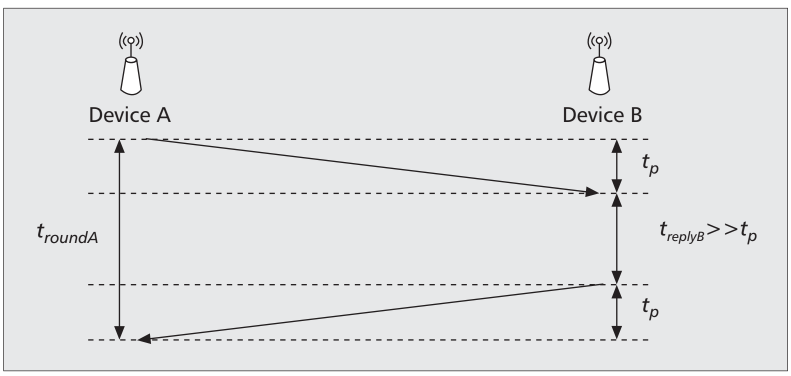
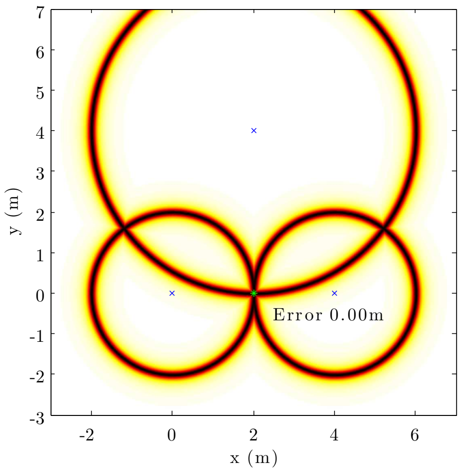
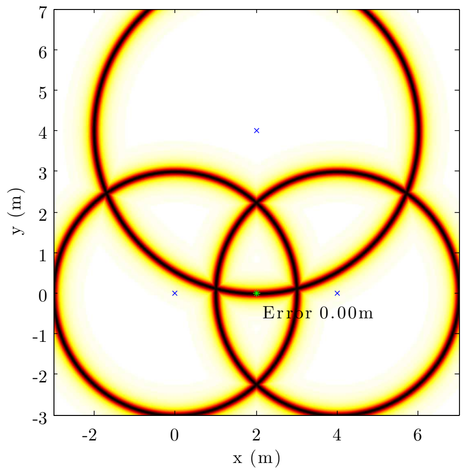


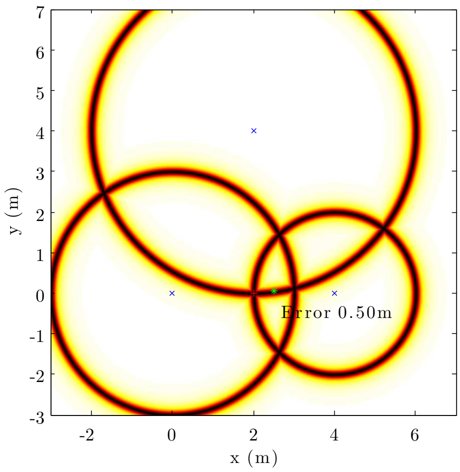
Figure 2.3: Outline of the two-way ranging procedure (from [13, Fig. 2]).



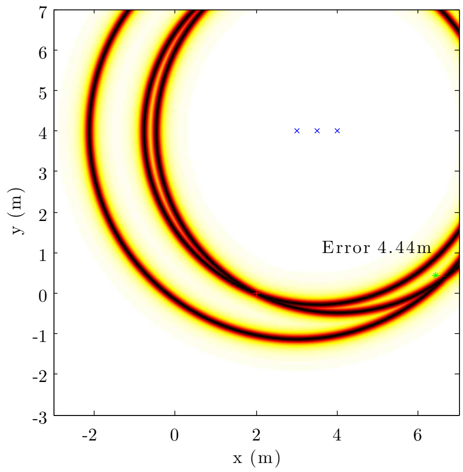
(a) No errors; perfect localisation.



(b) Both anchors at (0,0) and (4,0) have +1 m error; the effects cancel out.



(c) The anchor at (0,0) has +1 m error; the resulting error is of the same order of magnitude.



(d) The anchor at (3,4) has +1 m error; the resulting error is much larger due to the bad geometry.

Figure 2.4: Examples the effect of +1 m range errors on the ToA localisation error. The target node is always at (0,2), depicted with a red plus. The anchors are depicted as blue $\times$s and the estimate is a green star. The colour shows the superimposed contributions to the squared error from the different anchors (there is no meaning in the thickness of the circles).

parameter in this research project. From here on out, we will take the geometry for granted and try to minimise the ranging error.

Then what is causing time-based ranging errors? Sources of ranging errors can be found in propagation, interference and timing [11]. The complete tree of cause and effect is broken-down in Figure 2.5. Let us briefly review the leftmost sources of error and potential counter-measures as explained in [11]. Thermal noise is introduced in all electronic subsystems along the chain, and causes uncertainty in the moment of detection (Figure A.6a). It can be mitigated by better electronic design to a certain extent. In-band interference has comparable effects, but cannot be mitigated like thermal noise[2]. Multipath is the phenomenon of multiple time-shifted copies of the transmitted signal arriving at the receiver, due to reflections. The receiver must take care to detect just the original, because its arrival time corresponds to the length of the Direct Path (DP), which we want to measure. This is a non-trivial but possible task, which becomes harder when the DP is obstructed, thereby attenuated and/or delayed. The next category of errors is timing, that is: the time reference of transmitter and receiver may be different at any given time. Modelling the clocks of both as a linear function of the real time, they can differ in slope (clock drift or frequency offset) and time offset. Time offset is important in ToA localisation, where all nodes (both anchors and target(s)) need to have the same notion of time. Using TDoA measurements, the effect of this time offset between target and anchors is canceled, but the anchors need a shared notion of time. In RToA or TWR, no common notion of time is needed at all. Clock drift is typically encountered in smart dust, where low-cost oscillators are applied. The effect of this clock drift can be partially countered by Symmetric Double-Sided Two-Way Ranging (SDS-TWR) [15, §5.5.7.1]. Real clocks also have a non-deterministic term, called *jitter*, which can be partially cancelled by time averaging. The necessary averaging time depends on the jitter spectrum. Finally, interference can also cause uncertainty in the time detection and cause ranging errors.

2.4 Obstructed Direct Paths

From all these sources of error, the ODP is widely regarded as a significant source [16]. Let us see how this condition can arise in practical situations by means of Figure 2.6. The first channel, between Tx and Rx1, consists of two paths, of which the reflected path is weaker because it propagated over a longer distance[3] and incompletely reflected on a wall. Therefore, the direct path is the strongest and therefore this channel is classified as DDP [11]; the receiver should take care to detect this first path. As seen in channel Tx-Rx2, an obstacle can attenuate the direct path to such an extent, that another path becomes stronger. Therefore, this channel is classified as NDDP. Thick obstacles, such as walls, are known to cause an additional delay, with a positive ranging error of about the thickness of the obstacle [17]. Note well that even if this attenuated direct path is detected, it may be delayed because of propagation through the thick obstacle. Depending on the receiver sensitivity, one or more obstacles can attenuate the direct path so much, that it goes undetected (e.g. Tx-Rx3), which we will call UDP. As for propagation, DDP range measurements are generally good, because the assumption of the light speed holds. The accuracy of range measurements in NDDP depends on the delay introduced by the obstacle. UDP range measurements are positively biased; depending on the delay of the first detectable multipath component, the bias will be larger or smaller. We classify NDDP and UDP channels as ODP.

Note that, instead of DDP and ODP, the literature often speaks about line-of-sight (LOS) and non-line-of-sight (NLOS). Literally, these terms speak about DDP and ODP for visible light. It is useful to distinguish between DDP and ODP for the band of interest and for visible light, because electromagnetic waves propagate differently at different frequencies, see Figure 2.7.

[2] Actually, interference mitigation is indicated as a relevant research topic by [11, §VII].

[3] Of course, it is not the distance that attenuates a signal, but the area over which the energy is spread.

For example, if there is a line-of-sight between a transmitter and receiver antenna, there still may be an ODP condition, depending on the antenna pattern. Conversely, if there is no line-of-sight, the obstacle may be transparent at the used radio frequency.

2.5 Conclusion

We have seen that localisation errors are caused by ranging errors, modulated by bad geometry. We regard ODPs as the main source of ranging errors; however, we choose to see the environment, including obstacles, as the choice of the user. We cannot, therefore, remove this source of error. Consequently, we choose to try and mitigate the effects of the obstructed DP in the sequel.

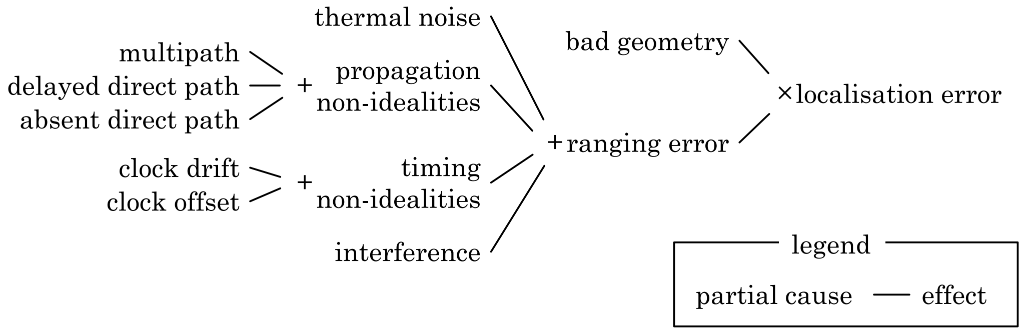


Figure 2.5: Causal breakdown of localisation errors, when using time-based ranging, after [14; 11]. The × sign symbolises that partial causes multiply to an effect, whereas the + sign symbolises additional partial causes.

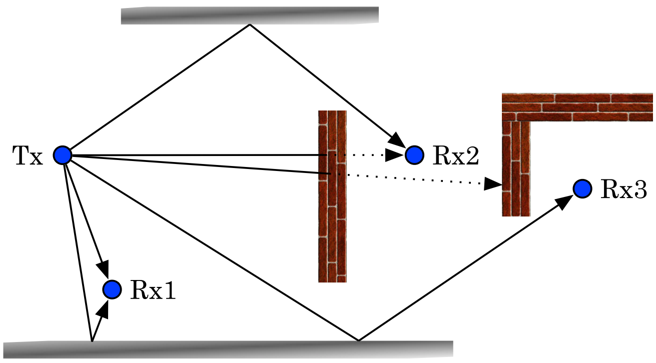


Figure 2.6: Possible multipath propagation scenario, after [11, Fig. 1]. We classify channel Tx-Rx1 as Dominant Direct Path (DDP), channel Tx-Rx2 as NonDominant Direct Path (NDDP) and channel Tx-Rx3 as Undetected Direct Path (UDP).

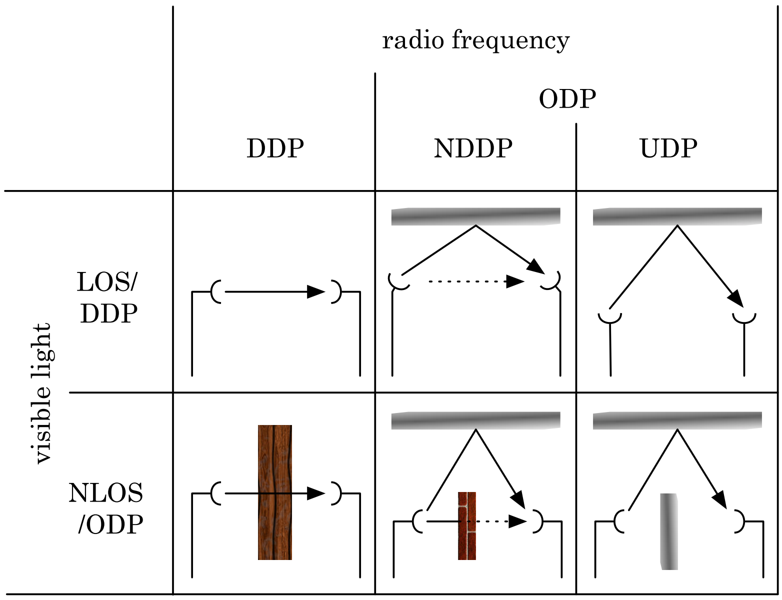


Figure 2.7: Channels are classified in DDP and ODP depending on the frequency band of interest.

3 Current ODP Mitigation Techniques

In this chapter, we will look at published methods of mitigating localisation errors in Obstructed Direct Path (ODP) conditions.

We start by reviewing published attempts to detect ODP conditions. Then we will review the attempts to decrease localisation errors, using ODP information. We conclude with a summary and decision on the sequel.

3.1 Detect ODP Conditions

A recent (2007) paper classifies the ways that localisation errors in ODP conditions are mitigated as follows [18]:

1 Detect ODP conditions based on a single Channel Impulse Response (CIR) measurement. [19; 20; 21; 22]
2 Detect ODP by tracking the range estimates through time.
3 Detect ODP by tracking the shape of CIRs through time. [18]
4 Detect ODP by tracking the position estimates through time. [23; 24]

Note that the above-mentioned methods are listed in order of increasing of complexity. For example, all but the first category need to track measurements and estimations for a while, before giving reliable output (before the channel can duly be considered time-variant). Furthermore, they are not that robust; if a node is switched on in an ODP condition, it might not be possible to reliably detect this condition. Smart dust nodes may be turned on only briefly to save energy. As a result, they are unable to track CIRs nor range nor location through time. Therefore, we preliminary disqualify all tracking methods and we will focus on CIR-based detection.

What CIR properties, or *features* can we expect to correlate with ODP conditions? The general idea is that the statistics of scattered paths do not change, while the Direct Path (DP) is attenuated or absent (undetected), see Figure 3.1. Of course, upon receipt of a CIR, we do not know what is DP and what is the rest. Therefore, we have to judge the CIR as a whole by means of features. We gathered features from both channel modelling and localisation research and discuss them below, ordered by popularity. Finally, we discuss the methods published to judge the channel conditions from the feature values.

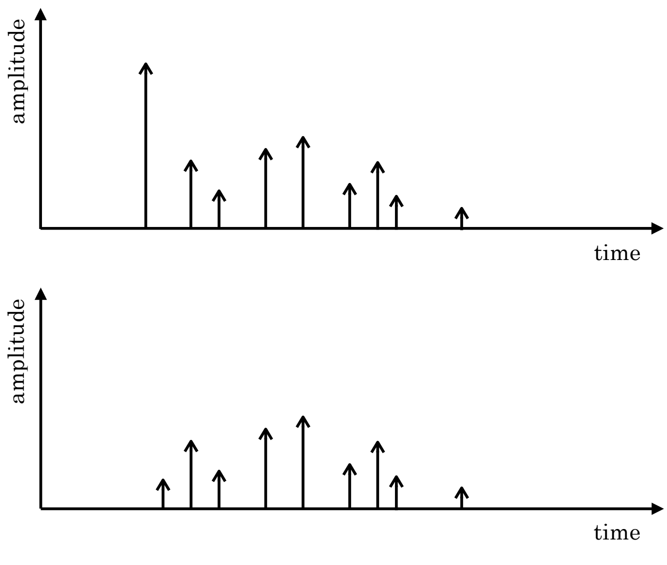


Figure 3.1: Principle of ODP detection: the statistics of the scattered paths are equal in DDP (upper graph) and ODP conditions (lower graph).

Note that in practical applications, a measurement of the CIR is not always available. (A CIR $h(t)$ is the linear response of the air channel to a Dirac delta excitation.) Often, the cross-correlation between the received baseband signal and the known transmitted baseband signal is available. This cross-correlation also describes the equivalent baseband response of the transmitter and receiver electronics, as well as the antenna response, denoted $\hat{h}_b(t)$. We will still use this cross-correlation as an estimate of the equivalent baseband CIR. In a digital system, this signal is time discrete, hence denoted Voltage Delay Profile (VDP) $\hat{h}_b[n]$, with $n \in [1, N]$. In incoherent receivers, the VDP is available only as magnitude, and is denoted $|\hat{h}_b[n]|$ (e.g. Figure 3.2). Taking the square gives the Power Delay Profile (PDP) $|\hat{h}_b[n]|^2$.

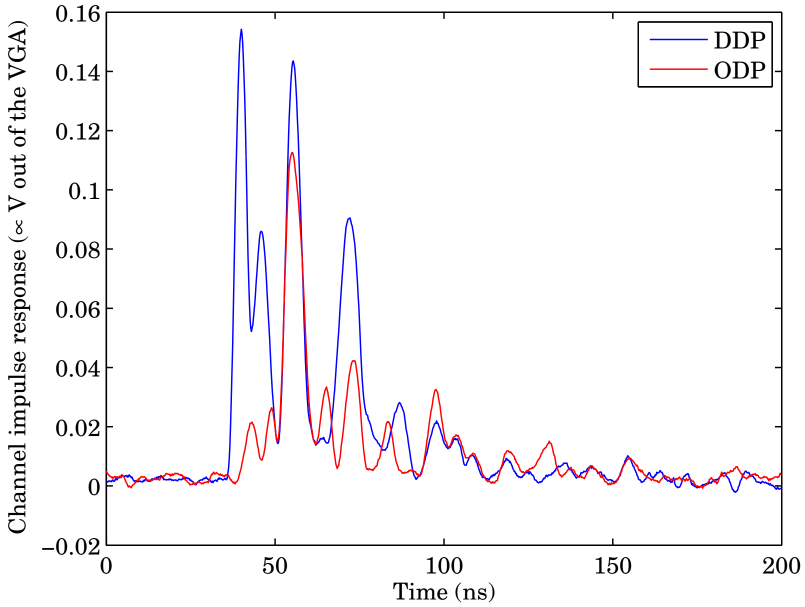


Figure 3.2: Voltage delay profile $|\hat{h}_b[n]|$ of an unobstructed range measurement between two antennas at 1 m in blue. The same measurement is repeated with a metal sheet obstructing the DP, plotted in red.

3.1.1 Mean Excess Delay and RMS Delay Spread

Two related features measure the temporal distribution of the power delay profile. The first feature is the mean excess delay τ_{MED}, which is the centre of mass of the power delay profile, with respect to the leading edge. The second feature is the Root Mean Squared (RMS) delay spread τ_{RMS}, which is the RMS width of the power delay profile around τ_{MED}. In ODP conditions, the DP will be attenuated, thereby shifting the centre of mass of the PDP to the right. If the leading edge of the DP is still detectable, this will make the τ_{MED} higher. Even if the leading edge is not correctly detected, we can still expect the τ_{MED} to be higher, because the scattered paths are more spread out in time than one DP and scattered paths. Similarly, the τ_{RMS} will be higher if the DP is absent or attenuated, because the energy in the scattered paths is more spread out than that in the DP.

The RMS delay spread is a popular parameter to describe channels. It is used to evaluate channel models [25; 26; 27; 28; 29; 30] and is also applied in many ODP detection schemes [19; 20; 31; 21]. The related mean excess delay is also often used in channel models [25; 26] and used in some ODP detection schemes [19; 21].

Both metrics can be calculated from the VDP $\hat{h}_b[n]$ as follows:

$$\tau_{\text{MED}} = \frac{\sum_{n=1}^{N} |\hat{h}_b[n]|^2 \cdot \left(\frac{n-1}{f_s} - t_0\right)}{\sum_{n=1}^{N} |\hat{h}_b[n]|^2} \quad \text{and} \tag{3.1}$$

$$\tau_{\text{RMS}} = \sqrt{\frac{\sum_{n=1}^{N} |\hat{h}_b[n]|^2 \cdot \left(\frac{n-1}{f_s} - t_0 - \tau_{\text{MED}}\right)^2}{\sum_{n=1}^{N} |\hat{h}_b[n]|^2}}, \tag{3.2}$$

where f_s is the sample rate and t_0 is the start of the first apparent path. Note that the value of t_0 matters for τ_{MED}, whereas it cancels out during the calculation of τ_{RMS}. This is also intuitive: τ_{MED} is the mean position of the received pulse energy, whereas τ_{RMS} is merely the width of the pulse energy around this position, wherever it may be.

The above formulas can be meaningfully applied over an infinite number of samples of a noiseless channel estimate. However, the measured channel estimate will contain noise. As a consequence, even samples at $t \to \infty$ have a contribution to the mean excess delay and RMS delay spread. As a result, we will overestimate both metrics.

To overcome this problem, we could set an amplitude threshold. For example, [32] and [33] suggest using a threshold at 30 dB below the highest signal component observed. We could also set a threshold based on the measured noise floor of the receiver, or a fixed time delay after the leading edge.

3.1.2 Kurtosis

Similar to the previous features that measure time spread, kurtosis is a measure for amplitude spread. It is used in some ODP detection schemes [19; 21].

Kurtosis is a measure for how peaked the probability density function of a stochastic variable is. We will use this definition (and not, for example, excess kurtosis):

$$\kappa = \frac{\mu_4}{\sigma_4}, \tag{3.3}$$

where μ_4 is the fourth moment about the mean and σ is the standard deviation of the distribution. The kurtosis of ODP sample amplitudes is generally lower than that of DDP channels. Assuming all VDP samples to be equally probable, a sample kurtosis can be meaningfully calculated from the VDP. As suggested by Monte-Carlo channel simulations done in [19], sample kurtosis is a metric that can discriminate ODP well in indoor environments. In [19], the kurtosis κ is estimated from the absolute samples $|\hat{h}[n]|$ of the VDP:

$$\hat{\kappa} = \frac{\sum_{n=1}^{N} \left(|\hat{h}_b[n]| - \overline{|\hat{h}_b[n]|}\right)^4}{\left(\sum_{n=1}^{N} \left(|\hat{h}_b[n]| - \overline{|\hat{h}_b[n]|}\right)^2\right)^2} \tag{3.4}$$

It can be verified that this is a biased estimator of the sample kurtosis, which is allowable for $N \gg 3$ [34, Eq. 9,11].

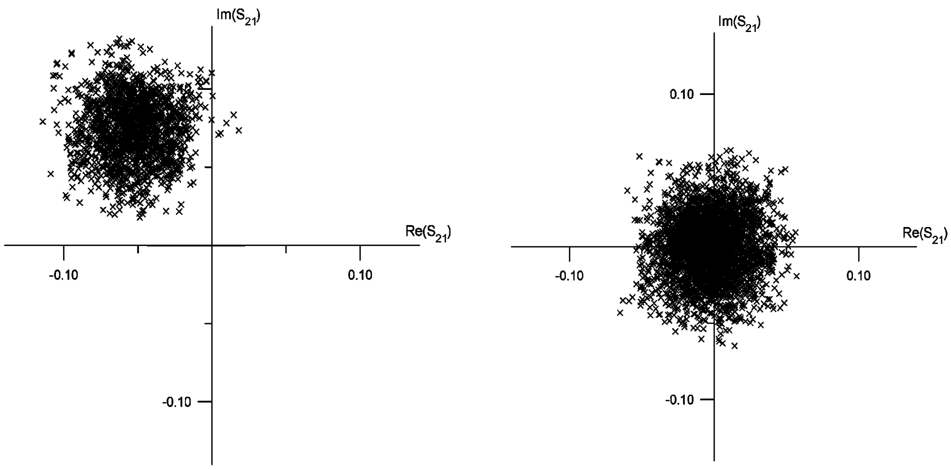
3.1.3 Ricean K-factor

The Ricean K-factor is a often used metric to describe multipath environments for narrowband signals. The K-factor is defined as the ratio between the power in the direct path and in the indirect, scattered paths. It also used in at least one wideband measurement campaign to characterise channels [33]; a low K-factor indicates an ODP condition.

To understand the origin of the K-factor, let us first consider narrowband channels. Narrowband channels can be described as a frequency-flat complex transfer gain S_{21}. A realistic channel consist of a DP and scattered multipath components [35]:

$$S_{21} = DP + M, \tag{3.5}$$

where DP is the gain of the direct path and M is the sum of the scattered multipath components. DP is considered deterministic, as it is determined by the antenna separation according to Friis transmission equation. M is considered to be zero-mean circular symmetric complex Gaussian-distributed ($M \sim CN(0, \sigma_M)$), as it is determined by casual changes in the environment. Examples of measured channel gains in a varying environment are shown in Figure 3.3. In this case, one could estimate the K-factor by dividing the center of the cloud (measure for DP) by some measure of the size of the cloud (M).



(a) Strong direct path (high K-factor). **(b)** Weak direct path (low K-factor).

Figure 3.3: Channel transfer gain measurements taken in an reverberation chamber [36, Fig. 3].

In the case of Ultra-Wideband (UWB) we cannot describe the channel by one phasor. Furthermore, we only have one channel realisation, so we cannot calculate standard deviations such as suggested above. Still, we could try to discriminate the power in the first and in the following paths of the PDP. Depending on the receiver bandwidth, more or less multipaths can be distinguished ('resolved') in the PDP. Different multipaths may be overlapping in the PDP, but algorithms such as CLEAN can obtain a list of discrete paths [25; 37; 38]. We can then divide the power in the first received path over the remaining paths, to get an estimate of the K-factor.

A simpler method, used in [33] according to a conversation with the first author, is to divide the height of the global maximum of the PDP by all other (non-global) local maxima. Implicitly, this is interpreting the global maximum as the most probable first path, and all other local maxima as the scattered paths.

If we can trust the PDP to start at the first detectable path, it makes more sense to divide the power in the first so many samples over the remaining power. Implicitly, we interpret the first so many samples as the most probable first path, and everything else as scattered paths. This method uses more samples and is therefore less susceptible to noise.

3.1.4 Number of paths

Some channel models use the detectable number of paths as a metric for the temporal spread of the channel [25; 26]. More detectable paths indicate an ODP condition.

To estimate the detectable number of paths, a CLEAN algorithm can be used and the number of detected components can be counted [25]. However, the outcome of this method depends on the Signal to Noise Ratio (SNR) of the measurement, the implementation of the CLEAN algorithm and the quality of the templates.

Therefore, we deem the $NP_{10\,\mathrm{dB}}$ and $NP_{85\%}$ more robust and more general [26]. $NP_{10\,\mathrm{dB}}$ is defined as the number of paths in all components within 10 dB below the strongest component. $NP_{85\%}$ is the number of paths that contain 85% of the energy of all of the received signal.

For smart dust applications, we could try to eliminate computationally complex operations such as CLEAN. Analogous to $NP_{10\,\mathrm{dB}}$, we could sum the duration of (i.e. count) all PDP samples at 10 dB below the strongest path. Analogous to $NP_{85\%}$, we could sort all PDP samples and measure the cumulative duration of the first samples that contain 85% of the total energy.

3.1.5 Rise time

As suggested in one ODP detection scheme [21], the rise time of the received pulse may be used as metric as well. A short rise time indicates a clean DP, a long rise time might indicate an ODP condition. The rise time can be defined as follows [21]:

$$\begin{aligned} t_{\mathrm{rise}} &= t_H - t_L, \qquad \text{where} && (3.6)\\ t_L &= \min\left\{t : |\hat{h}_b(t)| \geq \alpha\sigma_n\right\} \qquad \text{and} && (3.7)\\ t_H &= \min\left\{t : |\hat{h}_b(t)| \geq \beta\max\left(|\hat{h}_b(t)|\right)\right\}, && (3.8)\end{aligned}$$

where σ_n describes the noise floor, and α and β are picked empirically to describe the rising edge ([21] used $\alpha = 6$ and $\beta = 0.6$). Note that the signal voltage $|\hat{h}_b|$ instead of the power $|\hat{h}_b|^2$ is used.

The values α and β put a constraint on the dynamic range as follows. Assume that $\hat{h}_b(t)$ is time-continuous and demand that $t_{\mathrm{rise}} > 0$:

$$\begin{aligned} \alpha\sigma_n &< \beta\max\left(|\hat{h}_b(t)|\right) && (3.9)\\ \frac{\alpha}{\beta} &< \frac{\max\left(|\hat{h}_b(t)|\right)}{\sigma_n} && (3.10)\\ \left(\frac{\alpha}{\beta}\right)^2 &< \frac{\max\left(|\hat{h}_b(t)|^2\right)}{\sigma_n^2} = \mathrm{SNR}_{\mathrm{lin}} && (3.11)\\ 20\log_{10}\left(\frac{\alpha}{\beta}\right) &< 10\log_{10}\left(\frac{\max\left(|\hat{h}_b(t)|^2\right)}{\sigma_n^2}\right) = \mathrm{SNR}_{\mathrm{dB}}, && (3.12)\end{aligned}$$

which evaluates to a minimum SNR of 20 dB. Depending on the received signal, it may or may not be possible to evaluate the rise time.

3.1.6 Fitted exponential decay

Some channel models describe the PDP as an exponential decay [39; 30]. Although not used in any ODP detection scheme to the author's knowledge, the exponent may be used to describe the temporal spread of the PDP. That is, a strong exponential decay (high negative time coefficient) indicates a free DP, whereas a mild exponential decay (low negative time coefficient) may indicate an ODP.

Practically, when using this model to describe a received PDP, we need to fit an exponential curve to the PDP using a Least Mean Squared Error (LMSE) criterion [39]:

$$|h_{\mathrm{fit}}(t)|^2 \propto \exp\left(-\frac{t}{\tau_{\mathrm{fit}}}\right), \tag{3.13}$$

where τ_{fit} is the fitting parameter. τ_{fit} is suggested to have a proportional relation to τ_{RMS} [39], so τ_{fit} may not contain extra information, given that τ_{RMS} is already known.

3.1.7 Coherence Bandwidth

Another interesting feature that could be used is the coherence bandwidth, because it seems to correlate with the channel condition [27, Tab. 1]. Coherence bandwidth is a measure for how wide a bandwidth is conducted through the channel with comparable attenuation and phase shift. A high coherence bandwidth is indicative of an unobstructed DP.

In general, the coherence bandwidth can be calculated from the frequency domain representation of the (baseband) channel. This representation can be obtained by taking the Fourier transformation of the voltage delay profile $\hat{h}_b(t)$. As mentioned before, this complex-valued function is not always available, for example in incoherent receiver structures. Fortunately, under some assumptions (uncorrelated scatterers, to be precise), it is allowed to calculate the frequency domain representation based on the power delay profile $|\hat{h}_b(t)|^2$ [40].

In the latter case, the coherence bandwidth can be calculated by first calculating the Frequency Correlation Function (FCF) [40, Eq. 2.11]:

$$\mathrm{FCF}(\Delta_\omega) = \int_{-\infty}^{\infty} |\hat{h}_b(t)|^2 \exp\left(-\mathrm{j}\Delta_\omega t\right)\, \mathrm{d}t, \tag{3.14}$$

where Δ_ω is frequency difference in radians per second. The coherence bandwidth is defined as the first frequency difference where the normalised FCF drops below a certain value. Commonly used coherence bandwidths are $B_{\mathrm{coh},0.9}$ and $B_{\mathrm{coh},0.5}$, being the frequency difference at which the normalised FCF first drops below 0.9 and 0.5, respectively [27].

According to analytic [40, Eq. 2.14] and experimental models, there is a fixed relation between coherence bandwidth and delay spread, namely:

$$B_{\mathrm{coh},0.5} \approx \frac{0.28}{\tau_{\mathrm{RMS}}}. \tag{3.15}$$

Therefore, measuring the coherence bandwidth may not yield information if the RMS delay spread is already known. However, in [27], B_{coh} is measured and used alongside τ_{RMS} and their measured $B_{\mathrm{coh},0.5} \cdot \tau_{\mathrm{RMS}}$ varies between 0.21 and 4.45, so we will still measure both.

3.1.8 Path Loss Exponent

In Time of Arrival (ToA) measurements, both the CIR and a range estimate are available simultaneously. We could try to think of a metric that takes advantage of the combination of both pieces of information.

In an unobstructed DP channel, we expect the power in the DP to be proportional to the inverse squared distance. In the case of an ODP, this relation may be different. For example, in the case of an NonDominant Direct Path (NDDP), the power will lower because the obstacle introduced significant attenuation but little excess delay. No published results on this method exist, to the author's knowledge.

Let r denote the actual range and $E_{\mathrm{DP,1m}}$ the energy in the direct path at 1 m range. We expect the received direct path energy to be:

$$E_{\mathrm{DP}} = E_{\mathrm{DP,1m}} \cdot \frac{1}{r^2}, \tag{3.16}$$

Conversely, if we can estimate the energy $\hat{E}_{\mathrm{DP}}$ in the direct path and the range $\hat{r}$ from the CIR, we can make a quick-and-dirty estimate of the apparent path loss exponent $\hat{\lambda}$:

$$\begin{aligned} \hat{E}_{\mathrm{DP}} &= E_{\mathrm{DP,1m}} \cdot \frac{1}{\hat{r}^{\hat{\lambda}}} && (3.17) \\ \frac{E_{\mathrm{DP,1m}}}{\hat{E}_{\mathrm{DP}}} &= \hat{r}^{\hat{\lambda}} && (3.18) \\ \hat{\lambda} &= \log_{\hat{r}}\left(\frac{E_{\mathrm{DP,1m}}}{\hat{E}_{\mathrm{DP}}}\right) = \frac{\ln(E_{\mathrm{DP,1m}}/\hat{E}_{\mathrm{DP}})}{\ln(\hat{r})} && (3.19) \end{aligned}$$

Because of the non-linear operations, $\hat{\lambda}$ will be a biased estimator of the real path loss exponent λ. In NDDP conditions, the range estimate will typically have little positive bias, but the DP will be significantly attenuated. As a result, the apparent path loss exponent $\hat{\lambda}$ will be higher than 2. Conversely, in Undetected Direct Path (UDP) conditions, the first detectable path may be actually the constructive sum of multiple paths (for example, in the middle of a reflective corridor). In that case, the DP appears to be stronger than what could be expected from the ToA, resulting in an apparent path loss exponent smaller than 2. Therefore, we expect this apparent path loss exponent $\hat{\lambda}$ to deviate in either direction from 2 in ODP conditions.

3.1.9 Combining features

Recall the goal of evaluating features: we would like to know if a given range measurement is trustworthy, based on the CIR. We suppose that ODP channel conditions are the main source of error, so it is useful to know whether or not the direct path is obstructed during the given measurement (DDP or ODP). If f_i denotes feature i out of n, we want to know which probability[1] is greater, that is, whether

$$\Pr\left\{\text{DDP}|f_1, f_2, \ldots, f_n\right\} < \Pr\left\{\text{ODP}|f_1, f_2, \ldots, f_n\right\}. \tag{3.20}$$

From channel modeling, we generally only know the inverse; the probability density $\Pr\left\{f_i|\text{DDP}\right\}$ and $\Pr\left\{f_i|\text{ODP}\right\}$. Therefore, we use Bayes' theorem to invert the probability:

$$\Pr\left\{\text{DDP}|f_1, f_2, \ldots, f_n\right\} = \frac{\Pr\left\{f_1, f_2, \ldots, f_n|\text{DDP}\right\} \cdot \Pr\{\text{DDP}\}}{\Pr\left\{f_1, f_2, \ldots, f_n\right\}}, \tag{3.21}$$

mutatis mutandis for ODP. If the feature values f_i are independent, given the channel condition – a questionable assumption – we can factor (3.20) as:

$$\begin{aligned} & \Pr\left\{f_1|\text{DDP}\right\} \cdot \Pr\left\{f_2|\text{DDP}\right\} \cdot \ldots \cdot \Pr\left\{f_n|\text{DDP}\right\} \cdot \frac{\Pr\{\text{DDP}\}}{\cancel{\Pr\left\{f_1, f_2, \ldots, f_n\right\}}} \\ < \quad & \Pr\left\{f_1|\text{ODP}\right\} \cdot \Pr\left\{f_2|\text{ODP}\right\} \cdot \ldots \cdot \Pr\left\{f_n|\text{ODP}\right\} \cdot \frac{\Pr\{\text{ODP}\}}{\cancel{\Pr\left\{f_1, f_2, \ldots, f_n\right\}}}. \end{aligned} \tag{3.22}$$

Or, put differently:

$$\frac{\Pr\left\{f_1|\text{DDP}\right\}}{\Pr\left\{f_1|\text{ODP}\right\}} \cdot \frac{\Pr\left\{f_2|\text{DDP}\right\}}{\Pr\left\{f_2|\text{ODP}\right\}} \cdot \ldots \cdot \frac{\Pr\left\{f_n|\text{DDP}\right\}}{\Pr\left\{f_n|\text{ODP}\right\}} < \frac{\Pr\{\text{ODP}\}}{\Pr\{\text{DDP}\}} \tag{3.23}$$

where $\Pr\{\text{ODP}\}$ and $\Pr\{\text{DDP}\}$ are the prior probabilities of the ODP and DDP conditions; in lack of a use case, one could assume them to be equal. The factors $\Pr\left\{f_i|\text{DDP}\right\}/\Pr\left\{f_i|\text{ODP}\right\}$ are the evidence for DDP generated by each feature value f_i. To the author's knowledge, there are no analytical models that predict this evidence, for any feature. There are however simulational and experimental studies that evaluated the statistical properties of each feature under DDP and ODP conditions, see the literature references in Section 3.1.1–3.1.7, for example Figure 3.4. With such a histogram or Probability Density Function (PDF), it is possible to evaluate (3.23).

The independence assumption is questionable, because the features measure similar phenomena. Therefore, more advanced combination techniques exist, that take the dependencies between features into account. Examples are swarm enabled learning [22] or training Support Vector Machines (SVMs) [21].

[1] Indeed, the distribution of $\Pr\left\{\text{DDP}, f_1, f_2, \ldots, f_n\right\}$ is *hybrid*: the channel condition (DDP/ODP) is binary discrete, while the feature values f_i are continuous. As a result, $\Pr\left\{\text{ODP}|f_1\right\}$ is a probability, while $\Pr\left\{f_1|\text{ODP}\right\}$ is a probability density.

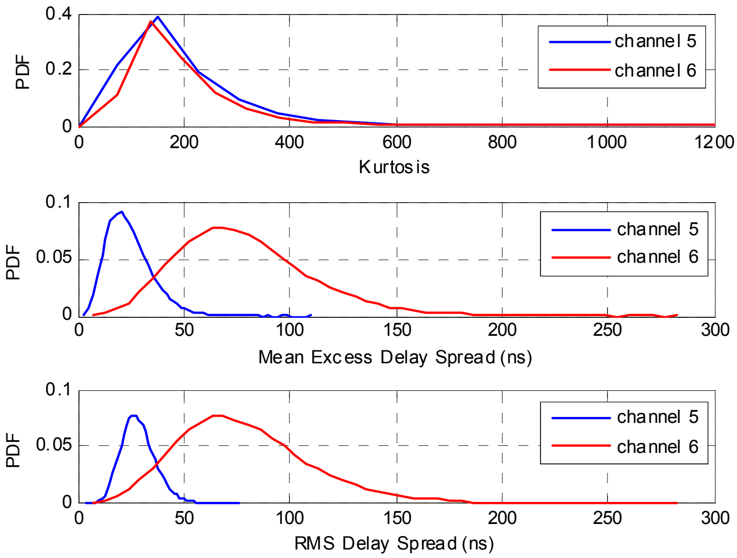


Figure 3.4: Statistical properties of three features in unobstructed DP (channel 5) and ODP (channel 6) conditions in an outdoor environment. Taken from [22, Fig. 3a].

3.2 ODP-aware Localisation

We can also mitigate the effect of ODP measurements by having more robust ways of combining range measurements. We can also try to incorporate the ODP detection results from the ranging stage. Finally, we can localise co-operatively, to 'work around' obstacles.

More robust ways of combining are proposed in the literature, such as using (distributed) Global Likelihood criteria [41], Least Median of Squares (LMedS) criterion [42] or by assuming Rayleigh distributed errors [43]. These methods do not require ODP detection, but assume that a minority of the measurements is erroneous.

If ODP detection information is available, we can treat the range measurements differently. The first strategy is (1) *identify and discard*: just discard the measurements that are classified as ODP, and localise using the remaining measurements. This strategy is straightforward, but does not always give unambiguous location estimated (for example, in the case of two DDP anchors and one ODP anchor). Therefore, we could try to (2) *mitigate*: try to correct the error of the ODP measurements and still use all measurements to localise. This method requires that we have some estimate of the error in an ODP measurement. Furthermore, we know that the corrected range estimate will have a large uncertainty. The resulting range estimate may disambiguate in otherwise ambiguous cases (e.g. 2 DDP/1 ODP), but will deteriorate localisation in general. To overcome this problem, we could try a (3) *hybrid* strategy: if enough range measurements are available that are classified DDP, discard the others. If not, try to correct the ODP range estimates and use them in localisation. This hybrid method is experimentally found to be the best strategy [21].

Finally, the nodes may co-operate, that is, they can act both as target and as anchor. In that case, many range measurements are available between all nodes. These ranges can be combined to localise all nodes, if the positions of only few nodes are given, as illustrated in Figure 3.5. This mode of localisation, called 'co-operative localisation' gives promising results [4; 17]. Especially when ODP information could be incorporated, we can expect this localisation mode to outperform classical localisation with distinct anchors and targets [17]. In applications that have a high density of nodes, co-operative localisation may look 'around' obstacles, as illustrated in Figure 3.6.

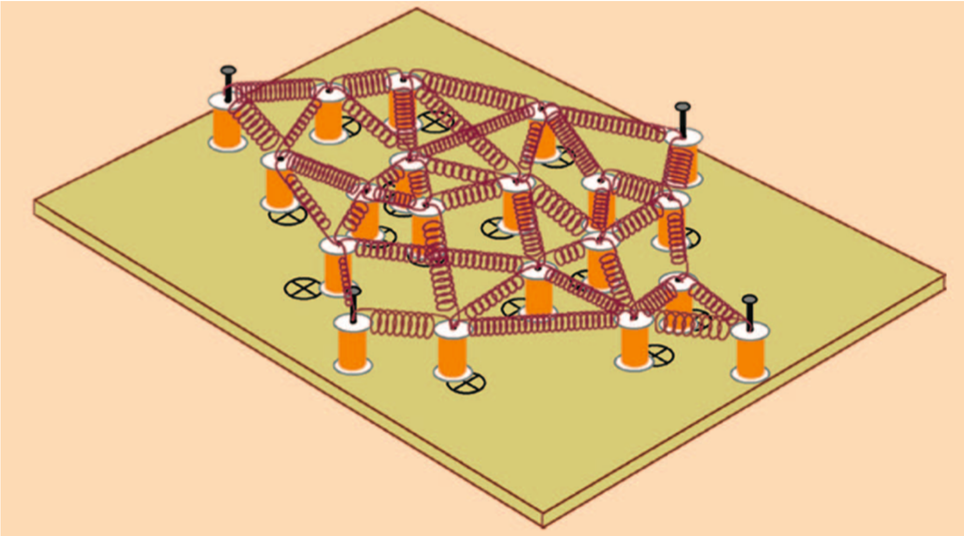


Figure 3.5: Analogy to co-operative localisation of 20 nodes, four of which have an a priori known location. Finding the location of the nodes is analogous to finding the minimum energy state of this system of springs and masses. Every spring has a natural length equal to the measured range between the two connecting nodes. The real location of every node is indicated with ⊗, while the a priori known nodes are masses, nailed to the board. Taken from [6, Fig. 2].

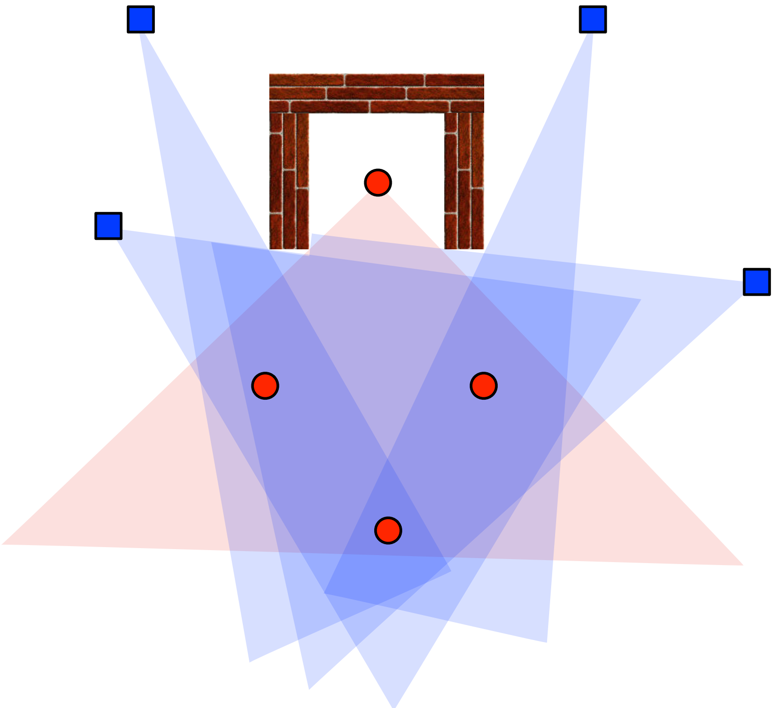


Figure 3.6: Example scenario where co-operative localisation with ODP awareness could be advantageous. The square blue anchors have an *a priori* known location, while the location of the round red nodes is to be determined. There is no line of sight between the obscured node and any of the four anchors. Fortunately, three other nodes are able to see the obscured node, hence are able to localise it.

3.3 Conclusions

In this chapter, we have reviewed the published attempts to mitigate localisation errors due to ODP conditions. First, we have reviewed features of the PDP that are known or expected to correlate with ODP conditions: mean excess delay, RMS delay spread, kurtosis, Ricean K-factor, number of paths, rise time, fitted exponential decay, coherence bandwidth, and path loss exponent. Evaluating any or all of these features could enable ODP detection.

Next, we have seen more robust ways of combining range measurements, with or without ODP detection information. If ODP detection is present, the *hybrid* strategy should be employed. That is, if there are enough measurements, discard the measurements classified as ODP. If not, try to compensate the error in the ODP measurements and use them anyway.

In the sequel, we will implement all features and try to combine them in order to detect ODP conditions. We will then implement this hybrid localisation method and evaluate the improvement in accuracy by means of measurements with IMEC's ranging set-up.

4 IMEC's Ranging Set-up

This chapter gives an overview of IMEC's current ranging set-up and the preparation of this set-up for Obstructed Direct Path (ODP) detection measurements.

We start by presenting an overview of the components of IMEC's ranging set-up, after which we elaborate on each of the aspect of the set-up. First, we give an overview of the used protocol, which is IEEE 802.15.4a. Second, we give an overview of its current software implementation. Third, we outline the current hardware implementation. Finally we indicate the shortcomings of this ranging set-up and how we will cope with them during our measurements.

4.1 Set-up Overview

IMEC is working step-by-step on an integrated Ultra Low Power (ULP) Ultra-Wideband (UWB) transceiver. That is, the transmitter-receiver chain is first designed on paper, using mathematical analysis, and then tested in MATLAB. Next, elements of this system are prototyped with appropriate technology; digital processing is prototyped using Field Programmable Gate Arrays (FPGAs), analog processing is prototyped using discrete components. Finally, after groups of subsystems have been validated using prototypes, they are integrated on mixed-signal Application Specific Integrated Circuits (ASICs), until all subsystems are integrated. This is a controlled way of realising complex electronic systems.

At the start of performing the practical research of this thesis (June 2010), the state of the ULP UWB tranceiver at IMEC was as illustrated in Figure 4.1 and photographed on Figure 4.2. A relatively autonomous transmitter was already realised as ASIC, integrating both the digital transmitter and the analogue modulator. As for the receiver, only the analogue demodulator was realised on silicon. The digital receiver was still prototyped in MATLAB, providing offline processing.

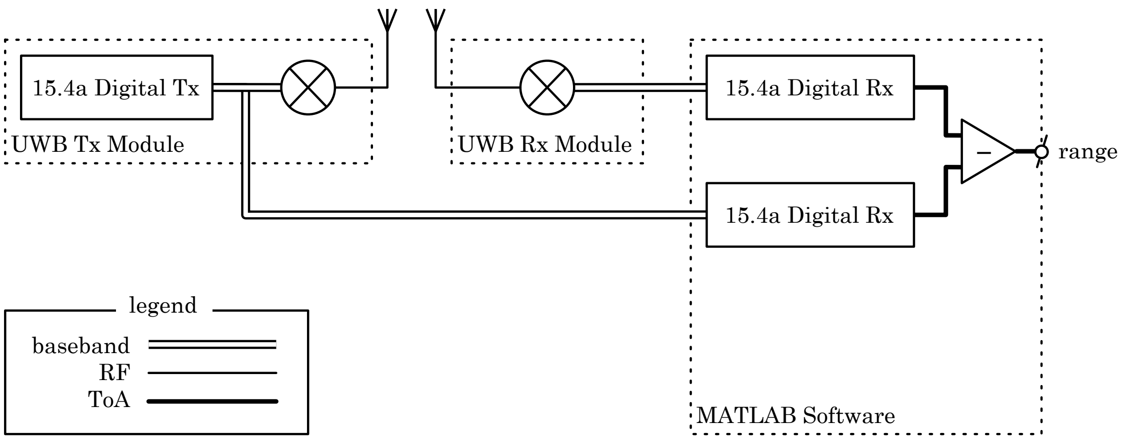


Figure 4.1: Overview of the components of IMEC's UWB ranging set-up.

This prototype set-up already demonstrates ranging capabilities; both digital receivers determine the Time of Arrivals (ToAs). The difference is the Time of Flight (ToF) of the radio signal, after being compensated for cable delays. When multiplied with the speed of light, a range estimate can be given.

IMEC strives to create a standards-compliant transceiver, so the radio protocol used by the transmitter and receiver is IEEE 802.15.4a, sometimes abbreviated as '15.4a'. Both the 15.4a digital transmitter and receiver are modelled in MATLAB, but as shown above, only the MATLAB digital receiver is used in the current ranging set-up.

Figure 4.2: Photograph of the ranging set-up, with the major components highlighted. The Power Amplifier (PA) is necessary to boost the transmitter's output power, the oscilloscope is used to capture the baseband signals for MATLAB processing, and the clock generator is necessary to clock the transmitter. A detailed schematic of the set-up is given in Figure 4.11.

4.2 The Protocol: IEEE 802.15.4a

Let us summarise the ranging procedure of IEEE 802.15.4a using the UWB PHY. Some context of the standard is given, but only if it serves to understand the ranging procedure.

IEEE 802.15.4 defines both a Medium Access Control (MAC) and PHY layer for Low-Rate Wireless Personal Area Networks (LR-WPANs). The MAC layer supports both contention and contentionless medium access. The IEEE 802.15.4a amendment adds two more PHYs: narrowband Chirp Spread Spectrum (CSS) and UWB [13]. As the latter is most suited for precision ranging and is used in IMEC's set-up, we will detail out the UWB PHY.

The RF signal of the UWB PHY consists of TeRnary digITs (trits) $\in \{-1, 0, +1\}$, that are encoded as pulses within chip intervals and modulated onto a carrier. The reciprocal of the chip duration T_c is the peak Pulse Repetition Frequency (PRF), the pulse duration is $T_p \leq T_c$. An example of symbolic trits, the corresponding baseband and RF signals are given in Figure 4.3.

These chips are used to convey PHY packets[1]. Every PHY packet (outlined in Figure 4.4) consists of three parts: the Synchronisation HeadeR (SHR), PHY HeadeR (PHR) and the PHY Service Data Unit (PSDU). The SHR starts with a preamble to enable the receiver to adjust its gain and synchronise. Next, the Start of Frame Delimiter (SFD) is transmitted to mark the end of the preamble (therefore sometimes called End Of Preamble (EOP)) and the beginning of useful data. The PHR contains essential information to decode the PSDU. Finally, the PSDU is transmitted, which contains the higher-layer payload.

[1] Notice that in other network standards, the physical layer entity only takes bits from network layer and modulates them onto the medium. Consequently, a synchronisation preamble is considered part of the *frame*. In IEEE 802.15.4, the physical layer entity actually adds some headers. Hence, a synchronisation preamble is considered part of the *packet*.

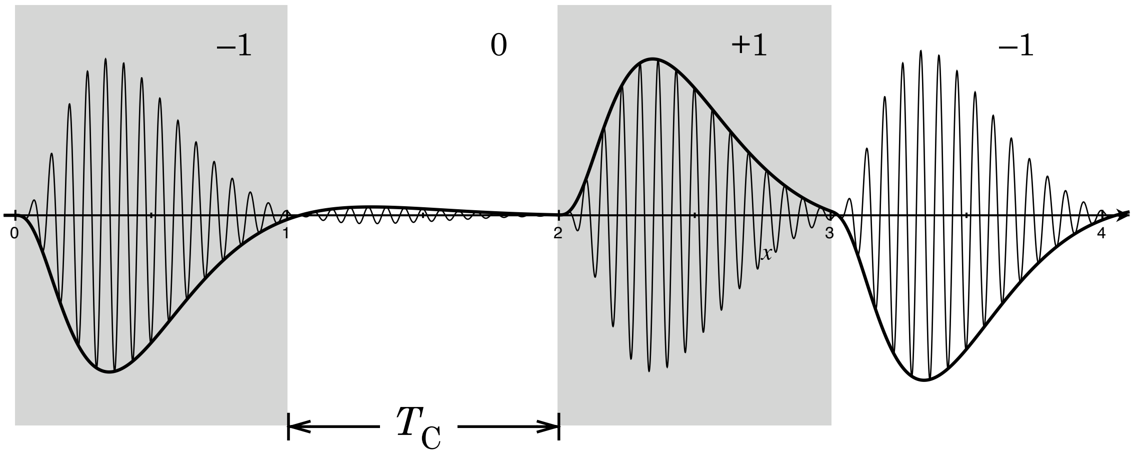


Figure 4.3: Example of an IEEE 802.51.4a UWB baseband signal (thick curve) and RF signal (thin curve). Four chip intervals are plotted, encoding the ternary sequence $\{-1, 0, +1, -1\}$.

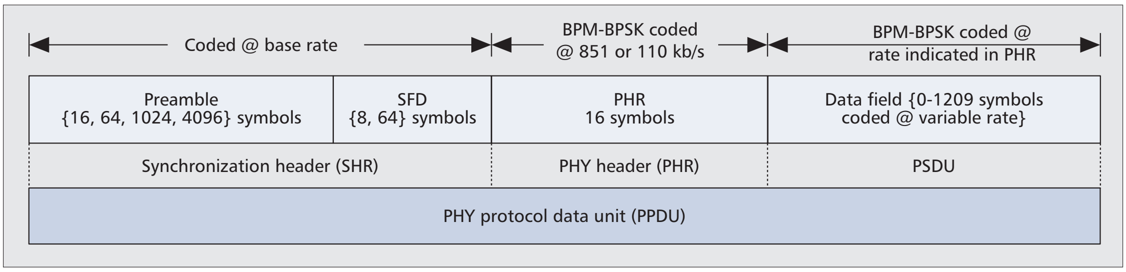


Figure 4.4: Format of the IEEE 802.15.4a UWB packet [13, Fig. 1].

To transmit data during PHR and PSDU, the UWB PHY uses Burst Position Modulation (BPM) to transmit the actual data bits and Binary Phase Shift Keying (BPSK) to transmit redundant parity bits. That is, burst of pulses are transmitted during a certain interval to signal a '0' data bit, and during a different interval to signal a '1' data bit. The polarity of the bursts is modulated to contain a redundant bit (see also Figure 4.5). This way, an incoherent receiver (which can only demodulate BPM information) can receive data bits, while a coherent receiver (which can also demodulate BPSK information), may take advantage of the parity bit to conceal errors and effectively increase sensitivity.

The short pulse of $T_p \sim 2\,\text{ns}$ [15, §6.8a.4] has a broad spectrum $\sim 500\,\text{MHz}$. The pulse shape can be chosen anything that correlates well enough with the reference pulse [15, § 6.8a.12.1], typically trying to fill up the available spectral mask. In reality, not one pulse is transmitted, but the pulses are repeated to convey information bits. Let us model this data signal as an impulse train or Dirac comb, i.e. Dirac impulses repeated at fixed intervals T. The convolution of the pulse spectrum with the data spectrum gives the transmitted spectrum, which is a Dirac comb (interval of $1/T$) enveloped by the pulse spectrum. This is bad use of the available spectrum, because the power is not equally spread out under the available spectral mask.

To smooth the spectrum to optimally use the spectral mask, the timing of the burst is randomised (within bounds), as well as the polarity of the chips within the burst. The seed for both pseudo-random processes is transferred in the preamble symbol, and the receiver can (must) lock to these pseudo-random processes. The preamble starts with the repeated transmission of the same symbol. This symbol is a perfectly balanced ternary sequence of 31 chips, spaced with silent chips (three in the case of Figure 4.5, but values of 15 and 63 are also possible [15,

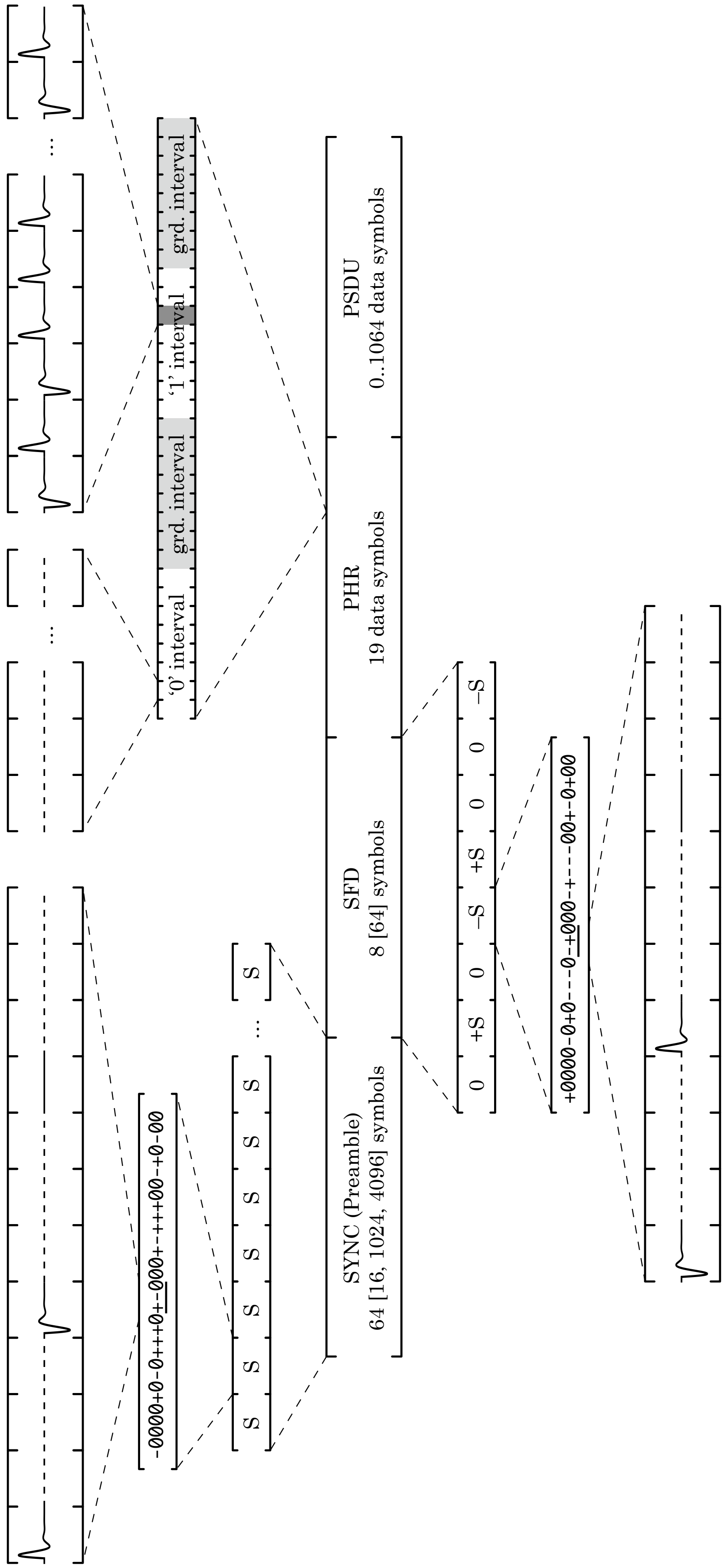


Figure 4.5: Impression of the bit-level IEEE 802.15.4a UWB packet. Values between square brackets are standard-compliant alternatives.

Tab. 39b]). Hence, the receiver should lock to the preamble both in *delay* (listen during the correct chip interval, 1 out of 4) and *code phase* (find the symbol start, 1 out of 31).

4.2.1 Ranging procedure

IEEE 802.15.4a specifies optional ranging support; devices with this support are called Ranging-capable Devices (RDEVs). Typical ranging is two-way [15, §5.5.7.1] and is outlined in Figure 4.6. One node sends out an Ranging Frame (RFRAME) and when the SFD left the antenna, it effectively starts its 64 GHz counter [44, p. 16]. A second node receives this frame, which also includes a request for acknowledgement. Upon receipt, the second node starts its counter, and stops it again when its acknowledgement leaves the antenna. Upon receipt of this acknowledgement at the first node, this nodes also stops its counter.

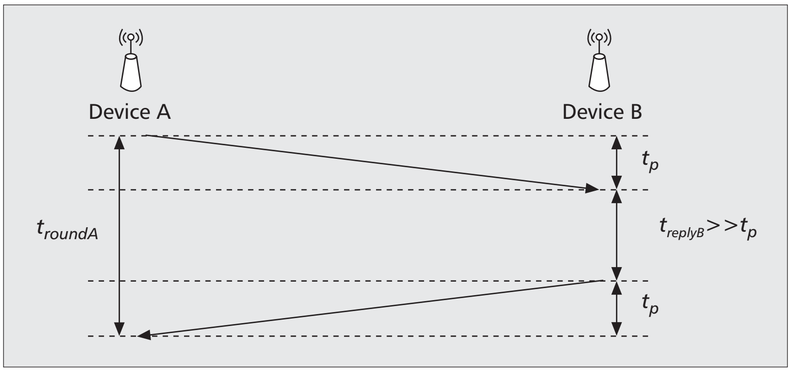


Figure 4.6: Outline of the two-way ranging procedure as specified in [15, §5.5.7.1] (from [13, Fig. 2]).

Both nodes send their counter values to some central node in a timestamp report [15, §6.8a.15], together with a Figure of Merit (FoM) that indicates the measurement uncertainty. This central node subtracts both time measurements and divides by two; this is the estimated Time of Flight (ToF). Multiplication with the speed of light gives an estimate of the range with known uncertainty.

The standard also mentions one-way ranging (possible when some nodes in the network have a shared notion of time) and Symmetric Double-Sided Two-Way Ranging (SDS-TWR) (to cancel out clock frequency differences between both nodes). Furthermore, an algorithm is suggested to obsolete the 64 GHz counter [15, §D1.2] and use slower sampling.

4.3 Current Software

A complete sender-channel-receiver chain was designed and modelled for simulation in MATLAB within IMEC. This section gives an overview of the simulation and indicates its most important modelling decisions.

4.3.1 Overview

The complete chain is modelled in `ieee802154a_MAIN.m`, and a schematic overview of this file is given in Figure 4.7. The transmitter encodes the bits of the incoming PSDU as ternary chips (one complex sample $\in \{-1+0j, 0+0j, 1+0j\}$ per chip). These chips are convolved with a channel model, that incorporates multipath propagation. To model the noise that is chiefly generated in the receiver electronics, a circular symmetric complex random value is added to each chip. Finally, these chips are fed to the receiver, that tries to recover the bits of the original PSDU. Note that the complex signals model time-invariant I and Q channels from transmitter to receiver, which is equivalent to a receiver oscillator that is perfectly synchronised to the transmitter oscillator.

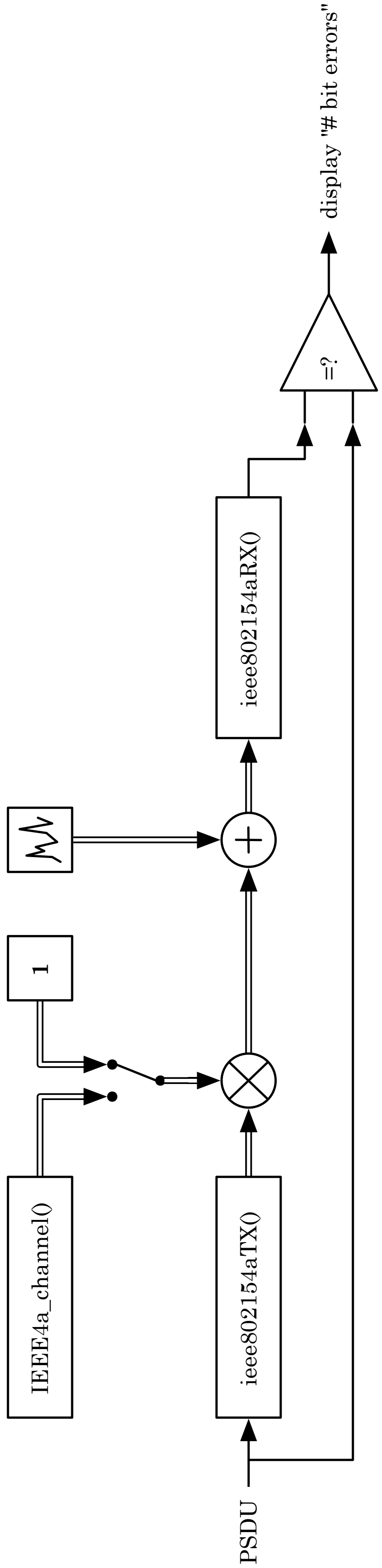


Figure 4.7: Schematic impression of `ieee802154a_MAIN.m`. Double lines represent complex-valued baseband signals.

The transmitter is depicted in Figure 4.8. Note that the SHRpreamble() function generates both the synchronisation preamble and the SFD.

The receiver is depicted in Figure 4.9. First, timingSynchronisationNonCoherentHistogramTracking() tries to synchronise to the incoming incoherent signal. That is, only the magnitude of the incoming complex signal is used. If synchronisation succeeds, a sample index (startPHR) results, which is used to decode the PHR and eventually the payload.

4.3.2 Digital Receiver

Because the Signal to Noise Ratio (SNR) of the received signals is generally low, an extensive procedure is needed to find and lock to the received signal (see Figure 4.10). First, we need to lock on the correct delay d (measured in chip intervals), to make sure that we are not listening during the spacing between the active chips. Next, we need to find the beginning of the code, or code phase φ (measured in trits). Then, we can look for the SFD. Let us look at the flowchart step-by-step.

For the first step, a number of samples that knowingly do not contain signal are analysed to determine the periodic correlation given the preamble code length. A correlation threshold is set that typically 1% (or more) of the noise samples will exceed, when correlated with the preamble code.

Given this threshold, we coarsely correlate four symbol lengths of samples with the known preamble code. Coarsely correlating means that we add up the sample magnitudes under a + or a – code trit and ignore the samples under a 0 code trit, for all delays d, for all code phase φ. In formula:

$$R_{\text{coarse}}[\varphi, d] = \sum_{n=1}^{N} \left|C[n]\right| \cdot \left|x'[(n+\varphi)L + d]\right|, \tag{4.1}$$

where φ is the code phase in trits, d is the delay in chip intervals, $C[n]$ is the nth trit of the ternary code, N is the code length, $x'[n]$ the nth complex received baseband sample (when sampled at the chip rate) and L is the delay (including spacing) between preamble chips. Note that x' contains $N \cdot L$ samples and is cyclicly indexable (i.e. $x'[n] = x[n \mod N \cdot L]$, where $x[n]$ are the measured complex baseband samples).

If the highest coarse correlation is above the threshold, we might have found the correct delay. Because we ignored the 0 trits of the preamble code, the code is not perfectly balanced anymore, so the code phase of this maximum correlation is disputable.

Recall that the threshold falsely accepts 1% of the 'symbols' (typically). Therefore we perform another coarse correlation on the next four symbol lengths. If the correlation on the same delay/code phase-pair again exceeds the threshold, we are confident to have found the correct delay, accepting a 0.01% probability of mistaking noise for signal.[2]

Finally, we perform a fine correlation on the next four symbols that also takes into account the samples that should be zero, in order to find out the correct code phase. With the same conventions as used in (4.1), this correlation amounts to

$$R_{\text{fine}}[\varphi, d] = \sum_{n=1}^{N} \left(\left|C[n]\right| - \frac{1}{2}\right) \cdot \left|x'[(n+\varphi)L + d]\right|. \tag{4.2}$$

In words: we multiply the samples under a $+1$ or -1 trit with $+1/2$, while multiplying samples under a 0 trit with $-1/2$. This way, although demodulated incoherently, the code is balanced again. Consequently, the code phase can be determined with higher accuracy. Furthermore, this cross-correlation can be used as a time discrete estimate of the equivalent baseband channel impulse response, denoted $|\hat{h}_b[n]|$ or Voltage Delay Profile (VDP).

[2] Because the maximum correlated delay/code phase-pair was found during coarse correlation, the code phase is disputable. Therefore, it would make sense to only confirm that the maximum correlation has the same delay.

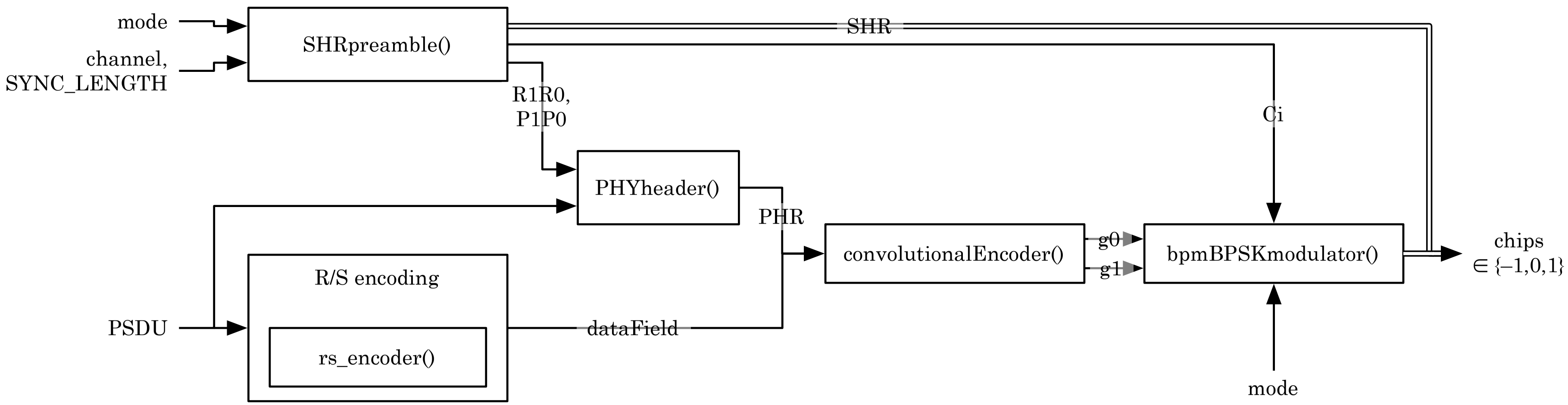


Figure 4.8: Schematic impression of `ieee802154aTX.m`. Double lines represent complex values. R1R0 signals the data rate and P1P0 signals the preamble duration, which are both fields of the PHR. g0 is the systematic or data bit (used for BPM) and g1 is the parity bit (used for BPSK). Ci is the preamble code, used to seed the pseudo-random generators in the modulator.

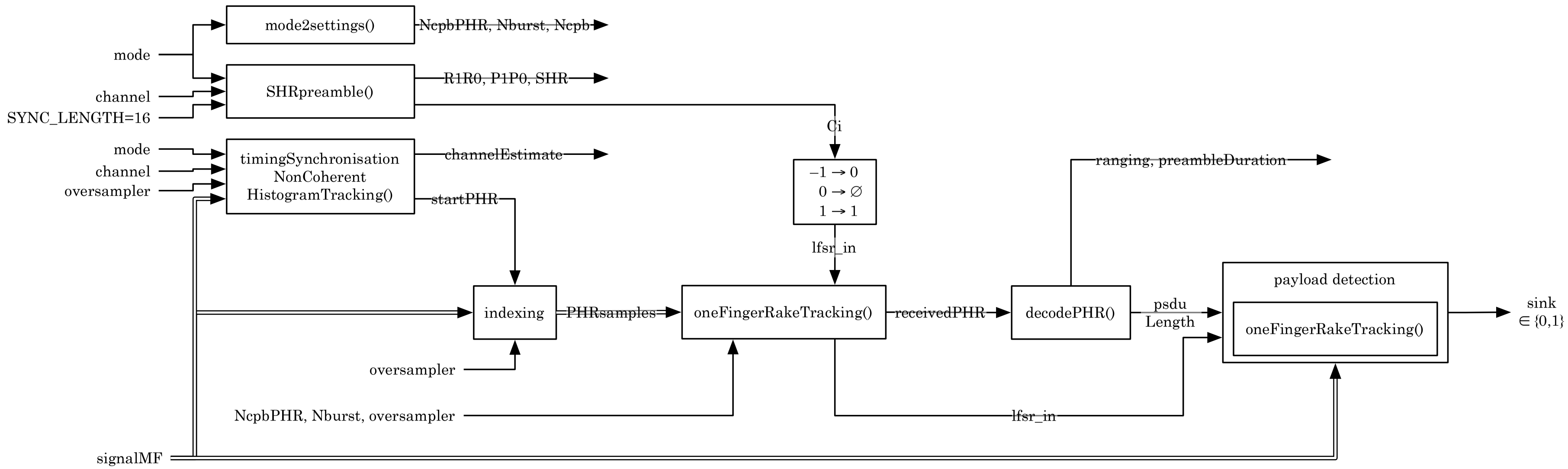


Figure 4.9: Schematic impression of `ieee802154aRX.m`. Double lines represent complex values. Ci is the preamble code and lfsr_in is the seed for the LFSR pseudo-random generators.

With the knowledge gained thus far, the probability distribution of the fine correlation of noise with the full code symbol is predicted. Using an estimate of the signal power, the probability distribution of the correlation of signal with the full code symbol is predicted. Using both distributions, a threshold to distinguish between symbol and noise is set. During the remaining symbol intervals, we non-coherently search for the SFD (01011001) with this threshold. During this search, we continuously perform fine correlation and track the maximum correlation delay and code phase, like a rake receiver. We average all the fine correlation results, after aligning the maximum correlation of each fine correlation. In doing so, we obtain an improved channel impulse response estimate or VDP $|\hat{h}_b[n]|$.

As soon as the SFD is found, the receiver is synchronised in delay and code phase, ready to decode the PHR and PSDU.

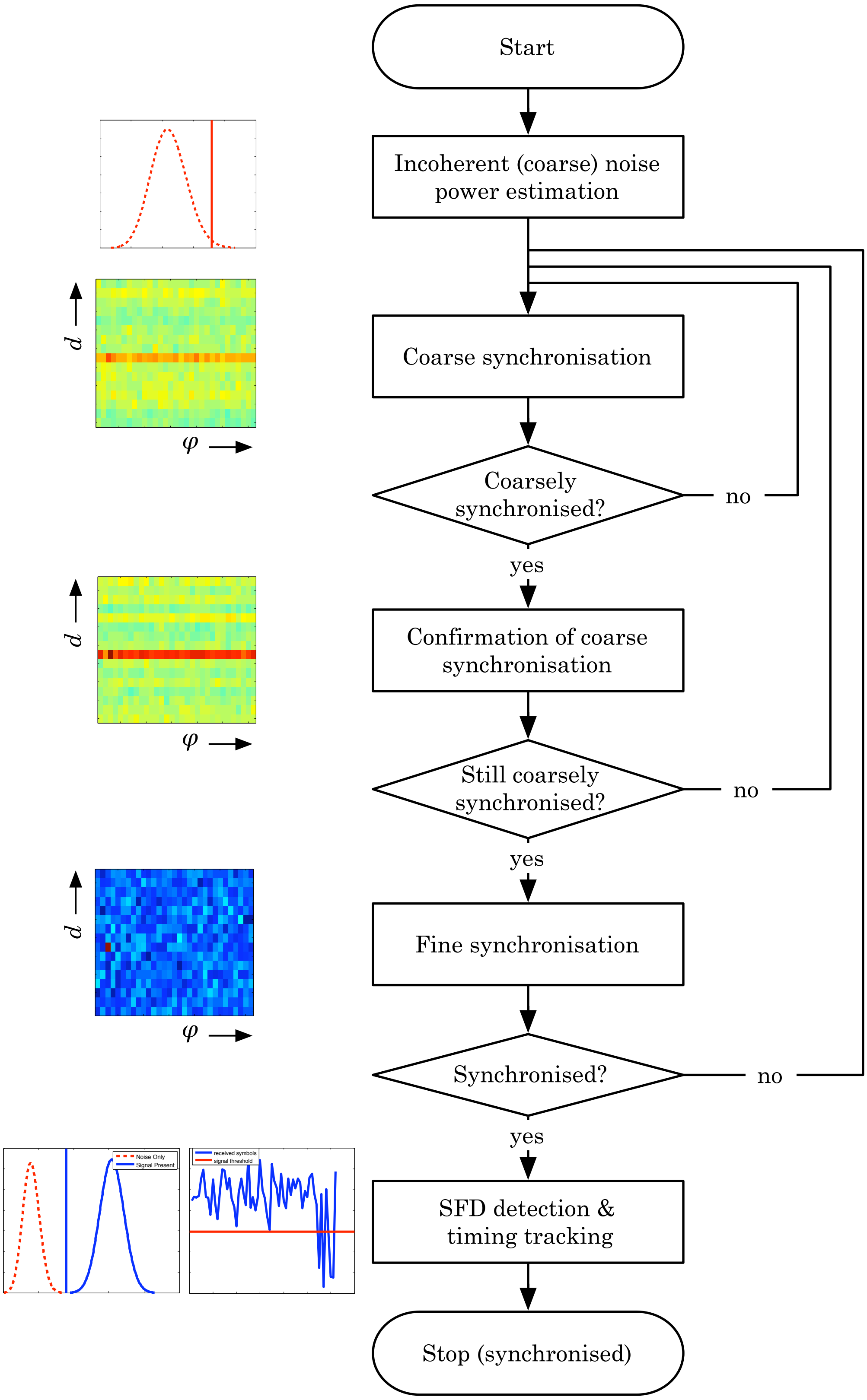


Figure 4.10: Flowchart of `timingSynchronisationNonCoherentHistogramTracking.m`. The failure (non-synchronised) final state is not drawn, this happens when the function runs out of samples during synchronisation or searching for the SFD.

4.4 Current Hardware

A stand-alone transmitter and the analogue part of the receiver were implemented on silicon by IMEC's Wireless Group [45; 46]. The chips are mounted on break-out PCBs, to which we will refer as 'modules'. The PCBs also contain biasing inputs and supply voltage de-coupling. The modules are combined in a demonstrator set-up as schematically depicted in Figure 4.11 and photographed in Figure 4.2.

The transmitter is programmed via Serial Peripheral Interface Bus (SPI) to send a IEEE 802.15.4a-compliant preamble, followed by a payload where g0 and g1 both are zero during 128 symbols. The digital controller receives the chip clock (499.2 MHz) from an external clock generator, because the on-chip clock divider that should derive from the Digitally Controlled Oscillator (DCO), does not yet work. The transmitter and receiver DCOs are programmed (via SPI) to oscillate at about 4.60 GHz. The RF signal from the transmitted module is amplified with a discrete amplifier, because the on-chip amplifier of the transmitter IC does not yet have enough gain. Elliptical pseudo-monopole antennas [47] are used to transmit and receive the UWB signal over the air. Finally, the receiver module demodulates the RF signal to baseband pulses.

An oscilloscope is used to capture both the pulse control signals (b0 (magnitude) and b1 (polarity)) and the baseband received signal (I and Q). These samples are fed to leadingEdgeDetection(), which is a variant of timingSynchronisationNonCoherentHistogramTracking(). In this variant, oversampling is properly supported. Furthermore, the VDP that is refined during fine synchronisation is used to find the peak and leading edge. The time difference between the peaks (or leading edges) of the transmitted signal (b0) and the received signal (I,Q) is used to estimate the range.

4.5 Practical Problems

While using the hardware demonstrator for ranging experiments, two practical problems were encountered. The USB-SPI interface slowed down the measurement cycle and the Variable Gain Amplifier (VGA) impaired the channel estimations.

4.5.1 Slow SPI-USB Interface

The ranging algorithm uses the received preamble, so we want the transmitter to send a preamble once for every range measurement. Unfortunately, after programming, the transmitter sends a preamble only once and then keeps looping the payload. The only way to get the transmitter to send another preamble is by resetting the transmitter and reprogramming. This programming took the best part of the ±40-second measurement cycle. In our measurement campaign, we will want to take many measurements, so this programming delay will severely constrain the number of measurements we can take. Therefore, we would like the programming to be significantly faster.

Furthermore, to initialise the measurements, we need to program both the transmitter and receiver module. Programming the receiver implies disconnecting the fragile USB to SPI interface cable from the transmitter module, running a somewhat unpractical GUI, manually entering some register values and reconnecting the interface to the transmitter module again. To run the demonstration on a different PC, not only the MATLAB files needed to be copied, but also this GUI, which does not promote portability. The manual entering of register values does not promote reproducibility. Therefore, we would like to obsolete the GUI and integrate SPI control in the MATLAB scripts. Also, we would like to support multiple SPI interfaces, so that the modules can stay connected.

The USB-SPI interface commonly available at IMEC consists of FTDI's FT232RQ USB to serial converter, together with a MSP430F149 microcontroller in a small housing (Figure 4.12). The

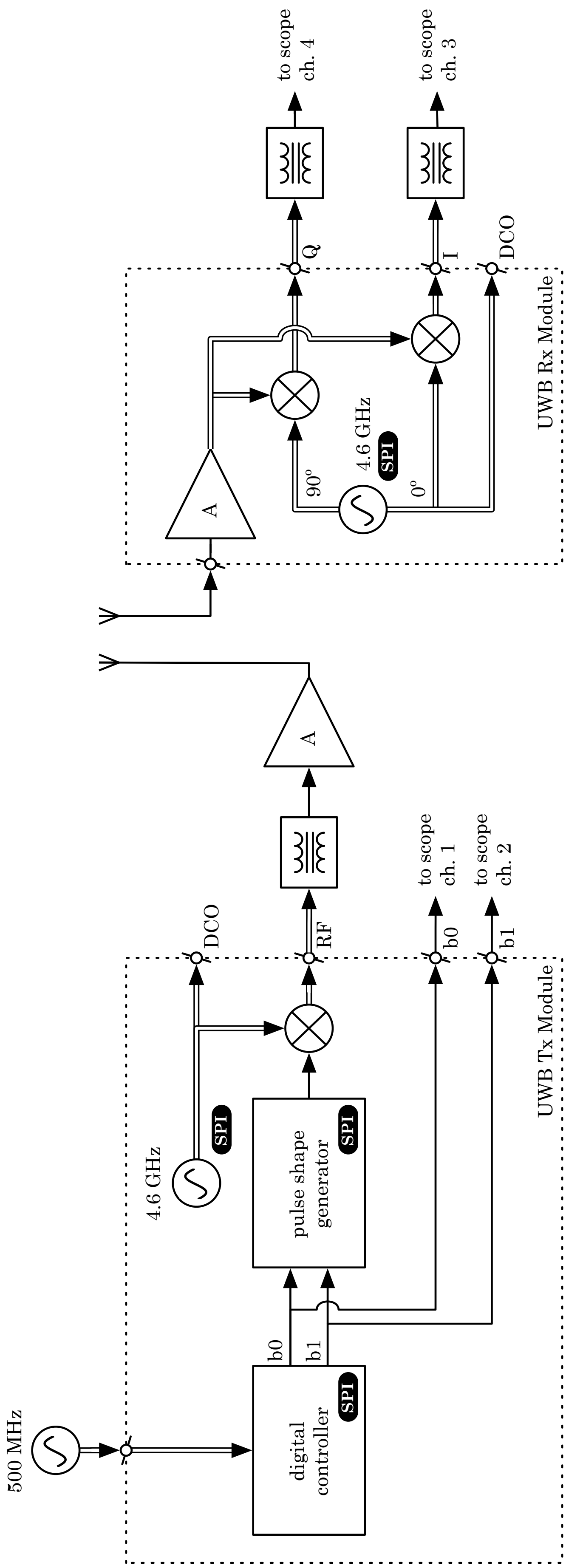


Figure 4.11: Schematic overview of the hardware of IMEC's UWB ranging set-up. Double lines indicate differential signals, which need to be converted using baluns if single-ended signals are required.

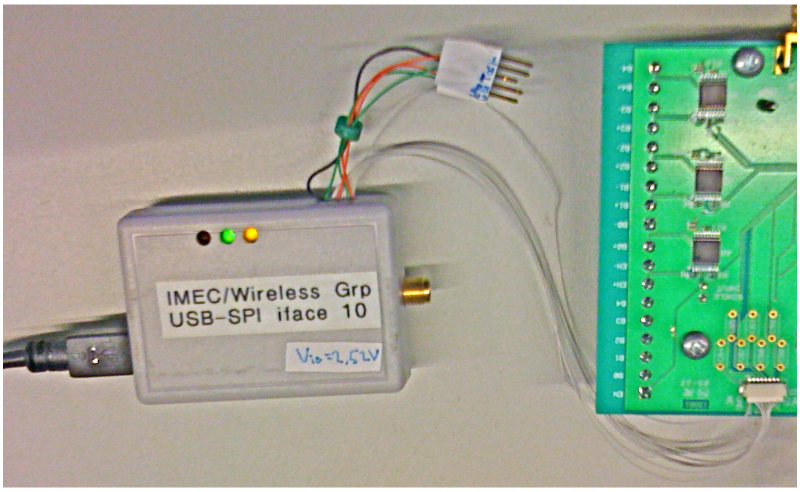


Figure 4.12: The USB-SPI interface commonly available and used at IMEC's Wireless Group. The white cable bundle contains the SPI signals and connects to the module. The coloured wires with the pin header are a break-out of serial and flow control signals.

microcontroller accepts serial signals from the USB to serial converter in a Universal Asyncronous Receiver/Transmitter (UART). This UART has a one-byte buffer. The original software prevented this buffer to overflow by writing bytes to the serial port one-by-one.

To speed up programming and to make it more convenient, the original software was ported from Visual C++ to MATLAB. The firmware of the microcontroller was adapted to have an 8-byte ring buffer with hardware flow control (Section C.1). The MATLAB driver first collects all the bytes it needs to write to the USB-SPI interface, and then writes it all at once to the serial port (see Section C.2). The hardware flow control, together with the 8-byte ring buffer takes care that the UART does not overflow. These enhancements achieved about a ×135 speed-up (see Figure 4.13), as well as a fully automated measurement set-up (multi-interface support obsoletes changing of cables and manually entering register values in the GUI).

Finally, the sample rate of the oscilloscope was reduced from 5 GS s^{-1} to 2.5 GS s^{-1} and only the necessary samples were retrieved. The measurement cycle now takes about 2 seconds, allowing to 'play around' with the demonstrator, which promotes experiments.

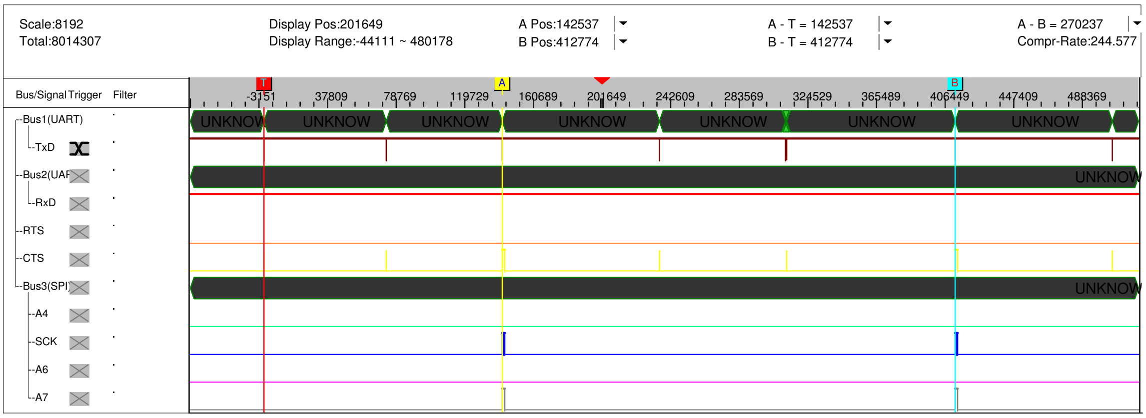
4.5.2 VGA Response

During experiments, the VDP always showed quite a bump after the main peak. At first we attributed this to multipath reflections, but it seemed so systematic that we decided to take away the air channel: we connected the transmitter to the receiver using a cable. The VDP is plotted in Figure 4.14.

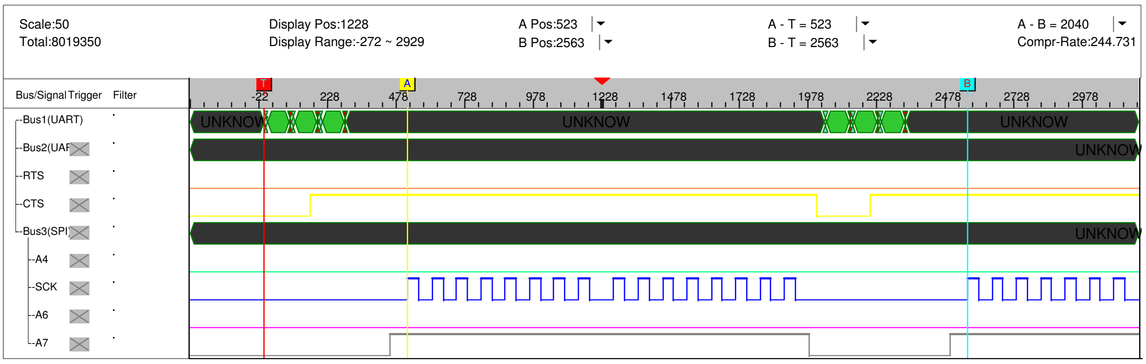
From this response, we can conclude that the bump is at least not only an effect of multipath reflections. To rule out errors in the algorithm, we analyse the received signals directly on the oscilloscope (Figure 4.15). Also there, we see that a sharp (2 ns) pulse results in a broader pulse (±10 ns) with a trailing bump. We conclude that the problem is not (only) in the algorithm.

To localise the source of this distortion, we connect the transmitter's RF output to a high sampling rate oscilloscope (Figure 4.16). We recognise an exponential envelope, which is intentional according to the IC designers. Note that we do not see a trailing bump here, and the envelope is about as wide as the b0 control signal. Concluding, we hypothesise that the main source of distortion must be in the receiver.

Instability of the receiver's VGA was a known problem, depending on the VGA's gain and bandwidth settings. Together with a designer of the VGA, we tuned the bandwidth and gain setting



(a) Original software: every RS232 byte is individually written to the port. After three RS232 bytes, one SPI register is written. The A–B interval is 27 ms.



(b) New software: many bytes are written to the port at once. Hardware flowcontrol prevents the UART overflowing (notice the CTS signal inhibiting the flow of incoming bytes). The A–B interval is 0.2 ms.

Figure 4.13: Logic analyser plots of the RS232 and SPI communication. The A and B markers are placed at the beginning of subsequent SCK bursts; the A–B interval is the time between writing two SPI registers; al time is measured in 0.1 μs units.

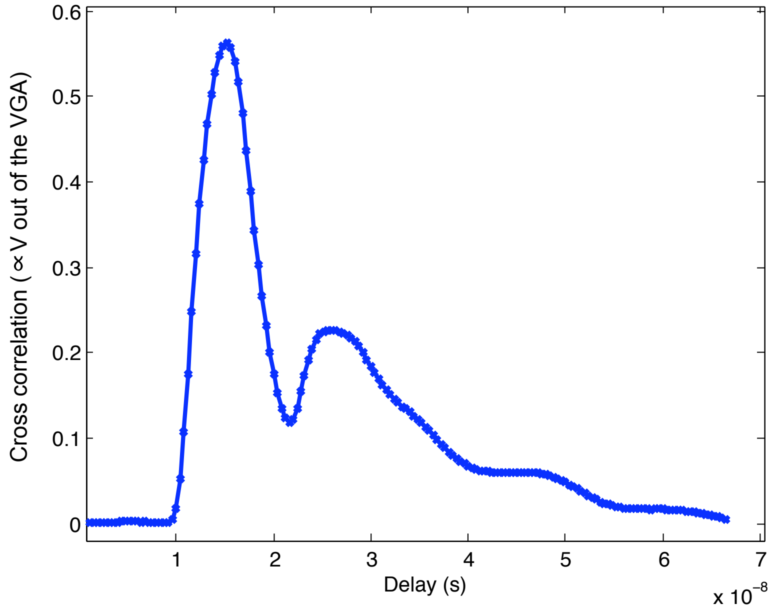


Figure 4.14: Estimated channel impulse response, when the transmitter RF output is connected with a 1 m cable, 6 dB attenuator and a DC blocker to the transmitter input (no Power Amplifier).

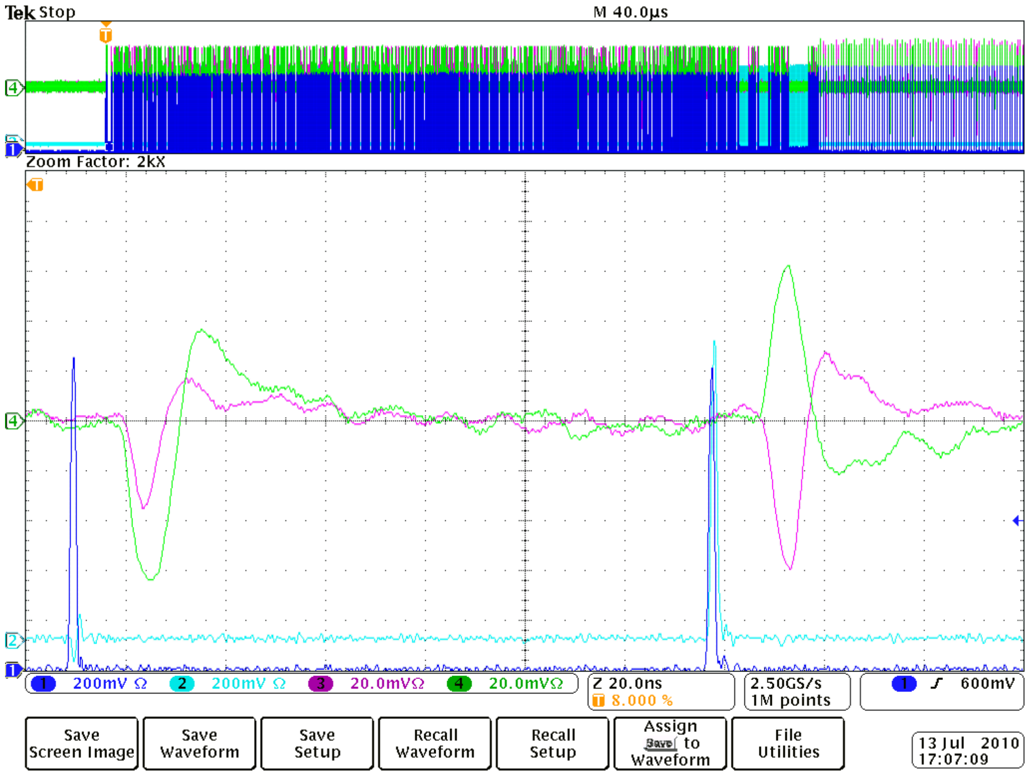


Figure 4.15: Oscilloscope photograph of the baseband sent and received signals. Blue and cyan are the transmitter's b0 and b1 signals, respectively. Purple and green are the receiver's I and Q signals, respectively. In the top window, all preamble is visible, while in the bottom window, two pulses are cut out (a +1 and a −1 chip, respectively).

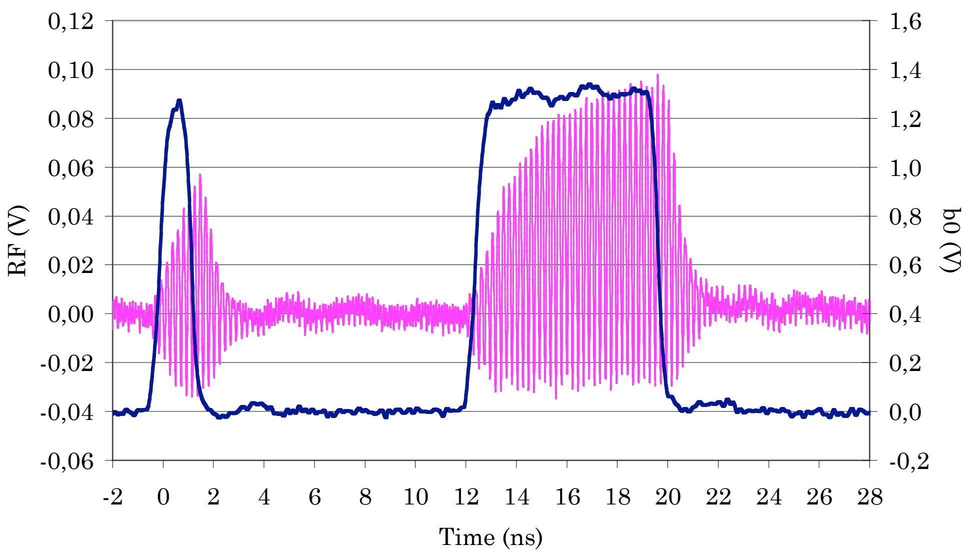
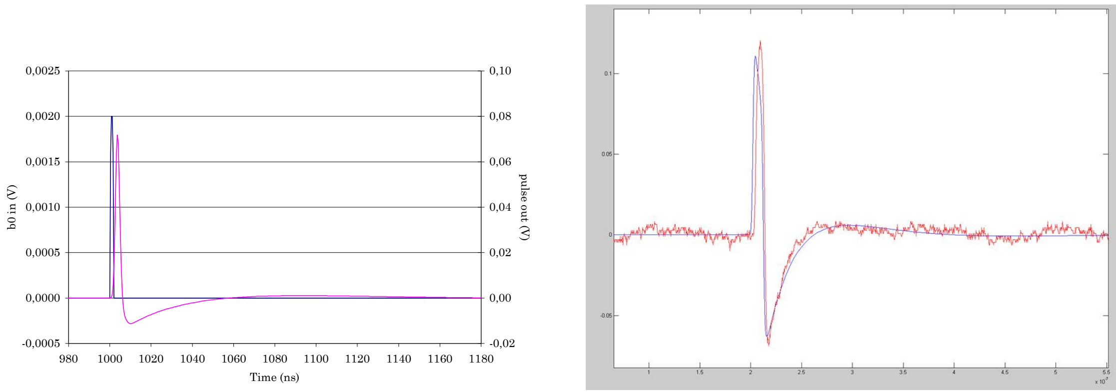


Figure 4.16: Transmitter RF output for one and for four positive pulses. The single pulse is the first from the preamble, the burst of four positive pulses is taken from the payload (there is a hiatus in the time scale). Notice that the transmitter's modulator is not 'watertight'; even when b0 is low, a carrier is clearly visible at the RF output. This causes an undesirable spectral peak at the carrier frequency.

to cause visible ringing. This ringing turned out to be a shifted Heaviside-function of the bandwidth setting (either the ringing occured, or it did not). With all the previous measurements, the settings were at a safe margin from this threshold. Therefore, we conclude that the trailing bump is not caused by the VGA's instability.

After the VGA, there is a 6th-order low pass filter ($f_c \sim 150\,\text{MHz}$) and before the VGA, there is a DC-block, which are both designed externally to IMEC and of which no detailed specifications were available to us. Cadence simulations by an electronics engineer on a similar design in different technology showed that such a filter could cause a distortion like this (Figure 4.17). If this would be the main cause of the distortion, it is a linear effect, for which we could (partly) compensate by an inverse filter.



(a) Cadence simulation result of a 2 ns baseband pulse on the TSMC receiver design, both input and output signals.

(b) Cadence simulation output of a 8 ns baseband pulse on the TSMC receiver design, with the measured response from the UMC receiver scaled to fit.

Figure 4.17: Simulation and measurement results on the receiver baseband distortion. Simulations courtesy Pieter Harpe.

Therefore, we tried to estimate the filter's response, by summing multiple channel impulse responses. Recall that the receiver is incoherent, that is: the phase of its oscillator has no fixed relation to the transmitter's oscillator. As a result, the baseband response rotates freely in the complex plane. Taking the magnitude of the complex response is a non-linear operation; in fact, the negative trail would be flipped to be positive. That clearly would not a good estimate of the receiver's behaviour, and unusable to compensate the distorted output signal. Alternatively to taking the magnitude, we tried to compensate for the DCO frequency offset between the transmitter and receiver. While trying, the frequency difference between transmitter and receiver turned out to be so unstable, that compensation was impossible. We did not investigate other ways of estimating the response of the VGA because of time constraints.

4.5.3 Coping Strategy

We have noticed, analysed and solved the slow measurement cycle: taking measurements formerly took 40 s and now take 2 s at half the sample rate.

We also noticed and analysed unwanted distortion caused by the receiver's VGA. Due to lack of time, we were not able to partly compensate for this problem. Therefore, we will keep in mind that the VDPs will be distorted and still investigate the feasibility of ODP detection and mitigation with this impaired signal.

5 Measurement Campaign

This chapter support the design of the measurement campaign, which is a measurement *plan* carried out using a measurement *set-up*.

We start by stating the objective of the measurement campaign. Then we identify all parameters that might affect measurement results and select a finite number of interesting values on each parameter. Consequently, we present a measurement plan that explores the parameter space as spanned by the aforementioned parameter values. Finally, we describe the measurement set-up that will help to efficiently take measurements.

5.1 Objective

We plan to take measurements to test the different localisation algorithms against. The algorithms should (a) be able to discern Dominant Direct Path (DDP) from Obstructed Direct Path (ODP) conditions, and using this knowledge (b) better localise the nodes.

The measurements should be representative of an applied indoor 'smart dust' localisation product. Representative means "typical of a class, group (...)", so sampling is implied [48]. In the context of this thesis, we would like to add non-uniform sampling to that definition. For instance, let us agree that a good newspaper is representative of the world's events. If the sampling of a certain newspaper were really uniform in time and space, this newspaper would be particularly boring: it would contain stories about John Doe going to his work, after walking his 1.3 dog and saying good-bye to his 2.3 kids. Therefore, events are sampled according to their relevance. Therefore, we define 'representative' as follows:

A representative set of a whole is a strict subset with finite members of the whole, where the elements of the representative set have the same relevance.

In this research, samples that put the algorithms to the test are more relevant than the 'easy' samples. Still, some 'easy' samples are needed to make sure that the algorithms work. Furthermore, we do not know in advance what samples are 'easy' and what samples are 'difficult' for the algorithms under test.

5.2 Parameters

Let $P_1, P_2, \ldots, P_n$ be all parameters that affect measurement results (for example, antenna and distance). Let us try to describe these parameters by independent scalars $p_1, p_2, \ldots, p_m$ (for example, azimuthal variance, directional gain and distance). All relevant measurement parameters can then be defined by $\vec{p} = [p_1, p_2, \ldots, p_m]^T$, that can be though of as point in the m-dimensional parameter space. Let us now try to identify all relevant parameters.

To test the ODP detection algorithms, we need to take range measurements. In other words, we send a frame from transmitter to receiver by air and use the power delay profile to detect an ODP condition. Roughly walking the chain, we identify these parameters: receiver and transmitter, protocol, antennas, distance, obstacles and surroundings.

To test the localisation algorithms, we need to take multiple range measurements for each localisation. The localisation difficulty will depend on the localisation principle used, the geometry of the nodes with respect to each other and the number of anchors.

There are infinite possible measurements $\vec{p}$ in the parameter space spanned by the abovementioned parameters, but there is only finite time to perform these measurements. Therefore, for each of the mentioned parameters, we will try to conclude what value or range of values is typical and interesting (that is, *representative*). This finite number of parameter values for a finite number of parameters can be thought of as spanning a hyperrectangle containing finite points. The measurement plan determines on what region of this hyperrectangle will be measured.

Let us now consider each parameter one by one, trying to select representative values. In order to do so, we sometimes need to perform preliminary measurements. Some parameters will just be fixed to one value, while for other parameters both extreme values are identified and sampled at zero or more values in between.

5.2.1 Receiver and Transmitter

The algorithms will work on a channel impulse response estimate or Voltage Delay Profile (VDP). We could measure this with dedicated channel sounding equipment, such as a pulse generator, carrier generator, mixer, and oscilloscope. However, the quality of the channel estimate thus obtained will not be typical of an embedded Ultra Low Power (ULP) radio; it will be too good. To get a more realistic channel estimate, we would need to model the imperfections typically present in an embedded ULP radio, such as, but not limited to, base-band frequency response, LO jitter, noise, and carrier leakage. We would then need to validate this model against a real radio and prove that we produced a competent model.

Alternatively, we could use the receiver and transmitter modules available at IMEC. The advantage is that we are sure that the resulting localisation performance will be realistically achievable. The drawback is that we cannot easily investigate the influence of the radio imperfections on localisation performance. In addition, we need to prove or make probable that IMEC's radio is representative of smart dust radios in general.

We choose to use just IMEC's modules, because they will result in realistic performance numbers. Meanwhile, we accept that we cannot prove the general validity of the results nor can we investigate the influence of the various radio imperfections on the performance. Still, we can compare algorithms amongst each other and draw qualitative conclusions.

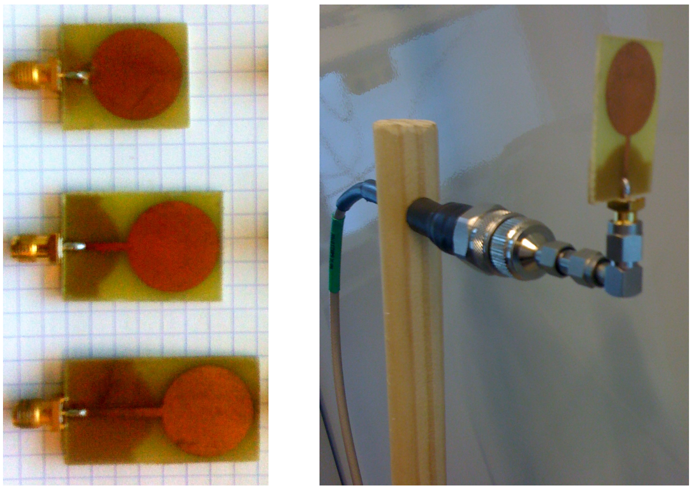
5.2.2 Protocol

Given the choice of transmitter and receiver above, the protocol is fixed to be IEEE 802.15.4a, which is outlined in Section 4.2. Within this protocol, three frequency bands can be used [15, § 6.8a]: the sub-gigahertz band (250-750 MHz), the low band (3.1-4.8 GHz) and the high band (6.0-10.6 GHz). The transmitter module supplied by IMEC operated in the low band at mandatory implemented channel 3 (centre frequency 4.5 GHz, bandwidth 500 MHz).

Within this channel, different modes can be selected, differing in data rate. Bearing in mind smart dust sensor networks, we use mode 2.2 with a relatively low datarate of 850 kbps [49, Table 1]. We use short preamble of 64 symbols, that should work in moderate channel conditions such as encountered in indoor smart home applications.

5.2.3 Antennas

In a 'smart dust' application, a small antenna will be connected directly to the electronics. Typically, the nodes can have any orientation, but 'up' and 'down' are fixed, in our case of tile-shaped nodes. Therefore, the antenna will be the result of a trade-off between cost, size and azimuthal omnidirectivity. Given the peripherals required to use IMEC's transceivers, it is not yet practical to directly attach the antenna to the electronics and move the combination around. Therefore we will propose a set-up in which the antennas are connected by cables to the electronics. Then, we will first verify that this set-up is representative in the sense that the antenna is the main radiating element and not the feed (which is not present in real smart dust nodes). Next, we will verify that the antenna pattern is maximum at the horizon and constant along the horizon. We conclude by a decision on antenna, mount and feed.



(a) With 0 mm, 7 mm and 14 mm of printed ground plane.

(b) Mounted with a 90° SMA coupler.

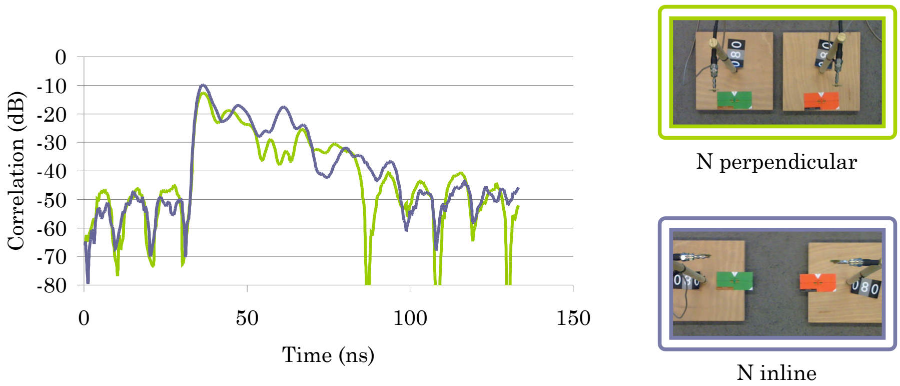
Figure 5.1: The PCB microstrip elliptical pseudo-monopole antenna.

One suitable antenna is the microstrip elliptical pseudo-monopole [50]. It is a relatively small (2 × 2.5 cm) and cheap (PCB) antenna (Figure 5.1a). We would like to fix the antennas in space, without the stands significantly changing the antenna pattern or reflections. Therefore, the antennas are mounted on wooden stands; the foot of the stand is a tile of Okoumé plywood, the 23 mm diameter pole is made of fir[1]. The cable is fed perpendicularly through the pole, with the antenna rotated to be upright (Figure 5.1b), because the reported plane of omnidirectivity is then coincident with the horizontal plane [50]. A 90° SMA coupler is used in an attempt to decrease cable currents (i.e. to avoid the cable being a significant radiator).

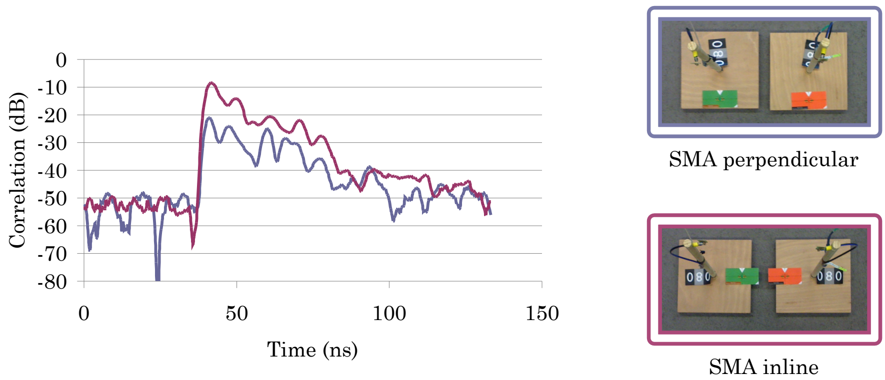
We would like to verify that cable feed plays no significant role in this set-up. In other words, we would like the main radiating element to be the antenna, because without knowledge of the smart dust node, the antenna is the best representation of a node with integrated antenna. Put differently, the cables are effectively invisible in the measurement set-up and can be fed to the antennas in any practical way. To get an indication of the influence of the feed, we keep the antenna separation and orientation the same (30 cm and facing each other), while changing the feed direction. We measure the power delay profile for an N-SMA adapter close to the antenna (Figure 5.2a) and for an SMA cable, with an N-SMA adapter further down the cable (Figure 5.2b). We conclude that with an N-SMA adapter close to the antenna, the feed direction has less influence. Still, we observe 3 dB variation in the Direct Path (DP). To minimise the influence of the feed, we will use the N-SMA adapter close to the antenna. To keep the residual influence constant, the feed will always be perpendicular to the antenna.

We now try to verify that the plane of maximum gain coincides with the horizontal plane. That is, at 0° elevation. We know that a ground plane is necessary for a monopole antenna, but a ground plane that is long with respect to the wavelength also causes elevational lobes in the directional pattern, thereby decreasing the gain at 0° elevation [50, § 2.2]. To find a representative length of the ground plane, we measure the DP power, while changing the elevation of one antenna. This way, we get a rough elevation pattern for 0 mm, 7 mm and 14 mm of printed ground plane (Figure 5.1a). The result is plotted in Figure 5.3. We notice that a 0 mm ground plane has an asymmetrical elevational pattern. The 14 mm variant starts forming lobes, pre-

[1] A 2 inch slab of fir is reported to attenuate 4-5 GHz by 5.0-5.8 dB [51]. We assume that this pole will attenuate by no more than 2.4 dB.



(a) With an N-SMA adapter close to the antenna.



(b) With an SMA cable, and an N-SMA adapter further down the cable.

Figure 5.2: Voltage Delay Profiles (VDPs) of two antennas at 30 cm distance, facing each other, for different feed directions.

sumably because even the 90° SMA coupler adds to the ground plane length. In a real application, the ground plane will be designed such, that no lobes appear. Therefore, we deem the set-up with a 90° SMA coupler and a 7 mm printed ground plane to be the most representative of an on-PCB antenna that is directly connected to the electronics.

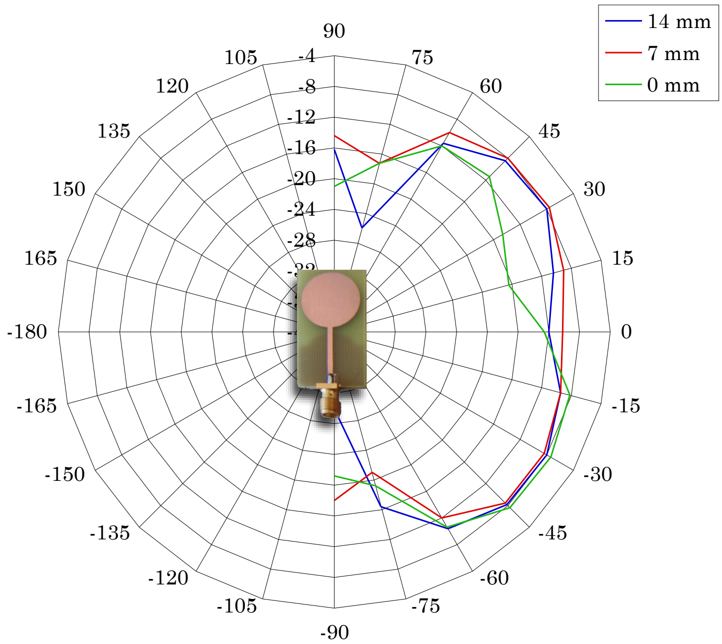
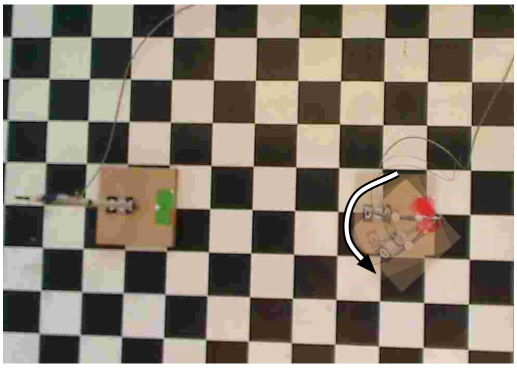


Figure 5.3: Correlation in the Direct Path for different receiver antenna elevation angles, while keeping the transmitter at 45° elevation and 80 cm from the receiver. We repeated the experiment for different receiver ground plane lengths, while keeping the transmitter at 7 mm ground plane length.

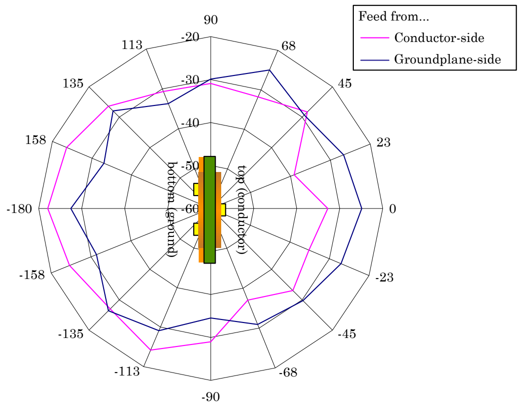
Next, we will validate that the antenna really is omnidirectional along the horizon. For that purpose, we put both antennas at 1 m separation, 80 cm above the ground. We keep the receiver antenna fixed, while rotating the transmitter antenna in steps of 22.5°. We perform this azimuthal scan with the transmitter feed coming from the side of the ground plane and once coming from the side of the conductor. The receiver feed always comes from the its conductor side (Figure 5.4a). We plot the strength of the DP in the Power Delay Profile (PDP) against the azimuth to obtain Figure 5.4b. We notice that when the feed comes from the side of the ground plane, the fluctuations in antenna gain are smaller. To verify that indeed the antenna azimuth does not significantly influence the measurements, we perform another sweep: we rotate the receiver antenna clockwise and the transmitter antenna counterclockwise (Figure 5.5a). The DP power and the estimated range are plotted in Figure 5.5b. Indeed, at almost all differential azimuth, about the same range estimate is given from a VDP like Figure 5.5c. Only when the receiver azimuth is −90° and the transmitter azimuth is +90°, the DP drops dramatically and the first peak in Figure 5.5d goes undetected. This strange result is less severe, but still observable when the antennas are fed from the side of the conductor.

We do not understand this phenomenon, and deem further research necessary but also out of the scope of this thesis. In the sequel, we can avoid the phenomenon by always pointing the ground planes of both antennas at each other (−180° in Figure 5.5d). Because we observed the phenomenon a little less severe with the feed coming from the conductor side, we will use that feed direction.

Concluding, we will use microstrip circular pseudo-monopole antennas, directly connected to a 90°-SMA coupler; elevation will be both 0°, polarisation will be vertical. The feed direction has influence, but is minimal when the feed comes from the conductor side of the antenna. The

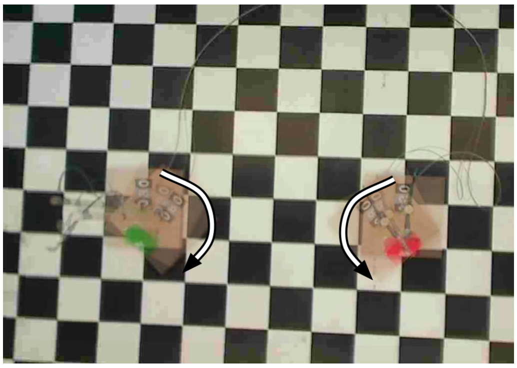


(a) The experiment set-up; the receiver antenna (left) is kept fixed, while the transmitter antenna (right) is rotated.

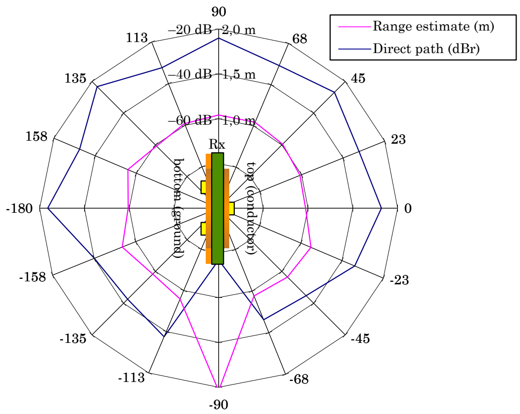


(b) The resulting antenna pattern estimate. The relative power in the DP is plotted against the azimuth.

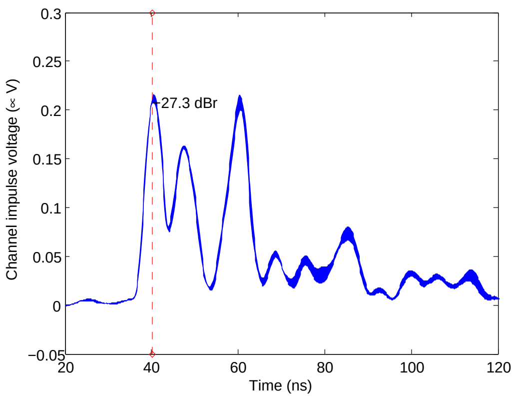
Figure 5.4: Antenna pattern estimate experiment; only the transmitter azimuth is varied.



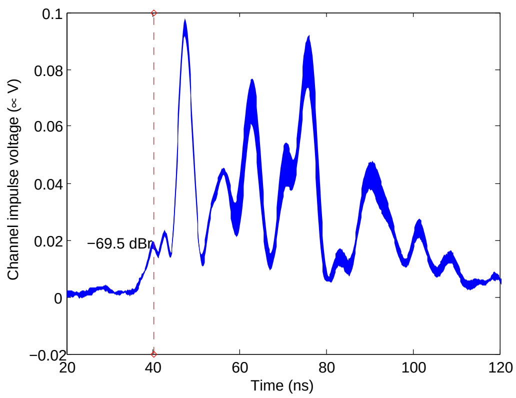
(a) The experiment set-up; the receiver antenna (left) is rotated clockwise, while the transmitter antenna (right) is rotated counterclockwise. Both antennas are fed from the side of the ground plane.



(b) The resulting antenna pattern estimate. The relative power in the DP and the leading edge range estimate are plotted against the azimuth of the receiver.



(c) Voltage Delay Profile (VDP) for 0° receiver azimuth. The dashed red line indicates the expected time of the direct path, knowing that the antenna separation is 1 m. The envelope indicates the variation through four subsequent takes.



(d) Voltage Delay Profile (VDP) for −90° receiver azimuth. The dashed red line indicates the expected time of the direct path, knowing that the antenna separation is 1 m. The envelope indicates the variation through four subsequent takes.

Figure 5.5: Experiment to verify independence of the azimuth; both transmitter and receiver azimuths are varied.

elevational pattern is maximum at the horizon and shows no lobes that are not representative for on-PCB antennas, when a 7 mm ground plane is used. The azimuthal pattern is not very omnidirectional and contains a deep null. We deem this null not to be representative, so if there is time to point the antennas at each other, the ground planes should face each other. Otherwise, the ground plane should always face the same direction (towards the corridor).

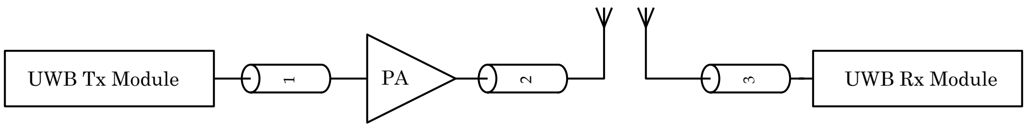
5.2.4 Distance

We decide to determine the size of the set-up by the maximum range of the receiver and transmitter chips with the current synchronisation algorithm. In a realistic application, both modules are connected directly to their antennas, with an air link in between (Figure 5.6a).

Currently, the transmitter's Power Amplifier (PA) does not work as it should, so we need an external amplifier. Also, the Low Noise Amplifier (LNA) of the receiver introduces much more noise than it should, effectively decreasing sensitivity. Because of the bulky set-up, we use cables to be able to move the antennas around. This current set-up is schematically depicted in Figure 5.6b.

We assume that the gain of the internal PA will be such, that the output power at the transmitter antenna is exactly legal. According to Appendix B, that gain is 13.6 dB with respect to the current output level. As for the LNA, the current Noise Figure (NF) is about 27 dB, where a NF of 5 dB is believed to be achievable. Assuming that the gain is improved accordingly, an effective improvement of 22 dB receiver sensitivity is realistic. Summed up, the future PA and LNA will add 35.6 dB to the current link budget.

(a) Ideal set-up.



(b) Current set-up.

Figure 5.6: Comparison of the gain and attenuation in the ideal and in the current set-up.

Experimentally, we determine that with 28 dB of attenuation between the current transmitter and receiver module (no external PA), the receiver is on the brink of losing synchronisation (50% success). 22 dB of attenuation yields 100% synchronisation. In the future situation, that will be $22 + 35.6 = 57.6$ dB of allowed attenuation.

We assume that both antennas are upright, but not facing each other. The used circular pseudo-monopole microstrip antennas have a reported azimuth gain between 3.02 dBi and 3.63 dBi for 4 GHz to 5 GHz with a maximum variation of between 1.69 dB and 2.54 dB [50, Table 2]. If we use the averages (3.33 dBi gain and 2.11 dB maximum variation), the minimum antenna gain on the horizon is 1.22 dBi. Including this worst case gain twice, the maximum allowed path loss is $57.6 + 1.22 \approx 59$ dB. Let us calculate the maximum antenna separation using Friis transmission equation, assuming a somewhat optimistic path loss exponent n of 2:

$$PL_{\max} = 10n \log_{10}\left(\frac{4\pi d_{\max}}{\lambda}\right) = 10n \log_{10}\left(\frac{4\pi d_{\max} f}{c}\right) \tag{5.1}$$

$$d_{\max} = \frac{c}{4\pi f} 10^{\left(\frac{PL_{\max}}{10n}\right)} = \frac{3\times 10^8}{4\pi 4.6\times 10^9} 10^{\left(\frac{59}{10\cdot 2}\right)} = 5.1\,\mathrm{m} \tag{5.2}$$

Concluding, the ideal system has a range of 4.6 m in the 2.2 mode, with the current algorithm and with the current antenna.

The available external PA, the Agilent 83017A, has a gain of 39 dB around 5 GHz [52, p. 8]. That is 3 dB more than the future improvement of LNA and PA (36 dB). This means that we should lose no more than 3 dB in cables and adapters to achieve the expected future performance. When using two WLU18 3 m cables, we lose 3 dB, excluding adapters (see Section B.5).

Because of the rigid cables, the antenna range is effectively 2.5 m around the module. With 0.5 m spacing between transmitter and receiver module, the inner rectangle of both ranges is 2.5 × 2 m (Figure 5.7). The antennas will move within this rectangle.

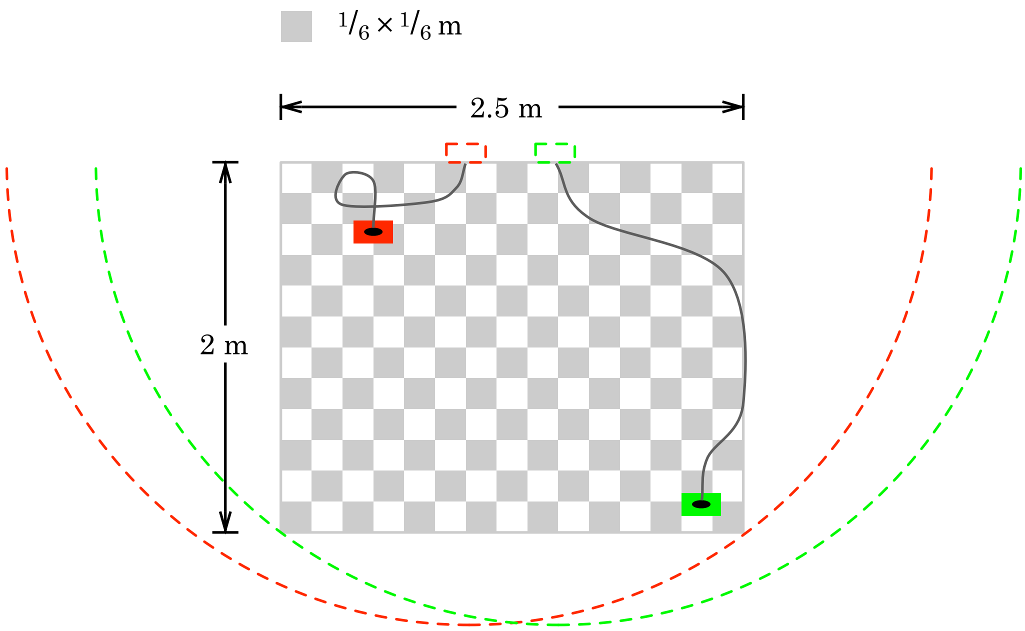


Figure 5.7: Top view of the set-up. The effective 2.5 m cable lengths from the transmitter and receiver module dictate the rectangular range of 2.5 × 2 m

5.2.5 Obstacles

In a real application, objects such as walls, furniture, ceilings and windows may obstruct the direct path by reflecting and absorbing the incident power. We would like to use mobile obstacles in the measurement set-up to represent these objects. (This kind of emulation is also applied in, for example, [24]). We will now synthesise what obstacles are representative; what material and what dimensions.

In [51], the reflection and transmission of 2.3 GHz and 5.25 GHz signals by 30 common building materials were assessed. The results for 5.25 GHz are summarised in Figure 5.8. Every marker is one of the tested materials, and some extreme cases are labeled. The pink curves are calculated iso-absorption lines, knowing that all incident power is either reflected, transmitted or absorbed.

We see that materials on the lower right actually do not alter the propagation significantly; almost all power is transmitted and none is absorbed or reflected. Note that by approximation, making a material thicker or thinner moves the marker horizontally (changes transmission), because reflection occurs in a relatively thin layer. We could look for plywood as a boundary case of some absorption and little reflection. More absorption and some reflection is given by cinder blocks (intermediate case). Finally, we could use a metal fence or sheet, which should give almost exclusively reflection.

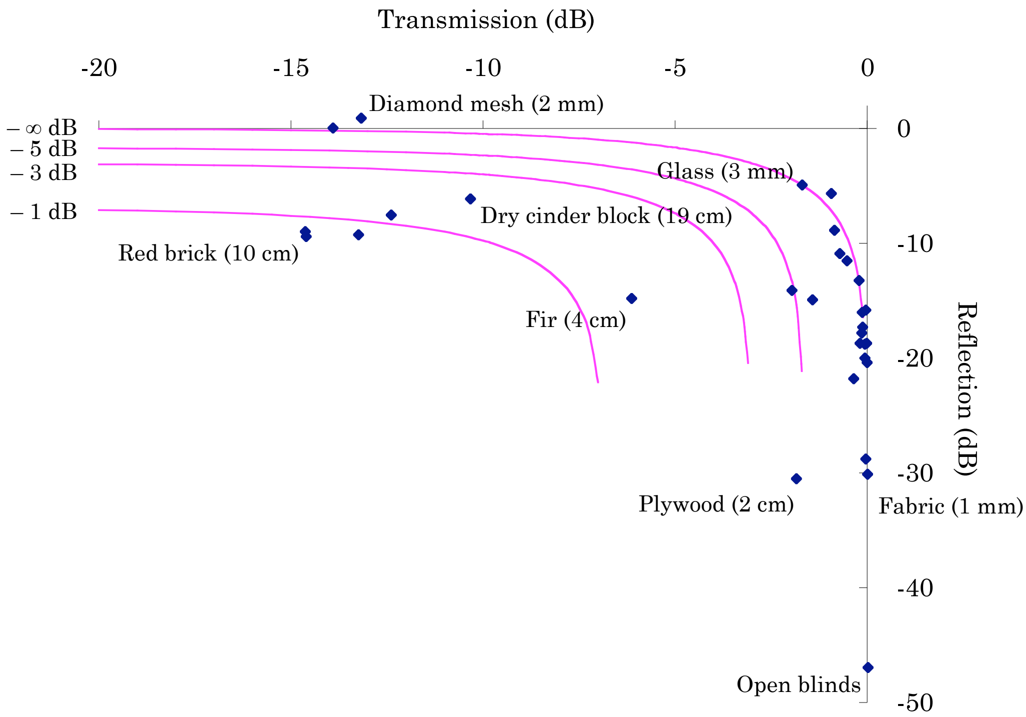


Figure 5.8: Summary of the transmission and reflection measurements on 30 common building materials from [51]. Pink iso-absorption curves are calculated and added, assuming that all incident power is either reflected, transmitted or absorbed.

Now how wide should these obstacles be to really shadow the receiver? If we apply geometrical optics, we predict that if all of the effective aperture of the receiver is obstructed, no signal will arrive at the receiver (apart from the power passing through the obstruction itself). We know that the effective aperture A_{eff} is given by:

$$A_{\text{eff}} = G\frac{\lambda^2}{4\pi}, \tag{5.3}$$

where G is the directional antenna gain and λ is the wavelength. If we imagine the aperture to be circular, we can calculate its radius in the optimal direction:

$$A_{\text{eff}} = \pi r_{\text{eff}}^2 \tag{5.4}$$

$$r_{\text{eff}} = \sqrt{\frac{A_{\text{eff}}}{\pi}} = \sqrt{G\frac{\lambda^2}{4\pi^2}} = \sqrt{10^{\frac{3.33\,\text{dBi}}{10}}\frac{(3\times10^8/4.6\times10^9)^2}{4\pi^2}} = 6.0\,\text{cm}. \tag{5.5}$$

So, to first approximation, we say that the signal is obstructed when the 12 cm-diameter cylinder between transmitter and receiver is fully obstructed.

In reality, the signal will diffract around the corners of the obstruction and still arrive at the receiver. If we model an obstruction as an opaque half plane, we get so called *knife-edge diffraction*. According to Huygens' principle, the absence of obstruction is a radiating half plane itself, so waves will propagate around the edge of the obstacle (Figure 5.9). To assess the impact of diffraction on the channel, Fresnel zones were invented [53]. Waves diffracted from within the first Fresnel zone have a 0°-90° phase shift with respect to the direct path. Waves diffracted from the second Fresnel zone have a phase shift of 90°-270° and as for the third Fresnel zone, waves have a phase shift of 270°-450° and so on. Fresnel zones are ellipsoids of which the major axis is the direct path between transmitter and receiver. Along this axis, the outer radius F_n of the nth Fresnel zone is:

$$F_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}}, \tag{5.6}$$

where d_1 is the distance to the transmitter, d_2 is the distance to the receiver and λ is the wavelength. A top view of the first three Fresnel zones is given in Figure 5.10.

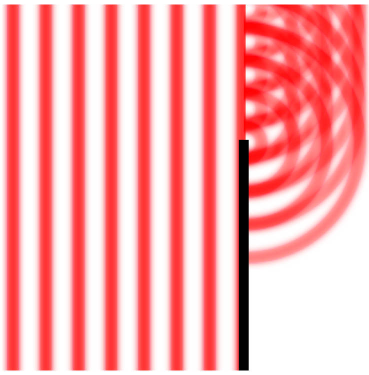


Figure 5.9: The knife-edge diffraction model. A plane wave is coming from the left, obstructed by an opaque half plane. All points that are not obstructed act as point sources themselves; the sum of all these point sources (the envelope in the picture) propagate 'around' the obstacle.

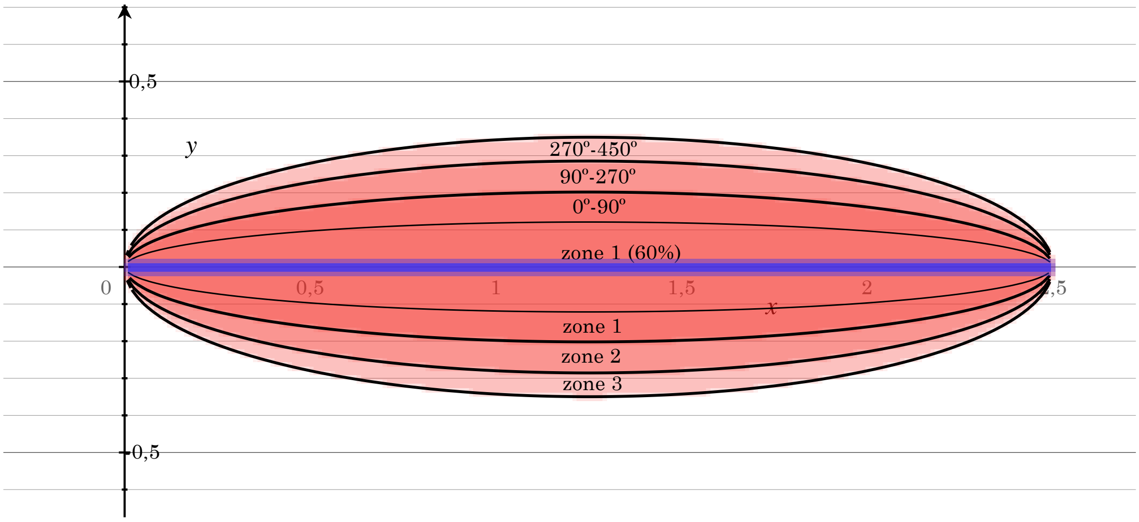
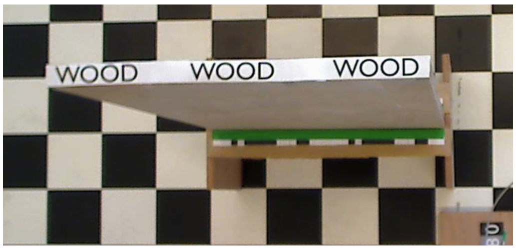


Figure 5.10: Top view of the first three Fresnel zones of a transmitter and receiver, 2.5 m apart, transmitting at 4.6 GHz. The dashed curve demarcates 60% of the first Fresnel zone diameter. In blue, the effective aperture cylinder between transmitter and receiver is added.

A signal can pass to the receiver if 60% of the first Fresnel zone is unobstructed [53, p. 20], [54]. Conversely, we might hypothesise that when 60% of the first Fresnel zone is completely obstructed, the transmitter is effectively hidden to the transceiver. To be on the safe side, we obstruct the the first until the third Fresnel zone. The radius of this zone is largest halfway transmitter and receiver, when transmitter and receiver are the farthest apart. Given the size of the set-up, this amounts to:

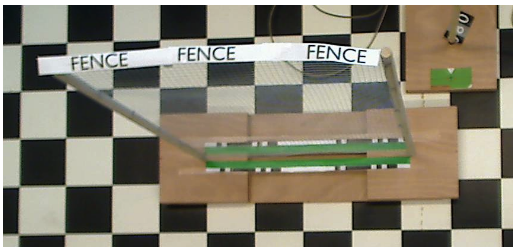
$$F_3 = \sqrt{\frac{3\lambda {}^1\!/\!_2 d_{\max} {}^1\!/\!_2 d_{\max}}{d_{\max}}} = \sqrt{\frac{3\lambda d_{\max}^2}{4 d_{\max}}} = \sqrt{\frac{3}{4}\lambda d_{\max}} = \sqrt{\frac{3}{4}\frac{3\times 10^8}{4.6\times 10^9} 2.5\,\mathrm{m}} = 35.0\,\mathrm{cm}. \tag{5.7}$$

Hence, real obstacles should be about 70 cm wide and at least 35 cm taller than the antenna height.

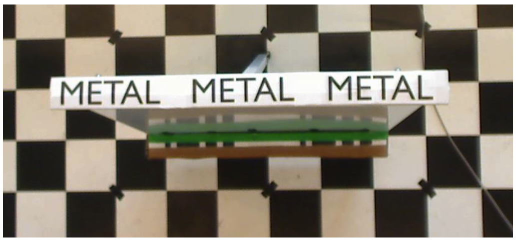


(a) The plywood panel.

(b) The cinder wall.



(c) The fence.



(d) The metal sheet.

Figure 5.11: Top views of the constructed obstacles.

The following obstacles were constructed (from low to high expected absorption), see Figure 5.11. A 61 × 122 × 3.6 cm plywood panel – two 18 mm Okoumé sheets glued together –, is available as a boundary case of little absorption (3.3 dB according to Figure 5.8). Next, a 70 × 120 × 10 cm cinder wall – made out of seven Ytong blocks – is built on a small chariot. We can expect it to absorb about 6 dB, according to Figure 5.8. A fence of 70 × 113 cm Quadro Galva, galvanised ∅0.65 mm wire, with square meshing of 12.7 × 12.7 mm is fixed between two wooden poles with plastic tie-wraps. Finally, a Commercial Of The Shelf (COTS) flap-over stand of 70 × 100 cm made of 1 mm sheet metal is used as the boundary case of all reflection. Additionally, the author can serve as an obstacle himself, being about 40 cm wide at the waist and 186 cm tall.

To find out to what extent these obstacles can be used to represent ODP channel conditions, we keep the transmitter and receiver antennas at 1 m separation and gradually shift an obstacle in between (Figure 5.12). We do this halfway transmitter and receiver (50%) and closer to the transmitter (25%) and repeat this experiment for all obstacle types. The leading edge range error as function of the partiality of the obstacle is plotted in Figure 5.13, using all measurements available at the end of the measurement campaign. The partiality is expressed as the lowest Fresnel zone that obstructed at both sides of the transmitter-receiver axis.

We notice that the wooden panel introduces no observable range error and conclude that it is effectively transparent. The cinder wall causes large errors close to its edges ($r = 0$), and seems to be transparant with a little bias through the middle ($r > 2$). We think the transient is caused by scattering at the rough edges of the wall, while the bias is caused by the lower wave velocity through cinder. The fence is surprisingly transparent, but the metal sheet and the researcher himself cause severe ranging errors.

In conclusion, metal indeed is the most severe obstruction and therefore is a good boundary case. Wood is almost transparent and therefore a good other boundary case. Except for the at the edges, the cinder wall causes a detectable but slightly delayed DP. A person is a severe obstruction as well, but not very practical and reproducible. Therefore, we will not use this obstruction in the sequel. In order to save time, we will not use the fence as well. Furthermore, obstructing two Fresnel zones is enough to cause Undetected Direct Path (UDP) conditions for metal and a delayed, obstructed direct path for cinder.

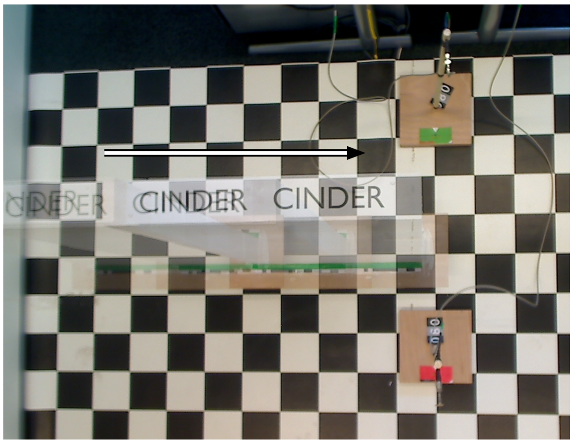
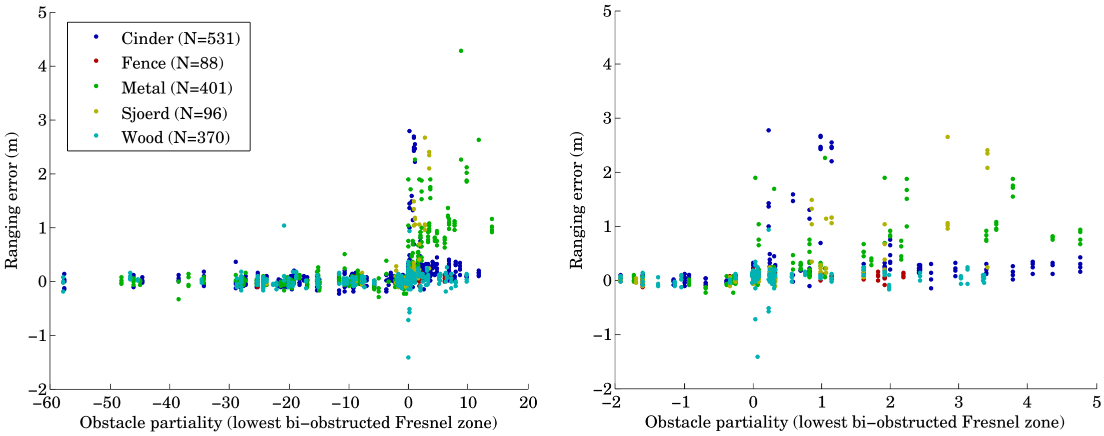


Figure 5.12: Top-view of the obstacle partiality experiment. The obstacle is moved and is oriented perpendicular to the transmitter-receiver axis.



(a) All takes with an obstacle on the floor.

(b) Zoom around $r = 0$

Figure 5.13: Range error as function of obstacle partiality r, for different materials.

5.2.6 Surroundings

The dimensions of the measurement set-up selected in Section 5.2.4 (2×2.5 m) are somewhat smaller than a practical room or office. Carrying out measurements within this rectangle in a normal room might be too 'easy' for localisation, because the multipath components will arrive relatively late and attenuated. Therefore, they are more easily discernable from direct paths, also for ODP detection.

Given that the measurement rectangle is constrained and our wish that the room and the rectangle are of about the same size, we can only try to shrink the room. A small reverberation tent is a good candidate, because it can emulate different multipath environments. Ideally, we perform the measurement campaign once in a small office and several times in a reverberation tent that emulates dense, medium and sparse multipath environments. This combination (a real indoor environment and a reverberation chamber) was used before [55]. Because of time, we only use a small office of about 3×5 m (Holst Centre Eindhoven, 3-010).

During the measurements, the researcher will sit still in the corner of the room. We suppose that he has negligible influence on the measurement, as he is outside of the measurement rectangle and at a fixed position. We say that all measurements that are taken in the same room with the same obstacle positions and obstacle types are taken in the same *scene*.

5.2.7 Principle and Number of Nodes

With range measurements, one can perform classical localisation as explained in Section 2.1, or co-operative localisation as outlined at the end of Section 3.2.

In classical localisation, only one of the nodes is a target and all other nodes are anchors. It is reasonable to assume all materials are reciprocal and linear, so there is no difference between the transmitter and receiver in a range measurement.

Let n be the number of nodes. In classical localisation, only one of them is the target, so the number of range measurements is:

$$N_{\text{class}} = n - 1. \tag{5.8}$$

These range measurements can be used as-is (Time of Arrival (ToA)) to emulate two-way ranging. Alternatively, they can be subtracted to simulate Time Difference of Arrival (TDoA) ranging. The minimum number of anchors for 2D ranging is three in both cases, so the minimum number of nodes is four.

In co-operative localisation, at most one range between every node and every other node needs to be measured. As a result, if there are n nodes, the maximum number of range measurements becomes the arithmetic progression:

$$N_{\text{co,max}} = \sum_{i=1}^{n-1} n = \binom{n}{2} = \frac{n-1}{2} n \lessapprox \frac{n^2}{2}. \tag{5.9}$$

Please note that this really is the maximum number of range measurements: it is also possible to localise with fewer range measurements. The lower bound can be understood from the case of classical localisation: to unambiguously locate a node in the horizontal plane, there should be at least three range measurements from every node to other nodes that are either localised or known *a priori*. Conversely, in the optimum case, every range measurement is useful and necessary to the nodes at both ends. Therefore, every node requires three times half a range measurement. In general, if we require every node to have m range measurements, the minimum number of total range measurements is:

$$N_{\text{co,min}} = n \cdot \frac{m}{2}. \tag{5.10}$$

The number of range measurements for classical localisation, minimum co-operative ($m = 3$ for 2D localisation) and maximum co-operative is plotted in Figure 5.14.

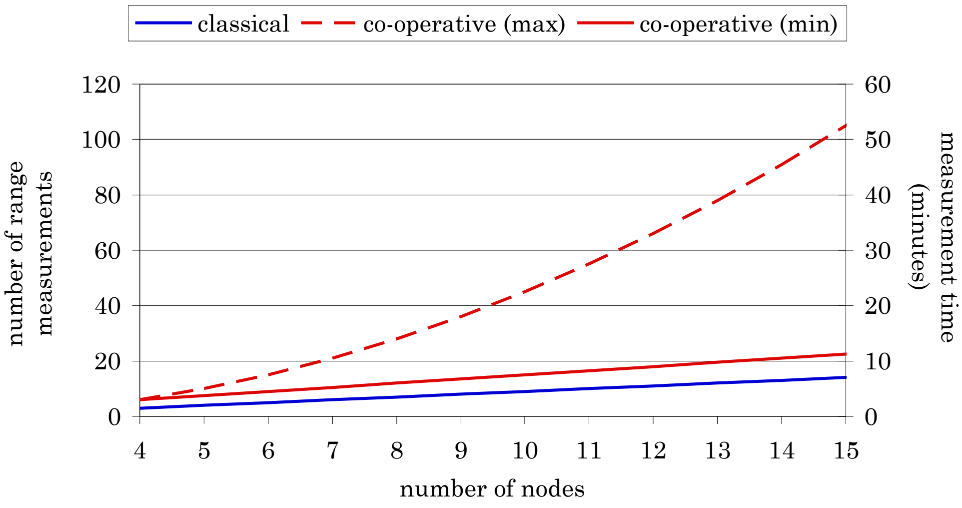


Figure 5.14: Number of range measurements given a classical and a fully co-operative algorithm. Dashed is the approximate expression for the co-operative case ($n^2/2$).

The number of range measurements will severely impact the consumed time. Moving the antennas and taking measurements will be the most frequent action, as opposed to, e.g. changing obstacles. Therefore we model the measurement time in one scene to be the product of the number of range measurements and the duration of moving antennas and taking a measurement. After the optimisations described in Section 4.5.1, transmitting, receiving and saving of one frame at 5 GSs^{-1} takes 4.5 s. We decide to take four subsequent measurements to be able to say something about the repeatability of the ranging. From practice, we know it takes 10 s to move the antenna to the next position. Adding up these numbers gives about 30 s per range measurement. Using this knowledge, we added a time scale to Figure 5.14.

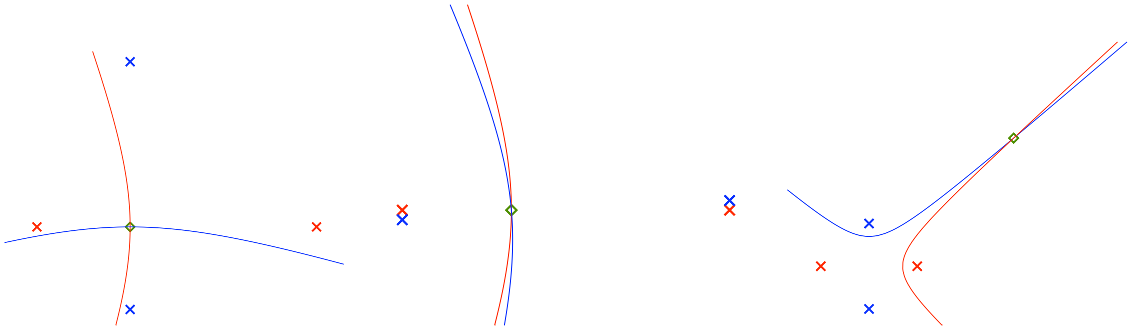
We decide to take measurements that allow testing of both classical and co-operative localisation algorithms. The number of nodes n and the number of neighbours m can be selected to fit the available time.

5.2.8 Geometry

In real applications, it might be necessary or interesting to localise nodes in the three-dimensional space. For this research, we reduce the number of measurements and the complexity of analysis by putting all the nodes in the same horizontal plane. We expect the height of this plane to have impact, because of ground and ceiling reflections. To reduce the duration of the measurement campaign, we decide to perform all measurements at a height of 80 cm, which is typical of a table-top application.

Given a plane, the nodes can be still be layed-out in many ways. Let us call the shape of the anchors with respect to the target the *geometry*. As can be understood from Figure 2.2, the perfect geometry for TDoA localisation occurs when the hyperbolas cross perpendicularly at the target, such as in Figure 5.15a. In that case, a ranging error causes a localisation error in the same order of magnitude. When the hyperbolas are almost parallel at the target node, only a slight ranging error causes a much larger localisation error along the grazing hyperbolas. The quality of the geometry can be quantified analytically using the Relative Geometric Dilution of Precision (RGDoP) [14, p.8-9].

To be able to investigate the effect of geometry, we use more anchors than strictly necessary for two-dimensional classical localisation. By selecting measurement subsets with better and



(a) Good geometry; the hyperbolas are perpendicular at the target.

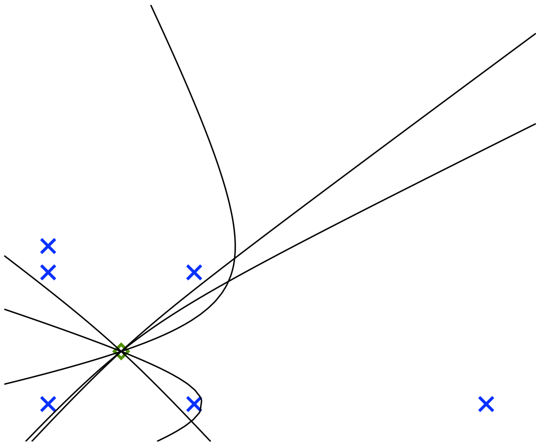
(b) Bad geometry; the hyperbolas are almost parallel at the target.

(c) Bad geometry; the hyperbolas are almost parallel at the target.

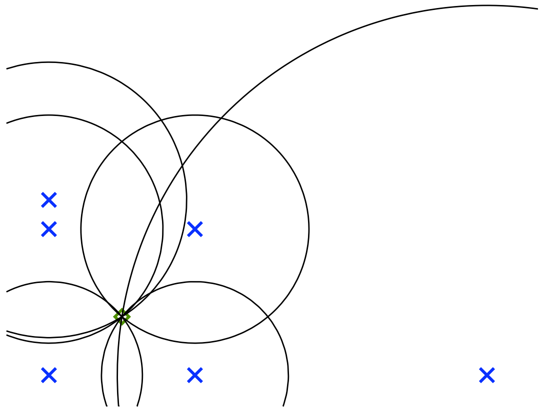
Figure 5.15: Different geometries and their effect on TDoA localisation accuracy. The red anchors generate the red hyperbola, the blue anchors generate the blue parabola. The green diamond is the target location.

worse geometry, we can see the effect of geometry on localisation accuracy. We intend to span the range from good to bad geometry, so we superimpose the anchor arrangements of Figure 5.15 and get the asymmetrical anchor arrangement of Figure 5.16. Notice that any combination of two hyperbolas or, equivalently, three circles can be drawn from the measurements, to yield a better or a worse geometry (Figure 5.16a and 5.16b). Selecting a combination of more than the minimum number of measurements, one can improve the geometry even more.

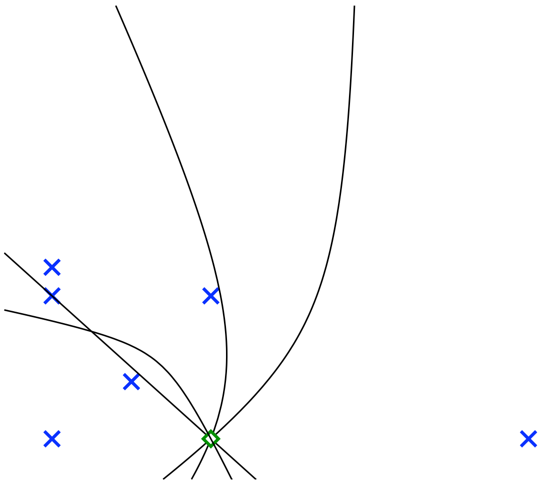
We can use the same node arrangement to test co-operative localisation. For that purpose we will measure all 21 possible ranges between the seven nodes. With these measurements, we can test both classical localisation (cf. Figure 5.16c and 5.16d) and co-operative localisation. However, to really test co-operative localisation, we need more nodes. We decide to use a grid of 5×4 nodes (Figure 5.17). Measuring all possible ranges would take 190 measurements, so we choose to measure only between every node and its neighbours, resulting in 55 measurements.



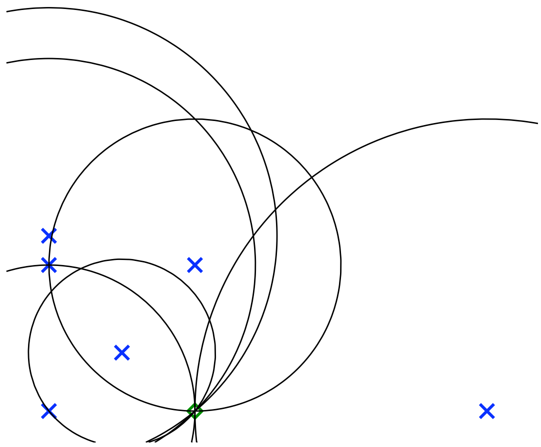
(a) 5 of the $6 \cdot 5 = 30$ hyperbolas that could be drawn from 6 ToA measurements. Note: this is synthetic TDoA, because ToA measurements are subtracted to mimic TDoA measurements.



(b) All circles that can be constructed from 6 ToA measurements.



(c) 5 of the $6 \cdot 5 = 30$ hyperbolas that could be drawn from 6 other ToA measurements in the same arrangement. Again, this is synthetic TDoA.



(d) All circles that can be constructed from 6 other ToA measurements in the same arrangement.

Figure 5.16: Asymmetrical anchor arrangement, providing subsets with good and bad geometry.

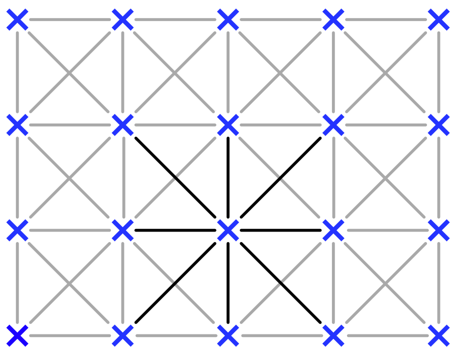


Figure 5.17: Grid of nodes, suitable for co-operative localisation. To reduce the number of measurements, only the range between every node and its nearest neighbours is measured.

Table 5.1: Summary of the selected parameter values.

Parameter	**Value(s)**
Receiver & Transmitter	IMEC's state of the art modules.
Protocol	IEEE 802.15.4a in mode 2.2 at 4.6 GHz.
Antennas	Microstrip circular pseudo-monopole with 7 mm ground plane, feed coming from the conductor side. Elevation 0°, vertical polarisation, azimuth: antenna groundplanes pointed at each other if possible.
Distance	Within a rectangle of 2 × 2.5 m.
Obstacles	{Nothing, 36 mm plywood, 100 mm cinder, 1.0 mm metal sheet}.
Surroundings	Small office (Holst Centre Eindhoven, 3-010).
Principle & #Nodes	{Classical localisation (7 nodes), co-operative localisation (20 nodes)}.
Geometry	In a horizontal plane at 20 cm from the ground, {in an asymmetrical pattern, in a grid}.

5.3 Measurement Plan

We will now pick a subset of the parameter values selected in the previous section, summarised in Table 5.1. Recall the goal of the measurement campaign: to take representative (i.e. typical and interesting) measurements, that allow to (a) evaluate the performance of ODP detection algorithms and (b) evaluate the performance of ODP-aware localisation.

As for the first part of the goal: to evaluate the ODP detection, we are a-typically interested in ODP and UDP channel measurements. During the selection of parameter values in Section 5.2, we already took 280 measurements, 135 of which with an obstacle on the floor. We have seen DDP, ODP and UDP channel measurements (Figure 5.13). These are saved to the measurement database and can therefore be used for fulfilling the first goal. The measurements we will take during the remainder of the campaign will also be added to the database, hence can contribute to the first goal as well. However, we think we have sufficient ODP channel measurements, so we can relax our a-typical interest in ODPs.

For the second goal, we will take measurements that actually allow to localise nodes. That is, we need multiple measurements taken in the same scene. What is a useful constellation of nodes, depends on the localisation principle: classical or co-operative.

For classical localisation, as mentioned in Section 5.2.8, we will take all 21 possible range measurements in an asymmetrical geometry with 7 nodes, to be able to study the effect of the geometry. We expect that the effect of an ODP will also depend on the geometry. That is: hiding some anchors will be worse than hiding other anchors. We pick two boundary cases and another case for the placement of the obstacle (Figure 5.18a). An obstacle at position A should have the least impact when all anchors are used to localise node 4, because range measurement 7–4 is almost redundant with 2–4. On the other hand, if only anchors 1, 2 and 7 are used, the range error will be amplified by the grazing circles (compare Figure 2.4d). An obstacle at position B should have medium impact if, for example, anchors 1,2 and 6 are used, because it generates a circle that is orthogonal with the other circles above the target. Finally, designate obstacle position C that hides two anchors with one obstacle, because it can constitute a 'difficult' case for localisation (for example, only use anchors 2, 5 and 7). We will perform this measurement with concrete, wooden and metal obstacles and once without obstacle. This experiment thus consists of 21 ranges × ((3 obstacle types × 3 obstacle positions) + once without), equals 210 range measurements.

(a) Obstacle positions for classical localisation. Note that all 21 possible inter-node range measurements will be taken, so all nodes can act as target and as anchor.

(b) Obstacle position for co-operative localisation. Note that from every node, only the ranges to the direct neighbours will be taken (see also Figure 5.17).

Figure 5.18: Obstacle and node configurations.

For co-operative localisation, we would like to evaluate the expectation that co-operative nodes can localise 'around' an obstacle (end of Section 3.2). For that reason, we suggested a larger number of nodes in Section 5.2.8. Given the grid of 5 × 4 nodes, there is only one position where we can really expect the nodes to localise around: in the middle of the constellation. For reasons of time, we only measure this scene with the cinder wall, the metal sheet and without obstacle. To reduce the measurement time even more, we will not point the antennas at each other, but always direct the groundplanes south. This experiment thus consists of 55 ranges × (2 obstacle types + once without), equals 165 range measurements.

5.4 Measurement Procedure & Automation

After the measurement, we would like to have all measured VDPs and respective metadata stored in the measurement database. (In fact, we store the oscilloscope capture and not the VDPs, because a VDP is an interpretation of the scope readings, as explained in Section 4.3.2. In principle, one should store observations and not interpretations.) Metadata is all information that is necessary to reproduce the measurement, such as antenna positions, orientations, obstacles, used transceiver and protocol.

The researcher could define a list of measurements that he plans to take, in terms of antenna positions and obstacles. Then, for every entry of the list, he could put the antennas and obstacles in the right position and take the measurement. Next, he should enter the metadata manually into the database. This method slows down the measurement procedure (655 measurements needed to be taken), and is error-prone because of the manual entering of metadata.

Alternatively, we could write a computer programme that generates the metadata of all measurements that should be taken in a convenient order. For every measurement, it presents the parameters to the researcher. The researcher then moves the antennas and obstacles accordingly. Consequently, the measurement is taken and saved. This would speed up the measurement cycle and reduce the risk of errors. Moreover, it would improve the reproducibility of the measurement: another researcher could run the programme in another room and meaningfully compare the results.

In reality, it is not feasible to generate all metadata in advance. For example, the experiments done in Section 5.2 needed to be done first, before the measurement plan of Section 5.3 could be drawn up. Therefore, for every series of measurements, the researcher writes a simple script that generates the metadata. Consequently, this smaller series of measurements is conducted by the computer programme. The advantage of this method is that the measurements of a series can be taken quickly after each other, while still being flexible enough to design new experiments if the results suggest so.

This programme is realised in MATLAB (Figure 5.19, Appendix D), and schematically shows the metadata of the current and the next measurement. A webcam is attached to the ceiling of the room, and visible markers are attached to antennas (Figure 5.20) and obstacles, so that the actual measurement configuration can be compared to the metadata. The photo is saved together with the measurements and the metadata, to aid the researcher in finding discrepancies between metadata and actual measurement configuration. To monitor the progress of the measurement (and to analyse the results on-the-fly), the measured PDPs are displayed as well. The measurement cycle for the researcher looks like this:

1. Arrange the antennas according to the displayed metadata.
2. Press 'Measure…'; a buzzer sounds to indicate the start of the measurement.
3. While waiting for the measurement to finish:
    1. Verify that the webcam photograph corresponds with the schematic view of the current metadata.
    2. Memorise the schematic view of the next metadata, i.e. memorise the antenna and/ or obstacle movement that needs to be done for the next measurement.
    3. Check that plausible PDPs are captured, i.e. compare the range estimates with the real range.
4. A bell sounds to indicate that the measurement is finished; quickly arrange the antennas and obstacles as memorised and press 'Measure…' again.
5. Go to step 3, until the last measurement of the programmed series is reached.

This cycle takes about 50 s if four takes are captured at 5 GSs^{-1} and if the antennas need to be pointed at each other. If the antennas always have the same orientation (south), it takes about 40 s.

Using this measurement automation, 280 preliminary measurements and 375 measurements for localisation were taken, together 655 range measurements. Most of the range measurements consist of four consequent takes, some less to save time, totaling to 2400 takes. The results will be discussed in the next chapter, and the performance of different ODP detection and mitigation techniques will be evaluated.

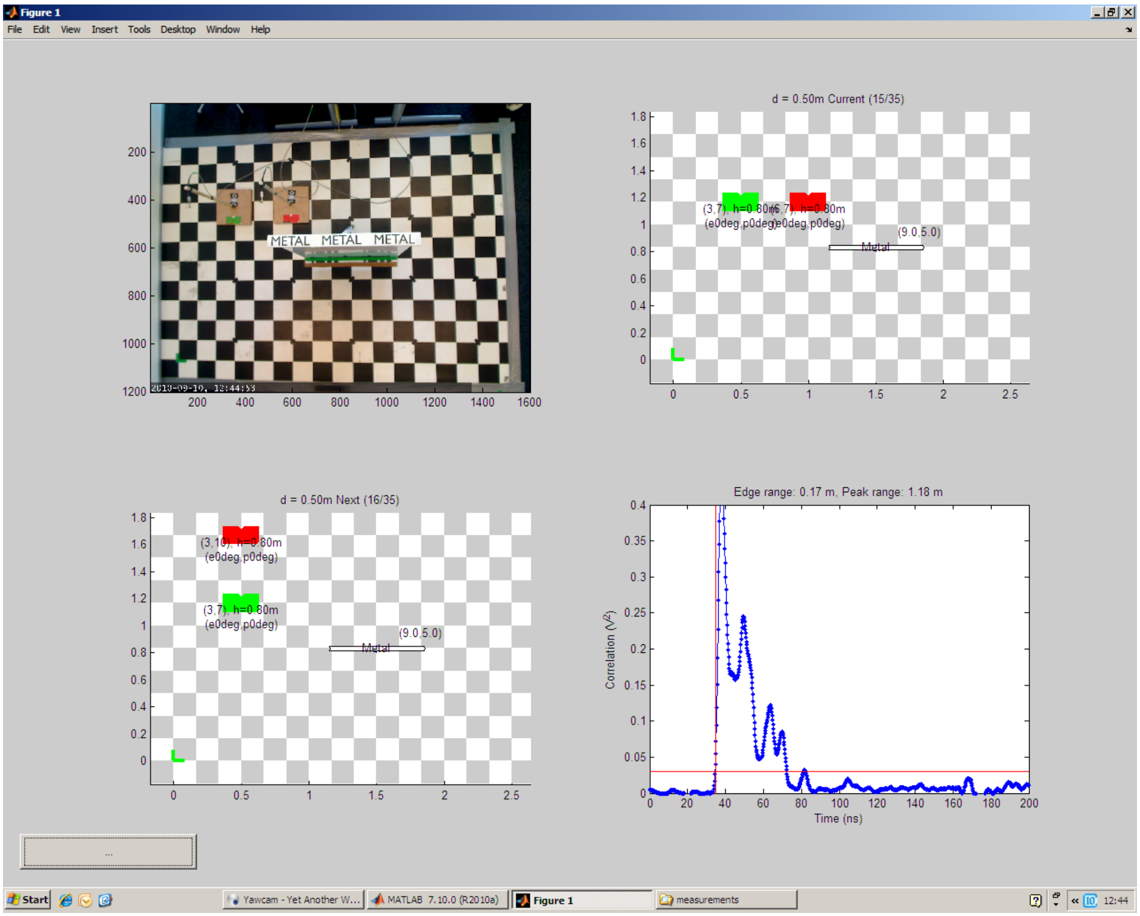


Figure 5.19: Screenshot of the measurement software. At the upper left, there is a webcam photo of the current scene and antenna positions. At the upper right, there is a schematical view of the measurement parameters that will be saved with this measurement. At the lower left, there is a schematical view of the measurement parameters of the next try. At the lower right, the measured PDPs are monitored.

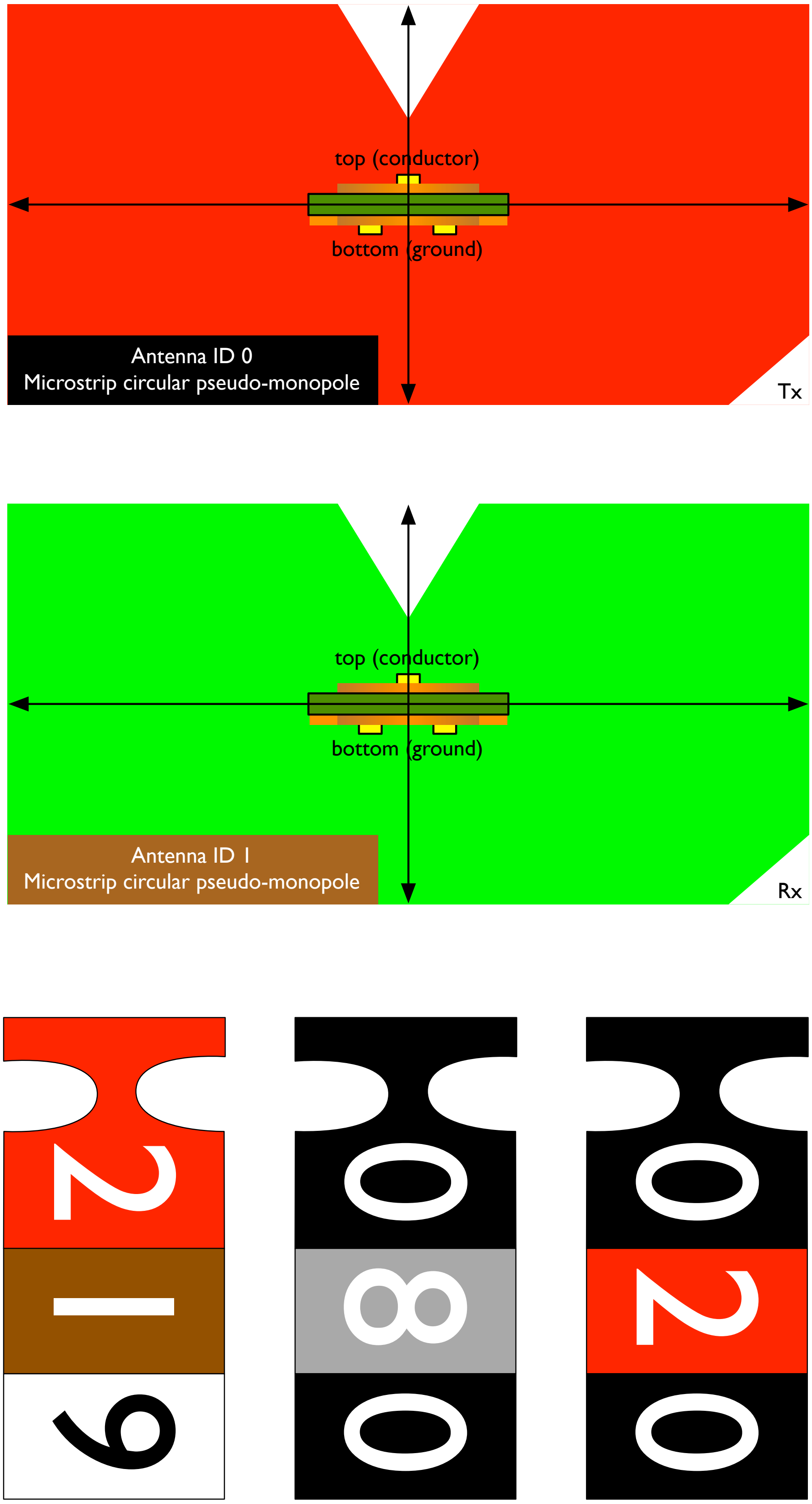


Figure 5.20: Printable antenna markers. The top-two markers are attached to the foot of the antenna stand, by means of push pins, to allow rotation of the marker. The bottom height markers are attached to the pole of the antenna stand by means of adhesive tape.

6 ODP Error Mitigation

In this chapter, the measurements taken in the measurement campaign of Chapter 5 are used to evaluate the Obstructed Direct Path (ODP) mitigation techniques for localisation of Chapter 3.

First, we use the range measurements to determine the best range estimating algorithm for line-of-sight (LOS) conditions. Then, we evaluate the competence of the features from Section 3.1 in predicting the ranging error caused by ODP conditions. Finally, we use a combined prediction to improve the localisation in ODP conditions, as described in Section 3.2.

6.1 ToA Detection

All time-domain ranging algorithms, to our knowledge, rely either on a channel impulse response estimate or Voltage Delay Profile (VDP) $\hat{h}_b(n)$ or a Power Delay Profile (PDP) $|\hat{h}_b(t)|^2$. In IMEC's ranging set-up, two algorithms to estimate the range from the VDP were used. These two algorithms are described and their performance is evaluated. Next, two improved algorithms are proposed and evaluated.

6.1.1 IMEC's Original Algorithms

First and most obvious, peak detection selects the instant where the power is highest. This algorithm is best understood by recalling that the power delay profile really is the cross correlation between the received and transmitted baseband signals. Therefore, where the cross correlation is highest, there is the delay that best describes the channel if it were just one time-shifted Dirac impulse. We expect this algorithm to be robust against changing signal strength, because then only the height of the peak is affected, but not its (time) position. At the same time, we expect this algorithm to be vulnerable to NonDominant Direct Path (NDDP) conditions, where a MultiPath Component (MPC) might be stronger than the (attenuated) direct path.

Second, leading edge detection selects the instant where the power first exceeds the noise floor, like in [56]. If we now interpret the VDP as a channel impulse response, we understand that the first leading edge is most probably caused by the direct path component. We expect this algorithm to be robust against attenuated direct path conditions, because it will always use the onset of the first detectable path. Therefore, even an early multipath that partly overlaps the direct path will not affect the detected edge. At the same time, the threshold value is only based on the noise floor (and not on the signal strength). Depending on the signal strength, the leading edge of the signal will cross the threshold earlier or later. We expect the latter effect to bias the results systematically (weaker signals cross the threshold later, hence appear farther away). Furthermore, random fluctuations of the noise floor will cause random (zero-mean) errors, to first order (linear) approximation.

In Figure 6.1, we compare the performance of both algorithms in LOS conditions, i.e. with a free first Fresnel zone ($r < -1$, see Figure 5.13). Especially the peak detection algorithm has many positive outliers (the indicated take is depicted in Figure 6.2a). Still, the lower envelope of the takes follows a slope that corresponds with the speed of light. Linear regression with a Welsch-weighted least squares criterion [57] turns out to follow the bottom envelope of the takes best[1]. We perform one fit where the slope is fixed *a priori* (i.e. to the speed of light) and another fit, where the slope is also determined by linear regression. The goodness of the fit is judged by the mean absolute error. The same two fitting methods are applied to leading edge detection.

[1] …when compared to ordinary least squares (no weighting), bisquare, Cauchy, fair, Huber, logistic and Talwar weighting [58].

In an attempt to understand and mitigate the outliers, we plotted the VDPs of the outliers indicated with a red arrow in Figure 6.2a. We note from Figure 6.2a that even in LOS conditions, the highest peak may not be caused by the direct path. Comparing Figure 6.2b to 6.2a, we note that in 6.2b many samples of the leading edge pass before the threshold is crossed. In contrast, in 6.2a, only five samples of the leading edge pass before the threshold is crossed. Furthermore, the VDPs of weak signals (large antenna distance) cross the threshold almost at the end of the leading edge, where strong signals cross the threshold at the beginning of the leading edge. This might explain the higher slope of the fitted line in Figure 6.1b, corresponding to a lower apparent propagation speed of $2.70 \times 10^{8}\,\mathrm{ms}^{-1}$.

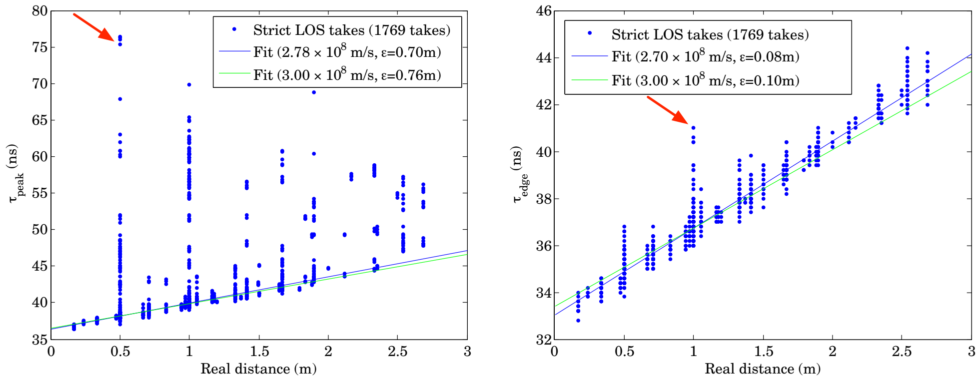
6.1.2 Proposed Improvements

To improve both algorithms, we propose the following modifications based on the observations above. For the peak detection, we suggest that not the global maximum, but the first local maximum after crossing the threshold is used, like in [59]. As long as the first indirect path does not overlap the direct path, it should be possible to detect the position of the direct path. See the dashed blue vertical line in Figure 6.2a.

To improve the edge detection, start by noting that we would like to see the start of the leading edge, even below the noise floor. However, the samples below the threshold have limited reliability because of the noise. We propose to estimate the real – invisible – start of the leading edge by linear extrapolation as follows. First smooth the PDP samples close to the threshold crossing using a 5-point moving average. Second, take the derivative and find the maximum slope to the left of the crossing, but right of the nearest local minimum. Use this maximum slope to draw a tangent line and take the crossing of this tangent with the $|\hat{h}_b| = 0$-axis to be the start of the leading edge. See the dashed red lines in Figure 6.2b.

The performance of these algorithms is illustrated in Figure 6.3, fitted to and applied on the LOS takes. Note that the fitted slope of Figure 6.3b indeed is closer to the speed of light than in Figure 6.1b. We also applied the algorithms on all takes (both non-line-of-sight (NLOS) and LOS), using the former fit and using a new fit. The accuracy is reported in Table 6.1, the precision in Table 6.2.

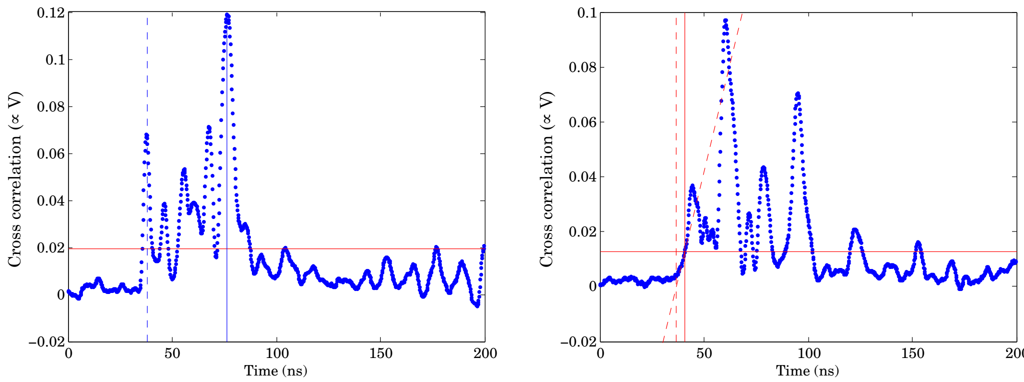
We conclude that both modifications have improved the ranging accuracy and precision with respect to their original algorithms. The extrapolated edge detection performs best for all sets, for all fits. The linear extrapolation still is a first order approximation to the pulse shape, so there may still be a distance-dependent bias of $\tau_{\mathrm{edge,extra}}$. Therefore, it seems allowable to use a slope fit, fitted on the LOS takes.



(a) Global peak detection.

(b) Leading edge detection.

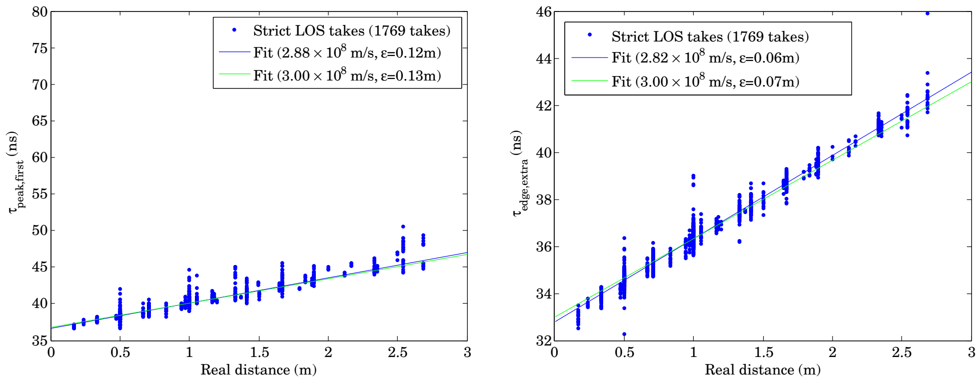
Figure 6.1: Detected delay versus the real distance, according to the original peak detection and edge detection algorithms. Only measurements that are LOS ($r < -1$, see Figure 5.13). Note that cabling and processing delay has to be subtracted (about 37 ns and 34 ns, respectively). The takes are approximated with a fitted slope (blue) and with a fitted offset (green). The takes indicated with arrow are elaborated in Figure 6.2.



(a) Peak detection mistake; one of the indirect paths is stronger than the direct path. The solid solid blue line is the time of the global peak, the dashed blue line indicates the first local peak after crossing the threshold (in red).

(b) Leading edge detection mistake; due to a high threshold with respect to the direct path power, we detect the edge too late. The solid red lines indicate the crossing of the threshold, the dashed red lines indicate the extrapolation of the leading edge.

Figure 6.2: Power delay profiles of ranging outliers in LOS conditions.



(a) First peak detection.

(b) Extrapolated leading edge detection.

Figure 6.3: Modified peak detection and edge detection algorithm performance. Only measurements that are LOS ($r < -1$, see Figure 5.13). Again, the takes are approximated with a fitted slope (blue) and with a fitted offset (green).

Table 6.1: Comparison of the mean absolute error in meters for the different detection algorithms. 'LOS' are 1769 takes with an obstacle partiality $r < -1$ (Figure 5.13), 'all' are all 2400 takes.

applied on...	LOS		all		all	
fitted on...	LOS		LOS		all	
fit type	slope	offset	slope	offset	slope	offset
edge	0.08	0.10	0.18	0.21	0.17	0.20
edge,extra	0.06	0.07	0.14	0.15	0.14	0.15
peak	0.70	0.76	1.35	1.46	1.24	1.46
peak,first	0.12	0.13	0.21	0.22	0.20	0.22

Table 6.2: Comparison of the standard deviation of the error in meters for the different detection algorithms. 'LOS' are 1769 takes with an obstacle partiality $r < -1$ (Figure 5.13), 'all' are all 2400 takes.

applied on...	LOS		all		all	
fitted on...	LOS		LOS		all	
fit type	slope	offset	slope	offset	slope	offset
edge	0.12	0.15	0.39	0.44	0.38	0.44
edge,extra	0.09	0.10	0.33	0.36	0.33	0.36
peak	1.59	1.72	2.29	2.48	2.12	2.48
peak,first	0.19	0.21	0.45	0.47	0.43	0.47

6.2 Error Prediction

We will now use the features listed in Section 3.1 to predict ranging errors.

Note that most literature speaks about 'NLOS detection'. However, of course, we are not interested in optical obstructions of the line between transmitter and receiver. Therefore, some literature uses the term 'ODP detection', which expresses the fact that we consider the channel conditions for the frequency band of interest (see also Figure 2.7). But ultimately, we are also not interested in radio frequency obstructions between transmitter and receiver. In fact, trying to achieve NLOS or ODP detection could be considered harmful. For example, if a wooden panel is placed between transmitter and receiver, there is both an NLOS and an ODP condition. However, the Direct Path (DP) is only attenuated little and almost no extra delay is introduced by propagation through the panel. If will be difficult to 'convince' an algorithm to detect this ODP or NLOS condition, because the panel is almost transparent for our band of interest. Even if we succeed, the detection algorithm will be very sensitive and, consequently, unduly classify many LOS and Dominant Direct Path (DDP) channels as untrustworthy. And in the end, the range measurement with the wooden panel *was* trustworthy.

Therefore, from here on out, we will use the features not to perform NLOS or ODP detection (binary classification), but to predict the ranging error (continuous estimation). By ranging error, we mean the difference between the range estimate by extrapolated leading edge detection and the real distance. We will use *all* features to predict the error, because we hypothesise that will yield a better error prediction. Let $f_1, f_2, \ldots, f_n$ be all feature values. Let $\hat{\varepsilon}_i(f_i)$ be the function that predicts the ranging error based on feature f_i. These predictions $\hat{\varepsilon}_1, \hat{\varepsilon}_2, \ldots, \hat{\varepsilon}_n$ need to be combined to obtain one joint error prediction ε like in Figure 6.4.

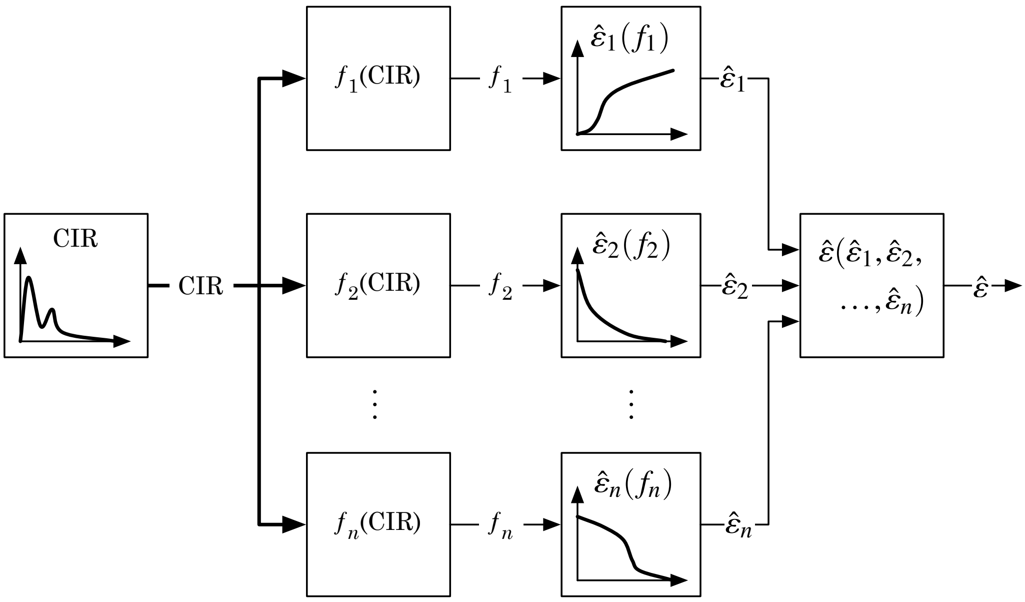
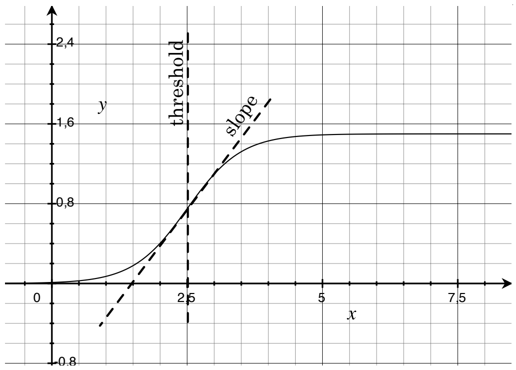


Figure 6.4: Multiple features f_i are used to produce multiple error predictions $\hat{\varepsilon}_i$, combined to obtain one joint error prediction $\hat{\varepsilon}$.

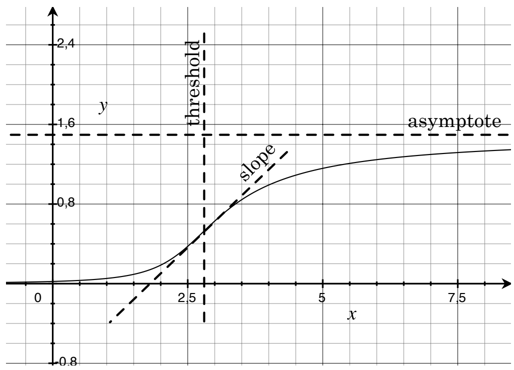
This approach leaves us with two questions. First, how should the error estimates $\hat{\varepsilon}_i$ be obtained from the feature values f_i, i.e. what is $\hat{\varepsilon}_i(f_i)$? Second, how should these error estimates be combined to obtain a joint error estimate, i.e. what is $\hat{\varepsilon}(\hat{\varepsilon}_1, \hat{\varepsilon}_2, \ldots, \hat{\varepsilon}_n)$?

6.3 Error Estimates

Concerning $\hat{\varepsilon}_i(f_i)$, unfortunately, no analytical models exist that predict the *kind* of relation between ranging error and each of the features. As literature often sets a threshold to a feature to distinguish between ODP and DDP, and the error is higher in ODP conditions, it seems reasonable to search for a monotonically increasing or decreasing relation. Also, we should be robust against feature outliers, so we look for a relation that saturates at extreme values. For the same reason, the hyperbolic tangent function is a popular activation function in artificial neural networks [60, p. 16]. This function can be characterised by a decision point and a slope, see Figure 6.5a. As it is used for decisions with two extremities, there is no need to specify its amplitude; there is an defined minimum and maximum value. In the case of estimating a ranging error, there is a minimum value, because we suppose the ranging error to be always positive. There is, however, no obvious maximum, so we look for a function that keeps rising, like a shifted arctan(·) function. For the error to approach zero, the argument must be large-negative. Therefore we choose to square the shifted arctan, resulting in a function characterised by threshold, slope and asymptote (Figure 6.5b).



(a) $y = 1.5(1/2 \tanh(x-2.5)/1 + 1/2)$



(b) $y = 1.5 \cdot (1/\pi \arctan(x-2.5)/1) + 1/2)^2$

Figure 6.5: Saturating fitting functions.

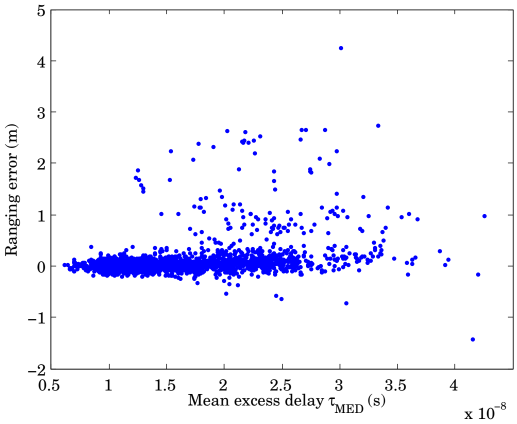
For each of the features mentioned in Section 3.1, we will now plot the ranging error as function of the feature value, to assess the relation. Because it is not easy to judge statistical properties by a 2400-point scatterplot, a bivariate histogram is also provided. This histogram is normalised along the error axis, i.e. the histogram estimates $\Pr\{\epsilon|f_i\}$. The histogram uses non-uniform bins along the feature axis, i.e. the bin widths are chosen to have the same number of takes in every bin. Finally, a squared arctan is manually fitted to follow the means of the bins.

The mean excess delay and ranging error of all takes are plotted in Figure 6.6a. The statistical properties of the relation are more easily judged in the bivariate histogram of Figure 6.6b. To calculate the mean excess delay, all PDP samples above −20 dB with respect to the global peak were taken into account. Note that the mean follows the fit quite smoothly. It is interesting to see that high mean errors correlate with high spread of the error.

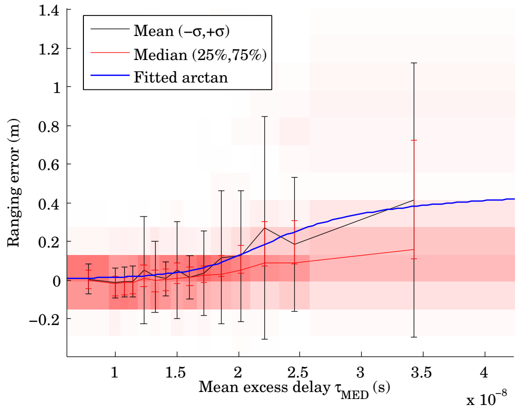
The Root Mean Squared (RMS) delay spread is also calculating by considering all PDP samples higher than −20 dB with respect to the global peak. The correlation with ranging error (Figure 6.7) looks similar to that of the mean excess delay, so the two features may be strongly correlated. Again (even more so), a high mean error correlates with a high error spread.

The K-factor estimate is calculated by dividing the energy (time integral over the power) in the first 7 ns after crossing the threshold over the next 70 ns of the PDP (Figure 6.8a). High K-factors are indicative of unobstructed, dominant direct paths, hence low error (Figure 6.8b).

The kurtosis is calculated from all channel impulse response samples above the noise threshold. The correlation with the error is weak (Figure 6.9), which might be caused by the distortion of the Variable Gain Amplifier (VGA) (Figure 4.14). Also the discrepancy between mean and median show that the kurtosis will not be a very reliable predictor of range errors.

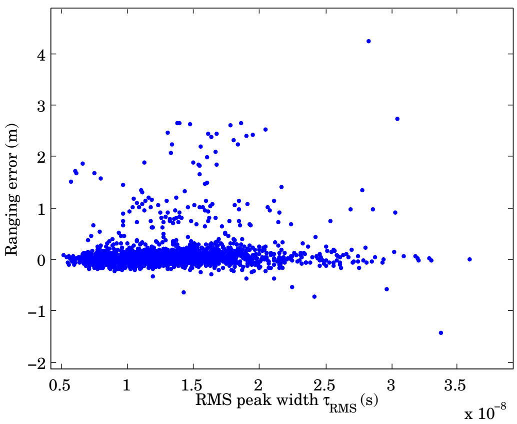


(a) Scatterplot with $N = 2400$ takes.

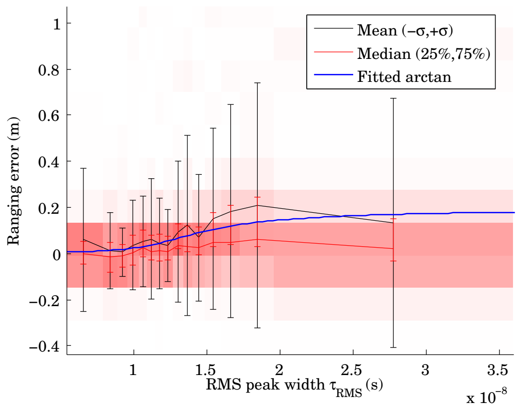


(b) Bivariate histogram, 160 takes per mean excess delay bin, including fitted $\hat{\varepsilon}_{MED} = 0.5 \cdot (1/\pi \arctan((\tau_{MED} - 20 \times 10^{-9})/6 \times 10^{-9}) + 1/2)^2$

Figure 6.6: Correlation of measured τ_{MED} with ranging error.

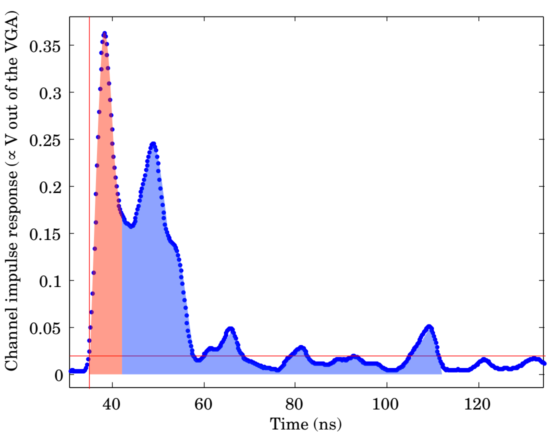


(a) Scatterplot with $N = 2400$ takes.

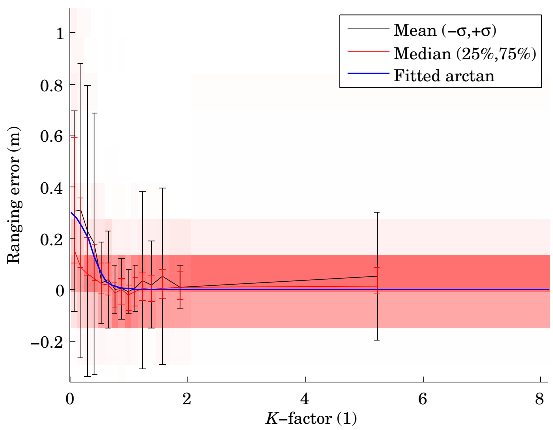


(b) Bivariate histogram, 160 takes per RMS delay spread bin, including fitted $\hat{\varepsilon}_{RMS} = 0.2 \cdot (1/\pi \arctan((\tau_{RMS} - 12 \times 10^{-9})/4 \times 10^{-9}) + 1/2)^2$

Figure 6.7: Correlation of measured τ_{RMS} with ranging error.



(a) Determining the K-factor from the channel impulse response. First 7 ns (red) are considered direct path, the next 70 ns (blue) scattered paths.



(b) Bivariate histogram of the estimated K-factor (160 takes per bin) with fitted $\hat{\varepsilon}_K = 0.4 \cdot (1/\pi \arctan((\tau_{RMS} - 0.45)/-0.2) + 1/2)^2$.

Figure 6.8: K-factor.

The rise time τ_{rise} histogram is plotted in Figure 6.10. Note that the typical rise time for low-error measurements is about 2.5 ns. We expected that higher rise times would correlate with higher errors, but both for lower and higher rise times, the expected error seems to increase. We cannot think of a physical explanation for this effect, so fitting to this effect could be harmful for the generality of our model. Therefore, we only fit the arctan to the increasing error to the right of 2.5 ns.

The coherence bandwidth is estimated by taking the Discrete Fourier Transform (DFT) of the PDP, and finding the lowest frequency for which the normalised amplitude drops below a threshold. An example normalised 1000-point DFT of an impulse is shown in Figure 6.11a. Note that for a given sample frequency, the frequency resolution is determined by the number of DFT points. For example, if the signal is sampled at 5 GSs^{-1} and a 1024-point Fast Fourier Transform (FFT) is used, the frequency resolution is $5 \times 10^9/1024 = 4.9$ MHz. Looking at the example, it is clear that a high threshold (0.9 or even 0.5) will already be crossed after a few DFT points. Consequently, the resolution of the coherence bandwidth will be low. Therefore, we lower the threshold to obtain a higher resultion. Experimentally, we found that a threshold of 0.2 on a 1000-point DFT gives a coherence bandwidth that best correlates with ranging error (Figure 6.11b).

The path loss exponent is estimated from the relation between DP power $\hat{E}_{\text{DP}}$ and range estimate $\hat{r}$ as follows:

$$\hat{\lambda} = \log_{\hat{r}}\left(\frac{E_{\text{DP,1m}}}{\hat{E}_{\text{DP}}}\right), \tag{6.1}$$

where $\hat{E}_{\text{DP}}$ is the energy in 4 ns centered around the first peak. To evaluate the feature, first $E_{\text{DP,1m}}$ needs to be determined. We do this by fitting an inverse square to $\hat{E}_{\text{DP}}$ as function of the real range, for all LOS takes (Figure 6.12a). Using this calibration, the estimated path loss exponent $\hat{\lambda}$ is calculated. Singular values as extreme as $\hat{\lambda} = -6000$ and $\hat{\lambda} = +4000$ are reached, but most values concentrate around $\hat{\lambda} = 2$, as expected. For $\hat{\lambda} > 2$, we see the expected trend of increasing error.

One feature was thought up during the measurements (and not taken from the literature): the time between the first (local) peak and the global peak, or *global peak delay*. Actually, it is the difference between τ_{peak} and $\tau_{\text{peak,first}}$. The measurement and performance of this feature is illustrated in Figure 6.13.

Two features were not implemented in lack of time: the fitted exponential decay and the number of paths. As mentioned in Section 3.1.6, the fitted exponential decay may not add much information compared to the RMS delay spread τ_{RMS}. The number of detected paths involves implementing or using a path detection algorithm like CLEAN, so the complexity may be too high for smart dust nodes anyway.

There is one interesting general tendency in all histograms: the expected (mean) error loosely correlates with the spread in the error. That suggests that we can use one number ($\hat{\varepsilon}_i$) to estimate both the error in the range measurement and the uncertainty in the range measurement.

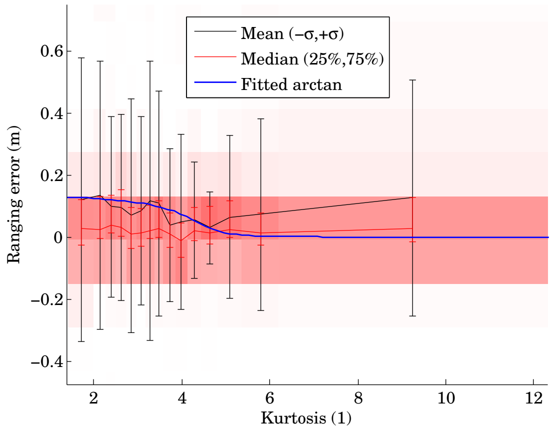
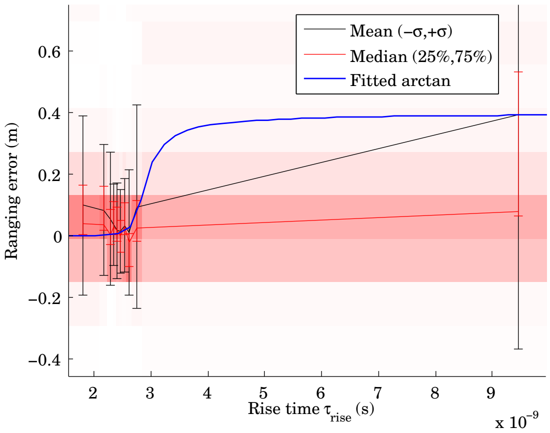
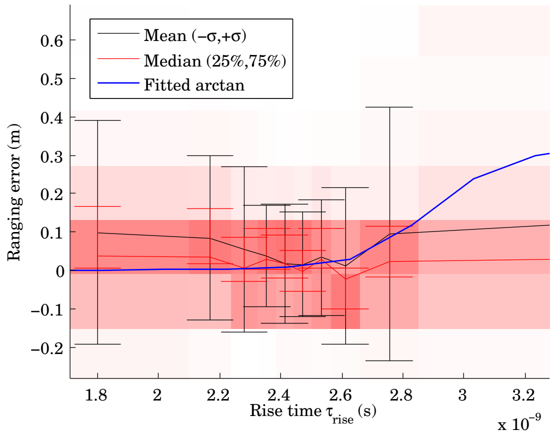


Figure 6.9: Sample kurtosis, fitted with $\hat{\varepsilon}_\kappa = 0.15 \cdot (1/\pi \arctan((\tau_{\text{RMS}} - 4.5)/-0.7) + 1/2)^2$

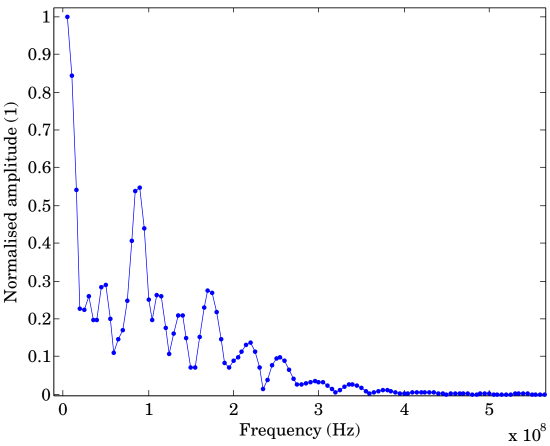


(a) Complete histogram.

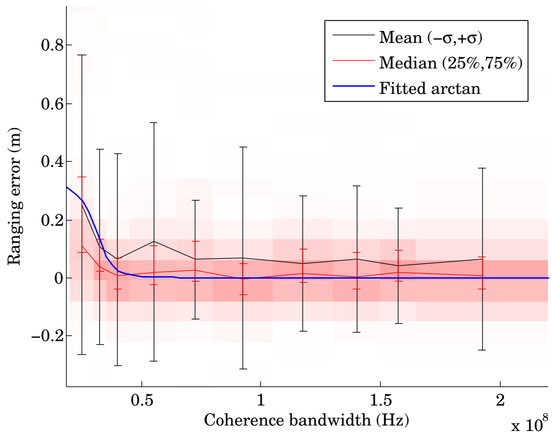


(b) Zoom around $\tau_{\text{rise}} = 2.5\,\text{ns}$.

Figure 6.10: Bivariate histogram of τ_{rise} (120 takes per bin) with fitted arctan.

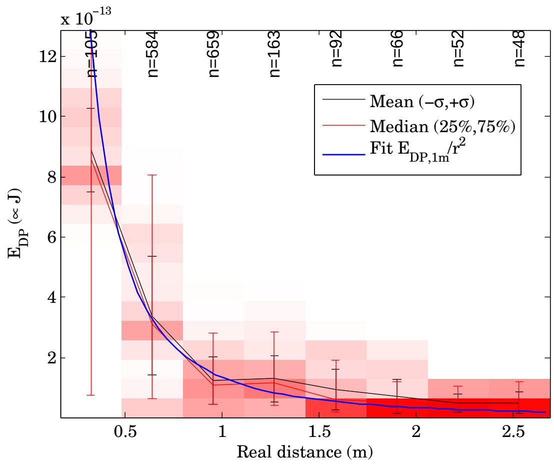


(a) Example 1000-point FFT of a PDP sampled at $5\,\text{GSs}^{-1}$.

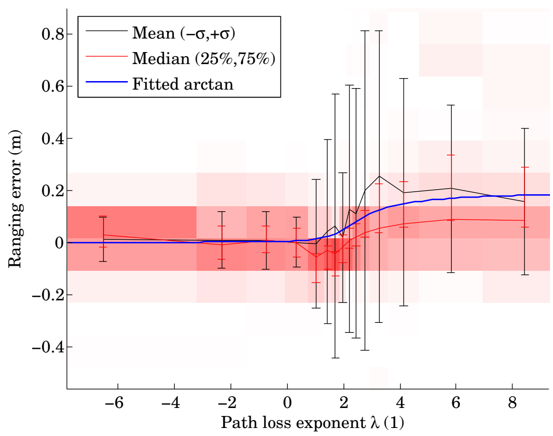


(b) Bivariate histogram (about 240 takes per bin) of the coherence bandwidth $B_{\text{coh},0.2}$ with fitted $\hat{\varepsilon}_\text{B} = 0.4 \cdot (1/\pi \arctan((\tau_{\text{RMS}} - 34 \times 10^6)/-6 \times 10^6) + 1/2)^2$.

Figure 6.11: Coherence bandwidth.

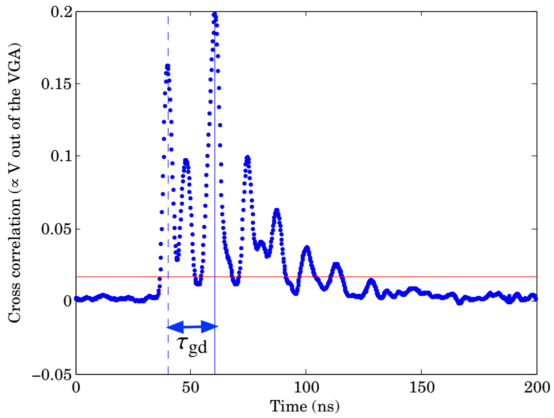


(a) Calibration of the pathloss estimator, including the Welsch-fitted $\hat{E}_{\mathrm{DP}} = 1.3384 \times 10^{-13}/r^2$.

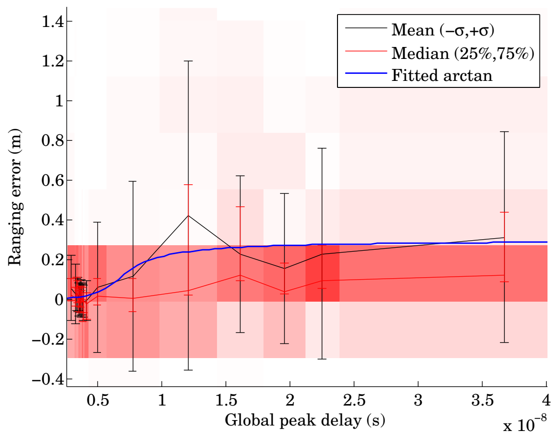


(b) Bivariate histogram (116 takes per bin) of the estimated path loss exponent around $\hat{\lambda} = 2$ with fitted $\hat{\varepsilon}_\lambda = 0.2 \cdot (1/\pi \arctan((\tau_{\mathrm{RMS}} - 2)/1) + 1/2)^2$.

Figure 6.12: Pathloss estimation.



(a) Example of measuring the global peak delay τ_{gd}.



(b) Bivariate histogram (160 takes per bin) of the global peak delay τ_{gd} with fitted $\hat{\varepsilon}_{\mathrm{gd}} = 0.3 \cdot (1/\pi \arctan((\tau_{\mathrm{gd}} - 6 \times 10^{-9})/2 \times 10^{-9}) + 1/2)^2$.

Figure 6.13: Global peak delay.

6.3.1 Combining the Error Estimates

Now that we have several error estimates $\hat{\varepsilon}_i$, how should we combine them? If the estimator functions $\hat{\varepsilon}(\cdot)$ are well fitted, the estimates are expressed in meters. Therefore, it makes sense to take a linear combination of the individual error estimates. Some error estimates correlate better with the real error than others, so we could use the sample correlation of each estimate with the real error as its weighting coefficient. However, this assumes that the error estimates are uncorrelated amongst each other. As the estimates measure related phenomena (temporal spread), this is a questionable assumption. To assess the correlation between the estimates, the correlation matrix is visualised in Figure 6.14. We can see that there is correlation between the RMS delay spread and mean excess delay, as expected. Also the coherence bandwidth is significantly correlated with these two temporal spread measures. Judging from the bottom row or leftmost column, kurtosis seems to be of negligible use in predicting the ranging error.

To estimate the ranging error, knowing that the individual error estimates are correlated, we find the optimum weighting coefficients by multilinear regression on all 2400 takes. That is, the coefficients are chosen to minimise the error in predicting the real ranging error. Note that the result of multilinear regression is not necessarily unique nor robust. It is not unique because multiple sets of coefficients may exist that yield the same error in predicting the range error. It is not robust because, two highly correlated estimates may (a) get the same coefficients, or equally well (b) a high and a low coefficient. On average, both regressions will yield the same error predictions. However, an accidental error in measuring the first estimate will have moderate impact in case (a), but a large impact in case (b). More robust multilinear regression methods exist (such as Support Vector Machines (SVMs) [21]), but are not tried in lack of time.

The multilinear regression yields the coefficients visualised in Figure 6.15. The negative coefficient for the RMS delay spread can be explained by knowing that RMS delay spread and mean excess delay are strongly correlated. Therefore, given that the mean excess delay is already incorporated with a high coefficient, the RMS delay spread is already incorporated 'too much', which needs correction.

If we then combine the ranging error estimates according to these weighting coefficients, we obtain a joint error estimate (Figure 6.16). Indeed we observe the same trend: a high error estimate is indicative for a higher mean error, but also for a higher error spread (higher uncertainty).

6.4 Localisation using Error Prediction

Now that we have one number (the joint error estimate $\hat{\varepsilon}$) associated with each range measurement, that correlates with the real ranging error and uncertainty, how to localise? For explanation purposes, let us consider the case of a severe obstruction (a metal sheet) hiding one anchor from the target, while five anchors are unobstructed in a good geometry (Figure 6.18). In the case of ODP-ignorant (*simple*) localisation, the ranging error of the hidden anchor causes a 40 cm localisation error (Figure 6.18a).

If the joint error estimate is correct, it makes sense to subtract this estimate from the range measurement, and use these corrected range measurements to localise. In this case, the error of this *corrected* localisation is 20 cm, a significant improvement. However, we see that the ODP range measurement is not corrected enough.

Therefore, we could set a threshold to the joint error estimate; all range measurements with a higher joint error estimate are discarded, all range measurements with a lower joint error estimate are corrected and used for localisation. We call this strategy *combined*, because we apply both a discarding and a correcting technique. Where to put the joint error estimate threshold (Figure 6.16a)? Let us first put a threshold on the *real* ranging error, i.e. above which error do we deem range measurements unusable? According to Table 6.2, the standard dev-

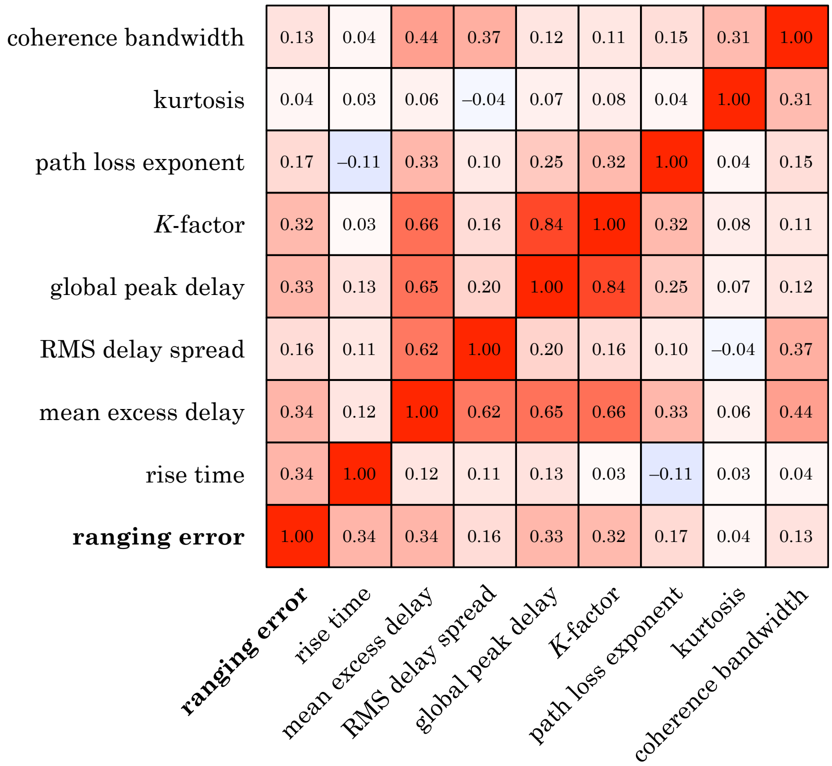


Figure 6.14: Correlation matrix between error estimates based on different features.

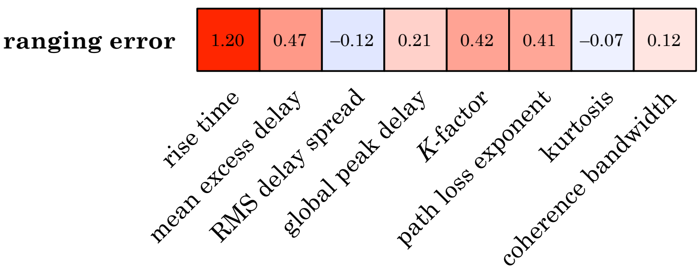


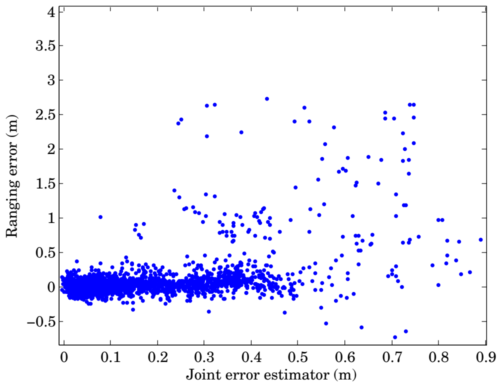
Figure 6.15: Weighting coefficients of the different error estimates, yielding optimum joint error estimation (Found by multilinear regression.)

ation of the range error in LOS conditions is 9 cm. Let us deem measurements with an error greater than $2\sigma = 18$ cm unusable for localisation (if the error would be Gaussian distributed, that means discarding 2.5% of the measurements). The first guess for the joint error estimate threshold thus is 18 cm. Let us look at the number of unduly discarded and unduly accepted range measurements as function of the joint error estimate threshold (Figure 6.17). Depending on the expected number of ODP and DDP anchors, it might be acceptable to have more false alarms than false accepts, in general. The expected number of ODP and DDP measurements depends on the application, which is yet undetermined. Therefore we take a typical case from our measurement campaign: 1 obstructed and 4 unobstructed anchors. Consequently, we set the threshold such, that on average, it a false alarm is four times as likely as a false accept: 28 cm. Using this threshold, in our example, the range measurement through the metal sheet is discarded. All other range measurements are corrected and combined (Figure 6.18c); the resulting localisation error is 4 cm. We notice that the corrected range measurements are actually over-corrected.

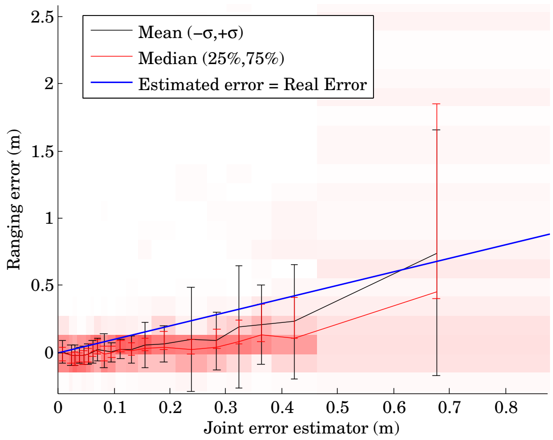
Therefore, we choose not to correct the range measurements deemed usable. So, again, the obstructed range measurement is duly discarded, but the other range measurements are used as-is; we call this strategy *discarding*. The resulting localisation error is 2 cm (Figure 6.18d).

We have now given one example with a severe obstruction and a good geometry. In this example, the ODP mitigation using the discarding strategy yields a good result. We have tried all strategies on all possible classical localisation problems, given the measurements. That is, for every of the scenes, for all target nodes, for all number of anchors (3 up to 6), for all combinations of anchors, we perform localisation; there are 2645 localisation problems in total. We expect the localisation error to depend on the geometry, so for every problem, we calculate the inverse Relative Geometric Dilution of Precision (RGDoP) according to [14, p. 8], which is a positive metric for the quality of the geometry. Histograms of the localisation error against the inverse RGDoP are given in Figure 6.19, both for accuracy and precision.

We conclude that for good geometries, the discarding strategy performs best in terms of accuracy, as also suggested by [21]. For our measurement database, the accuracy improved from 23 cm in ODP ignorant localisation to 15 cm by discarding probably-ODP measurements. The precision stays the same: 29 cm. Note that the other strategies (combined and correcting) are more precise, but less accurate.

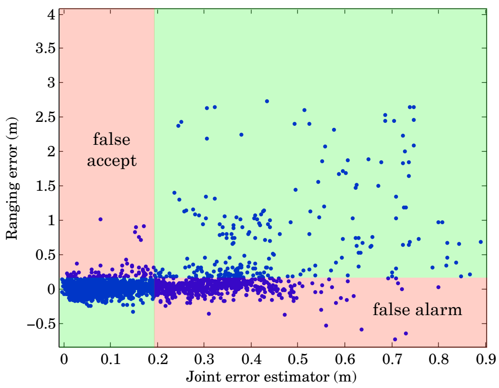


(a) Scatter plot of the real ranging error against the joint error estimate.

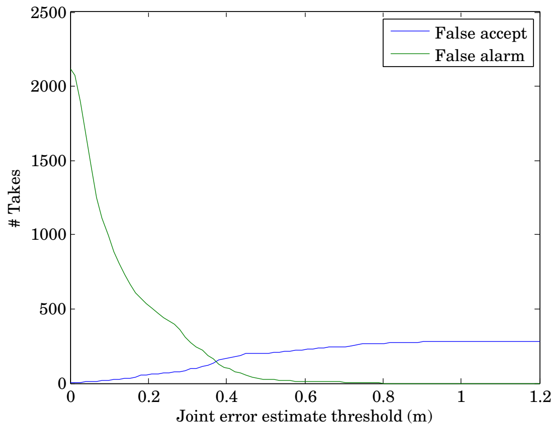


(b) Bivariate histogram of the real error against the joint error estimate.

Figure 6.16: Pathloss estimation.

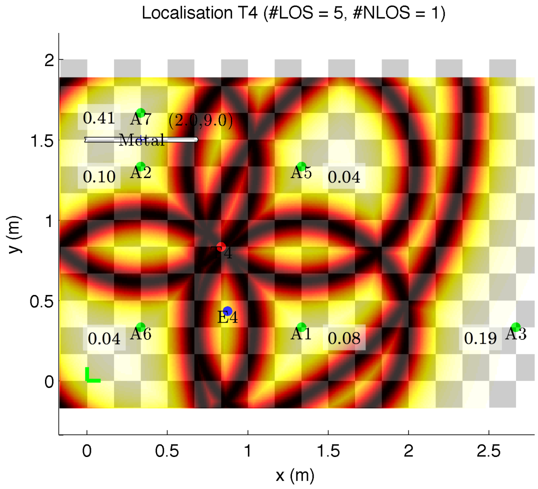


(a) False alarm and false accept regions for a threshold of 18 cm both on the real and the estimated error.

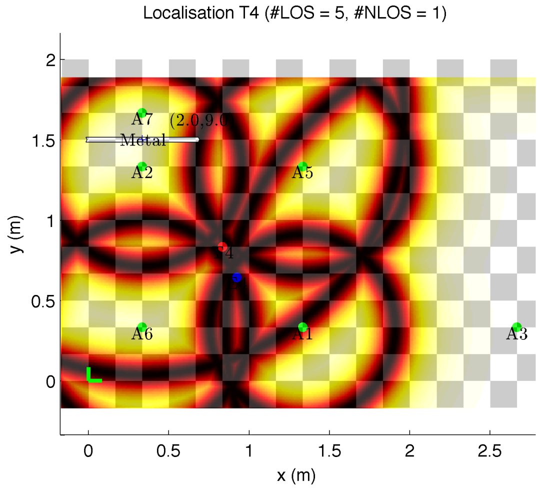


(b) Relation between the threshold on the error estimate and the number of false alarms/accepts for an 18 cm threshold on the real error.

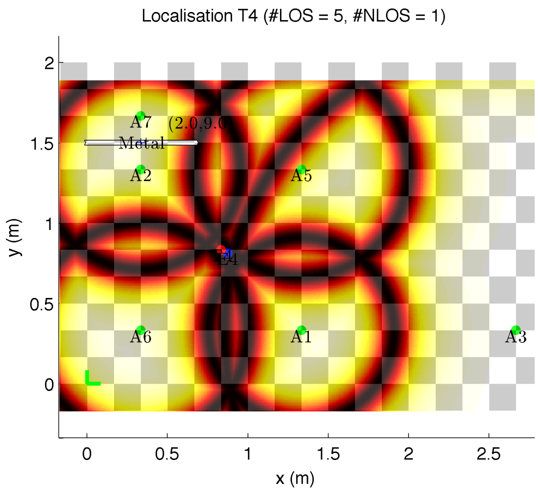
Figure 6.17: Setting a threshold to distinguish usable from unusable range measurements.



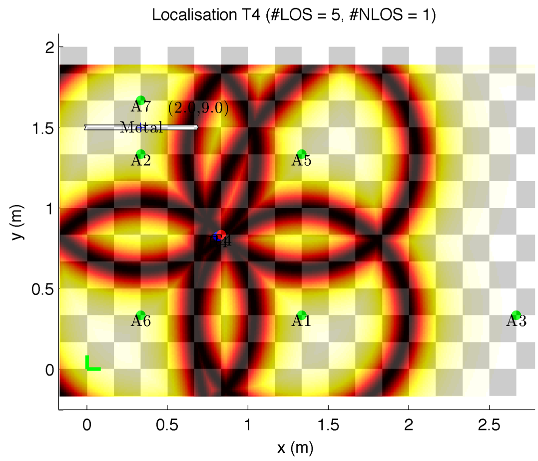
(a) *Simple*; no ODP mitigation applied; all range measurements are combined using a LMSE criterion. The joint error estimates of the range measurements from every target in meters are indicated. The localisation error is 40 cm.



(b) *Correcting*; all range measurements are corrected by subtracting the estimated range error and then combined using a LMSE criterion. The localisation error is 20 cm.

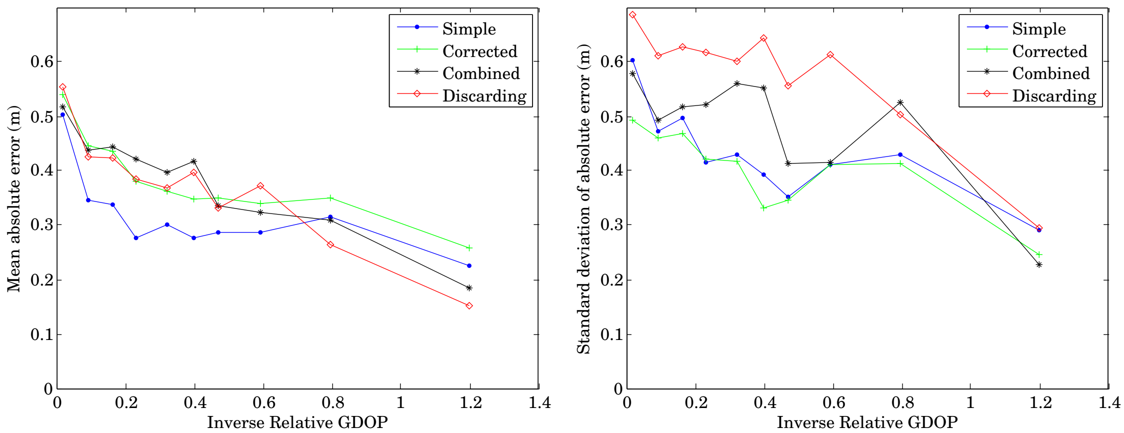


(c) *Combined*; the range measurements with a joint error estimate higher than 28 cm are discarded, the remaining ranges are corrected. The localisation error is 4 cm.



(d) *Discarding*; the range measurements with a joint error estimate higher than 28 cm are discarded, the remaining ranges are used as-is. The localisation error is 2 cm.

Figure 6.18: Example localisation using several ODP mitigation strategies. The nodes are named and colored: anchors Ai are green, the target Ti is red, the estimate Ei is blue). The inverse RGDoP of this geometry is 1.34.



(a) Accuracy as function of geometry.

(b) Precision as function of geometry.

Figure 6.19: Performance histograms of the four different localisation strategies, over all 2645 possible localisation problems. The geometry bins contain about 265 problems each; there are 10 bins.

7 Conclusions and Recommendations

We will now first answer the research question. Next, we indicate open questions encountered during this research project and propose a research agenda.

7.1 Conclusions

Let us recall the research question: *what localisation accuracy and precision can be achieved under Obstructed Direct Path (ODP) conditions, by means of (1) mitigation of the effect of ODP conditions on the individual ranging errors and (2) ODP-aware combination of the range estimates?*

A measurement campaign, representative for indoor, small scale, smart dust applications was conducted to be able to evaluate mitigation techniques suggested by literature, thereby answering the research question. Collaterally, a new improvement to leading edge detection was proposed, that improved accuracy from 8 cm to 6 cm over the line-of-sight (LOS) measurements taken in the measurement campaign.

As for question (1), we tried to estimate the ranging error of single range measurements by means of Power Delay Profile (PDP) features. By fitting a curve to the measured feature values against the real ranging error, we obtained a ranging error estimate for each feature. We used multiple features, of which the rise time τ_{rise}, mean excess delay τ_{MED}, global peak delay τ_{gd} and the Ricean K-factor performed best in predicting the actual ranging error. These and other estimators were joined by multilinear regression to obtain one error estimate. This joint error estimate turns out to predict both the value and the uncertainty in the actual ranging error (Figure 6.16b). We can subtract this joint error estimate from the range measurements to get a better range estimate. By localising with range measurements thus corrected, we do not improve accuracy, in general.

As for question (2), we can also put a threshold on the joint error estimate in order to discern usable and unusable range measurements. By discarding range measurements with a joint error estimate higher than this threshold, we can localise with the remaining range measurements. Under good geometry (inverse Geometric Dilution of Precision (GDoP) above 0.7), the localisation performance improves by this strategy; under bad geometry, the performance worsens (Figure 6.19a). Not correcting these remaining range measurements yields the best accuracy.

7.2 Recommendations

We start with recommendations that assume the overall ODP mitigation approach to be right, as generated by the measurements (first) or analysis results (second). Next, we will recommend to question the overall approach. We conclude by mentioning an idea for co-operative localisation under ODP conditions.

7.2.1 Measurements

The strange dependency of the PDP shape on the azimuth of the antennas (Figure 5.5) is not understood, and therefore ignored. However, for future measurements, we should understand the antenna pattern and the influence of the antenna feed. The antenna and/or feed should be designed to avoid deep nulls, so that the antennas need not be pointed at each other, making the localisation measurements more realistic.

Of the fixed parameters enumerated in Table 5.1 (transceivers, protocol, antennas and surroundings), the surroundings seems to be the one that could influence the results the most. That is, the conclusions may change significantly if the campaign were conducted in different surroundings. Therefore, the first recommendation is to repeat the measurement campaign in another room and/or in a reverberation chamber. The latter effectively emulates a range of surroundings, because the Ricean K-factor can be tuned by placing absorbers and by changing the antenna orientation[1].

Another fixed parameter is the height of the localisation plane. For our measurement campaign, this was 80 cm above the ground. From the few measurements we took at 20 cm, we noticed the PDPs to be quite different, probably due to ground waves. We decided to continue at 80 cm, in order to save time. However, surface waves could be a quite common phenomenon in, for example, table mount applications. Therefore, we recommend to also take measurements in a plane close to the ground. Taking threedimensional measurements could be a next step, because the pattern of scattered paths will be different, because rooms are, in general, not rotationally symmetric.

7.2.2 Analysis

A remarkable non-ideality of the measurements is the baseband response of the receiver Variable Gain Amplifier (VGA) (Figure 4.14), which positively biases metrics for temporal spread, and negatively biases kurtosis. As a result, the error estimating curves will be different compared to those of a receiver without this defect. To make the curves more generic and possibly improve the discriminatory power of the features, we would like to remove this effect. Either the receiver electronics should be improved, or the measurement results should be corrected. The latter enables using the existing measurement database. The non-ideality is known both by measurements and simulations, so it should be possible to compensate for the non-ideality.

Instead of using the (extrapolated) leading edge of the received signal, a (multi-)template matching algorithm like CLEAN could be used for first path detection [37; 38], which may lead to improved ranging performance. Furthermore, if the features were calculated based on the list of discrete multipaths thus made available, the feature performance could become more generic and less dependent on, for example, the sample rate. Alternatively, the first path could be sought with Serial Backward Search for Multiple Clusters, which performs reportedly well [11].

The fitting of feature values with an arctangent is highly disputable. The underlying problem is the lack of analytical models that predict the relation between ranging error, distance, environment, and feature value. To the author, only an inhomogeneous linear relation between τ_{RMS} and the distance [20] is known. More models should be developed and incorporated.

To evaluate the dependency of localisation strategies, the inverse Relative Geometric Dilution of Precision (RGDoP) is used as quality metric for the geometry. It is calculated according to [14, p. 8], which assumes multilateration instead of trilateration. The validity of this use should be verified.

For simplicity, all localisation problems were analysed with the same strategy, regardless of the number of anchors. However, in a problem with three anchors, of which one is obstructed, there is a great penalty in discarding this one range measurement (in fact, the problem becomes underdetermined). It makes more sense to use correcting in these low-number-of-anchors problems [21, § V.D]. It is expecting that this hybrid strategy improves the performance for low RGDoP.

Measurements that allow for co-operative localisation were taken, but were not used in lack of time. It should be verified whether co-operative localisation algorithms, using the proposed

[1] [36, Fig. 10-11] suggests that changing the antenna orientation is a suitable way of tuning the effective K-factor in reverberation chambers, next to using absorbers. This is unlike polarisation, for changing polarisation causes the K-factor to fluctuate heavily over frequency.

strategies, are able to localise around the obstacle. It would also be interesting to test the performance of distributed algorithms, as suggested by the demonstrator design of Appendix A.

7.2.3 Approach

We should also question the ODP mitigation approach on a higher level. Towards the results of this project, some assumptions and decisions have been made, such as rejecting tracking methods and setting a threshold based on an assumed false alarm/false accept ratio. In fact, these decisions should be based upon a use case. Therefore, our first recommendation is to find realistic applications where localising smart dust could be applied. More particularly, the available computing power and memory resources should be known, as well as boundaries on DDP/ODP ratios and typical geometries.

Depending on the use case thus formulated, it may turn out that ODP detection is not necessary at all. For example, if many range measurements are available of which a guaranteed minority is ODP, Least Median of Squares (LMedS) localisation may turn out to work well, without the need of ODP detection [42].

7.2.4 Outlook

The joint ranging error estimate developed in this project is disputable; its calculation is solely based on measurements carried out in a single environment. We already mentioned that for that reason, the campaign should be repeated in other environments and we expect the fitted curves to become quite different. To realise a product that performs well in all environments, co-operative nodes could communicate statistics about the environment with each other (K-factor, for example) which will make the ODP detection better adapted to the environment.

Epilogue

After the (to some extent) systematic research of the past chapters, there are some things left I deem worthy of noting. First, answers and questions outside the scope of this research project. Second, the persons I enjoyed working with and feel indebted to.

Reflections

Personally, it was a challenge to be responsible for half a year of time (nominally). In comparison to my three-month internship, I noticed I was not always as focused as I could have been. Making a planning is useful, accepting in advance that it will need revision. A three month interval might be a good time box to allow for useful playing around, while still controlling the overall progress and bearing.

When looking around in the laboratories both at IMEC and the University of Twente, I got the impression a lot of time (hence money) is wasted on searching for equipment and cables. On a more conceptual level, I share Jac's observation that a lot of researchers are *Einzelkämpfer* (lone fighters). As a result, they tend to do their work individually and experience little need for standardisation. Of course, standardisation may also restrict the space to play around and freely investigate, but I believe that this disadvantage is overrated, while the gain in standardisation is generally underestimated.

Although not as concrete as it should be, I believe smart dust will have many useful applications that may contribute to the quality of our lives. At the same time, more dystopian applications, such as pervasive government surveillance, can also be envisaged [61]. Freedom of information legislations and, more fundamentally, always coupling personal responsibility to the power generated by knowledge, might prevent these applications [62; 63].

In general, the awareness of the risks inherent to smart dust technology is too low. As many people work on the technology, every person realises only a small piece of technology. Assuming the existence of free will, every person bears a small piece of responsibility for the final consequences. The *communis opinio* seems to be that this infinitesimal responsibility can be rounded to zero. This numerical imprecision does not obey conservation of responsibility and is, therefore, misleadingly false.

The ignorance of possible health effects of the radiation present in the laboratories is another concern. We might trust that national and world-wide regulations are based on health effect studies. On the other hand, the field strengths occurring in the laboratories sometimes exceed the governmental limitations (in this project, by 36 dB, see Section 5.2.4). Therefore, I argue that a literature study on the long term health effects of Ultra-Wideband (UWB) power should be done and applied to the laboratory practice.

Acknowledgements

It was a good experience to have you, Yakup, as my daily supervisor. It started out in our joint problem definition, but you kept encouraging me until the end and feeding me with Bastogne. Jac, I was constantly surprised by your clearness of mind and mild attitude. Dries, your peace and subtle humour was therapeutic, as well as your personal involvement. I already knew you, Arjan, but I was not prepared for the supervision that ranged from accurately finding typos to personal counseling. I am not sure how I would have finished the project without. Thanks, Mark, for your trust in me, especially during the last weeks when a lot of work still needed to be done.

I also learned a lot from other group members. Dilpreet, you taught me the meaning of being humble, by example. Hans, your positive attitude gave me serious hope. Nauman, I enjoyed your daily attention and I loved to discuss about quite a range of topics with you. Sundeep and Shrisail, I think we are overtime competitors, which created a relationship. Thanks, Kathleen, for trusting me with your Nissan ‘rijdt niet weg’ Almera (actually, it did, perfectly). I enjoyed getting to know Jacqueline, XiaoYan, Ruben, Maya, Simonetta, Barend, Kezheng, Lindsay, Ben, Alex, Frank, Coby, Guido, Julien, Fabien and Joonseong, and sharing some life events.

Thanks to my housemate in Enschede, Stefan, for the therapeutic sessions, your humour and plain brilliance. Octo, you are a skillful teacher and a versatile craftsman; I loved listening to your adventures. It was a pleasant surprise to get to know Matthijs, Joeri, John, Chrisje, Jasper and Ard, being my housemates in Eindhoven.

I’m not sure how I would have finished the writing effort without the logical and emotional support of my parents Martin en Cobi. You have been particularly faithful, in fact already quite some years. Thanks Sifra for your seemingly random phone calls, thanks Norbert for literally sharing your enthusiasm with me.

Finally, I would like to thank my Designer for giving me the opportunity to explore a universe that can be trusted to be consistent. I was only to able to finish this report because He knows the meaning of struggle, first-hand.

A Smart Dust Demonstrator

This appendix documents the design of a demonstrator, that shows typical features of smart dust sensor networks. The design also serves to direct research and development activities, by setting a goal that is a firm step forward, but still within reach.

We start by requiring what the group of smart dust nodes should be able to do. Then, we start designing and picking Commercial Of The Shelf (COTS) components as much as possible. For the component that is not yet commercially available, we first select the technology most suited for precision localisation. Then we search for research projects on this technology that are as close as possible to the localisation specifications. We conclude by summarising the design of the smart dust nodes and by making explicit what step forward the realisation of this demonstrator would be.

A.1 Requirements

We intend to design a demonstrator that shows typical features of envisioned smart dust, while it is still possible to realise within a few years. Typically, smart dust sensor networks can sense and/or actuate some quantity as function of position. The networks consist of many nodes, thereby realising dependability [64; 5]. To add position awareness, they co-operatively localise themselves [4; 17]. The nodes can save energy by communicating just with their neighbours, that can relay a message to its eventual recipient.

The demonstrator will consist of 200 small nodes, shaped as 2×2 cm tiles. One side of every node is fully covered by a full-colour, low-resolution display. At the other side, there is an optical sensor. The nodes shall be able to localise themselves with 1 cm mean absolute error. The nodes shall be able to communicate amongst each other within a range of 1 m at tens of kilobytes per second. Finally, they shall last at least one year without maintenance.

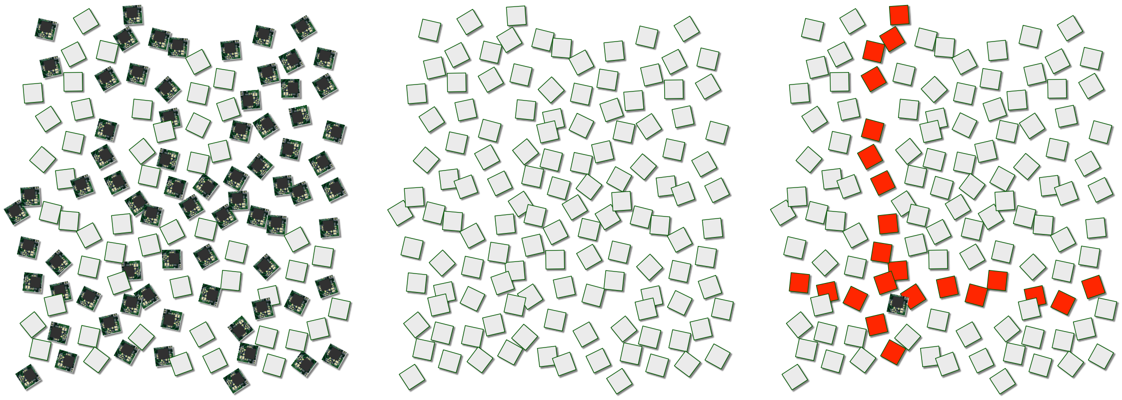
This demonstrator could demonstrate features of smart dust networks as follows:

Dependability/redundancy Throw a bucket with 200 nodes out over the table. Typically, 100 nodes are disqualified for display purposes, because they landed face-down, see Figure A.1a. Fortunately, that does not matter, because there are enough nodes that landed face-up.

Co-operative localisation Turn all nodes face-up (Figure A.1b) and then flip one node fade-down again. Using the optical sensor, the node will notice it is flipped up. Using co-operative localisation, the nodes will decide on a crosshair that is displayed around the flipped node, see Figure A.1c. Move a node into or out of the crosshair to see that it corrects it colour within time. Or shuffle all nodes and see that the crosshair is restored after time.

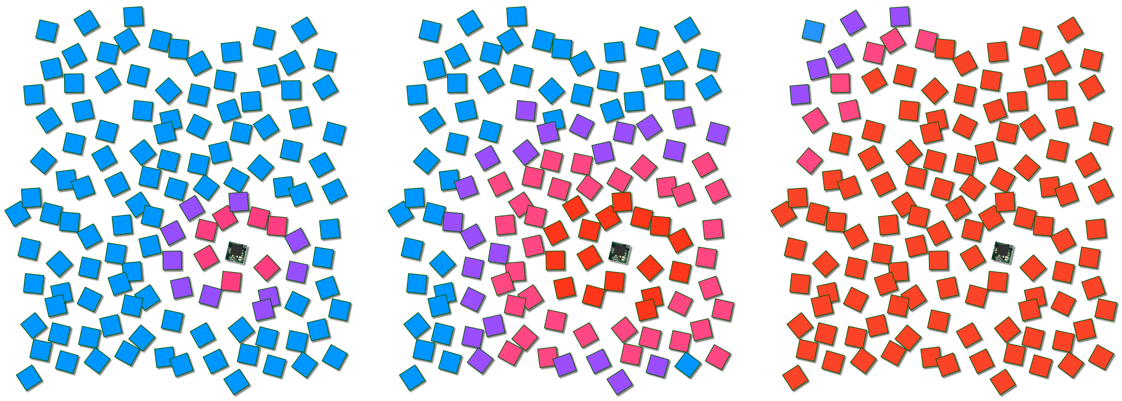
Short range mesh Flip one of the face-up nodes. This node will then use just enough power to inform its nearest neighbours, that then change colour. Again, these nodes inform their neighbours, so that the colour change spreads out like a oil spill, see Figure A.2.

Sensor network Distribute the sensors face-up over a low-resolution printed colour picture on the table. Using localisation and their optical sensors, the nodes will store the picture in a distributed fashion. As soon as a node is localised and the colour is stored, the node shows its underlying colour, see Figure A.3a. One can put the nodes in the bucket again and randomly spread them out elsewhere, see Figure A.3b. After a while, the shuffled nodes will reproduce the picture using their joint memory, see Figure A.3c.



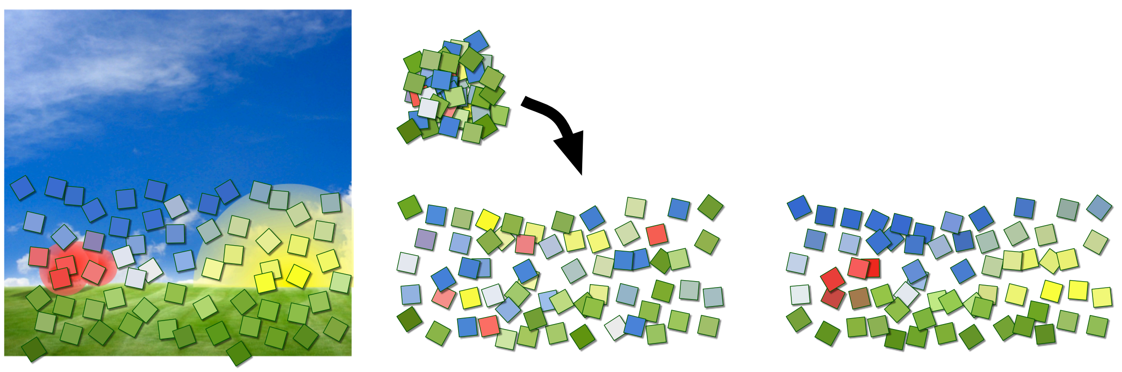
(a) First throw of smart dust nodes on the table.
(b) All nodes flipped to face-up.
(c) Flip one node to localise.

Figure A.1: Smart dust demonstrator; nodes on a table.



(a) Flip one node to start the spill...
(b) ...that propagates ...
(c) ...and propagates.

Figure A.2: Smart dust demonstrator; oil spill demonstration.



(a) Spread the nodes over a low-resolution picture to train the nodes.
(b) Gather the nodes and spread them out again.
(c) After a while, the original picture is replicated.

Figure A.3: Smart dust demonstrator; picture replication demonstration.

A.2 Start Designing Using Commercially Available Components

The smart dust nodes shall have a structure as outlined in Figure A.4. To create this demonstrator, we try to find COTS components that comply with their specifications.

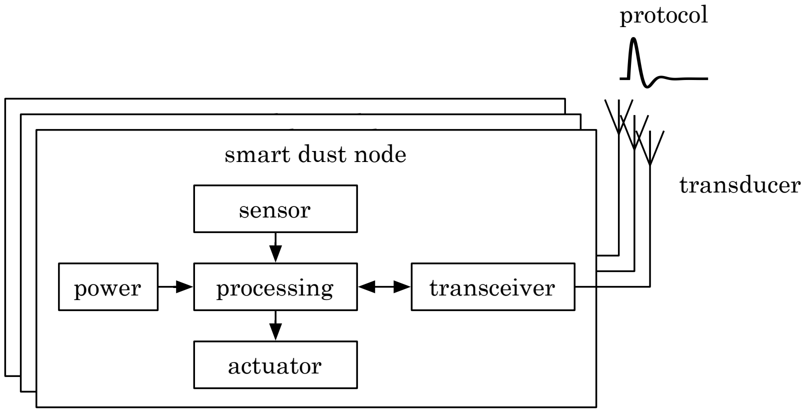


Figure A.4: Break-down of a smart dust system.

Each node will consist of an actuator, a sensor, a processor, a tranceiver, a transducer (e.g. an antenna), and a power supply, see Figure A.4. The actuator is a full-colour display. Monochrome bistable Liquid Crystal Displays (LCDs) are available, that only need $4\,\mu$J per update [65]. We suppose that colour displays of this type will have energy consumption of the same order of magnitude. The sensor is a Red Green Blue (RGB) photosensitive cell. Duty-cycling $200\,\mu$W MIPS^{-1} microcontrollers are available at low cost [66]. The demonstrator can be powered by a IEC PR44 zinc-air cell, which contains 3600 J of energy. We conclude that, apart from transducer and tranceiver, all components needed for the demonstrator are commercially available.

A.3 Localisation Technique

The choice of the transducer and transceiver depends on the medium that we will use. In the demonstrations, the nodes will be close together; they may not always see each other, but should be able to hear each other acoustically. Although there are promising results using acoustic localisation [67], we would like the technology to be able to penetrate walls [9]. Therefore, we choose radio communication.

There are quite some techniques for radio-based localisation. Recall that localisation is the estimation of a node's position in Euclidian space.[1] The quantifiable entities that can be derived from Euclid's axioms are *position*, *distance* and *angle* [68]. As we want to know position, but have no direct way of measuring it, we need to measure distances and/or angles. Distances cannot be measured directly, so we can use the two properties of propagating plane waves that depend on the distance: power and delay. All known localisation techniques, such as Received Signal Strength (RSS), Time of Arrival (ToA), Time Difference of Arrival (TDoA), Return Time of Arrival (RToA), Angle of Arrival (AoA) [10, ch. 8], can be classified in this scheme. See also Figure A.5.

What localisation technique(s) would be appropriate for our demonstrator? Angle-based measurements involve either moving antennas or antenna arrays[2] which involve complex electronics and increase size. As the smart dust nodes need to be small and simple, we will not measure AoA. Measuring distance by means of power has the advantage of simplicity; almost all COTS

[1]As opposed to curved spaces, most notably as conceived by Einstein.

[2]One could argue that antenna arrays fundamentally measure delays and combine these delays to obtain an angle of arrival.

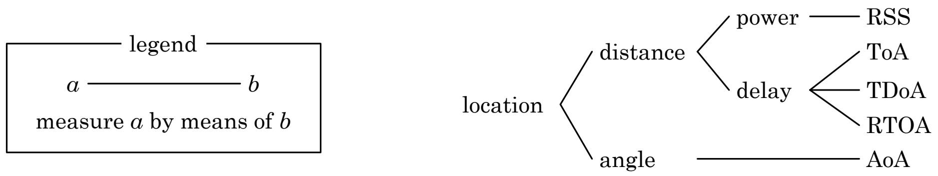


Figure A.5: Classification of localisation techniques, based on Euclid's axioms and the observable properties of plane waves.

transceivers already have some Received Signal Strength (RSS) Indication (RSSI) that can be used for this purpose. The disadvantage of this measurement technique is that a path loss exponent equal to 2 is assumed, which is especially questionable for indoor environments. According to [69], in indoor environments, time-based localisation outperforms power-based localisation if a direct radio path is present, but as soon as the direct path goes undetected, power-based measurements perform better. We choose for time-based localisation because of its high potential accuracy, keeping in mind the realistic possibility of Undetected Direct Path (UDP) channel conditions. What specific technique (ToA, TDoA or RToA) we will depend on system-level design decisions.

What frequency and bandwidth is necessary to obtain 1 cm localisation accuracy? Let us assume that localisation and ranging accuracy are in the same order of magnitude, given reasonable node geometries. There is a fundamental Cramér-Rao lower bound on ranging precision [70]. Although this bound is only asymptotically true for high Signal to Noise Ratio (SNR) [71], it indicates there is a relation between the achievable precision and the used bandwidth:

$$\sigma_{\hat{\tau}} = \frac{1}{2W\sqrt{SNR}} \tag{A.1}$$

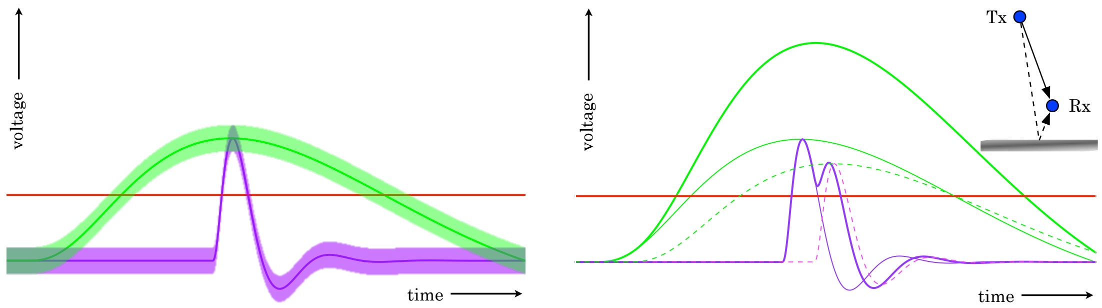
$$W = \sqrt{\frac{\int_{-\infty}^{\infty} \omega^2 |P(\omega)|^2 \,\mathrm{d}\omega}{\int_{-\infty}^{\infty} |P(\omega)|^2 \,\mathrm{d}\omega}}, \tag{A.2}$$

where $\sigma_{\hat{\tau}}$ is the standard deviation of the estimated time-of-flight, W is the RMS bandwidth of the transmitted pulse, SNR describes the Additive White Gaussian Noise (AWGN) channel and $|P(\omega)|^2$ is the energy spectral density of the baseband pulse. For a given noise floor, we can see that choosing a high bandwidth decreases the error. This relation is also intuitive: we want to measure with low uncertainty in time, so the phenomenon must be concentrated in time. Consquently, the phenomenon must be spread out in frequency (Fourier's uncertainty principle[3], [72]). Practically, we see that time detection of wideband pulses is more resistant to additive noise, see Figure A.6a. Although not modeled by this Cramér-Rao lower bound, the detection of wideband pulses is also more resistant to multipath components, see Figure A.6b[4]. For these two reasons, ultra-wideband time-based ranging is widely regarded as the technology of choice for precision ranging [70]. A 20 cm standard deviation is realistic for indoor ranging, using 3.10-5.15 GHz at 0 dB SNR [73; 74]. This accuracy is still one order of magnitude larger than what is needed for the demonstrator, so we might need to improve on that. We choose to look for an Ultra-Wideband (UWB) transceiver and compatible antenna.

A.4 Transceiver Availability

Obeying the demonstrator principle, we start by looking for commercially available UWB transceivers. These could be complete solutions, chipsets or chips. One of the few complete commercial localisation products at the time of research (January 2010) is Time Domain's

[3]...of which the Heisenberg uncertainty principle is a well-known consequence.
[4]The multipath induced time uncertainty is described by a more elaborate Cramér-Rao lower bound [??].



(a) Additive noise causes uncertainty in the detected time of arrival. Due to the lower slope of the narrowband signal, this uncertainty is higher for narrowband pulses than for wideband pulses.

(b) A multipath component (dashed curve) one multipath component causes the summed signal (thick curve) of the narrowband signal to be detected too early, while the detection of the wideband signal is unaffected.

Figure A.6: Received time-domain pulses for ranging. The purple pulse is wideband, the green pulse is narrowband. The crossing of the received signal with the red threshold is the detected time of arrival.

PulsON 400. It claims to have 7 cm ranging resolution and 354 m range, and in general, Time Domain's products are applied successfully [24]. However, the size of 7 × 10 cm disqualifies it for a smart dust demonstrator. Furthermore, it is a complete but closed solution. The Ubisense Tag Module is smaller (4 × 4 cm, TDoA/AoA, 15 cm accuracy), but also is a closed solution. That means that it will be hard to understand and improve upon. Therefore we discard this option.

Looking for a chipset or chip, we note that many commercial UWB chipset vendors are united in the WiMedia Alliance. To promote interoperability, WiMedia stimulates her members to produce UWB PHYs that have a ECMA-369 compliant MAC-PHY interface. ECMA-369 specifies an optional ranging interface [75, sec. 7], which suggests that WiMedia members might have produced chip(set)s with ranging functionality. Therefore all members of the WiMedia Alliance that seemed to produce UWB chipsets, were contacted during January 2010 to find out whether or not they implemented ranging, see Table A.1. If they reacted, we also tried to understand why it was not possible and not even with some work. Unfortunately, no commercial chip or chipset was available, because ranging was seen as a niche, as far as we know.

Finally, we start to look for solutions in research institutes, with a preference for standard-based solutions. At the time of research there were four published projects on indoor localisation using UWB, see Table A.2. Of these, only IMEC was suggesting collaboration. IMEC realised an integrated, Ultra Low Power (ULP) IEEE 802.15.4a-compliant transmitter and the analogue part of the receiver [77; 46]. The receiver has a peak power consumption of 10 mW, which corresponds to 78 μW at the lowest duty cycle [45]. The transmitter consumes at least 650 μW, but only when it is on. The analogue (baseband) output of the receiver is sampled by an oscilloscope and processed offline using MATLAB. The measurement is also synchronised to the transmitter, so effectively, ToA ranging can be performed. In line-of-sight (LOS) conditions, an accuracy of 9 cm mean absolute error is achieved (Figure 6.1b). Their demonstrator also consists of compact and cheap antennas (2 × 2.5 cm, on-PCB) [50]. IMEC plans to integrate transmitter, receiver and processing, to finally obtain a single chip that could function as a radio for smart dust nodes. Therefore, we decide to continue with IMEC and see what next steps should be taken there to come closer to the proposed demonstrator.

A.5 Conclusion

The design for a smart dust demonstrator was made. Each node measures 2.5 × 2.5 cm, consists of a zinc-air cell battery, a photo cell and passive full-colour LCD. Communication and

Table A.1: Reaction of WiMedia Alliance members [76] on our request for ranging-capable devices.

contacted & reacted	products	comments
NIT of Tiruchirappalli	OFDM-MB	It will take another year for completion.
Alereon, Inc. (form. Intel)	AL5100/AL53xx	No ranging support, might be possible when you write 'deep' code, but no time for support.
Wisair Ltd.	WSR601	No ranging support, does not know other vendors who do support.
Pulse LINK, Inc	PL3100 CWave	No ranging support.
contacted, no reaction	**products**	**comments**
Wionics (was RealTek)	RTU7010/7012	
Staccato (Artimi)	SC4501	
Sigma Design (form. Blue7)	Windeo	
ITI Techmedia	–	Commercialisation institute.
ASTRI	–	Research institute.
not contacted	**products**	**comments**
Focus Enhancements, Inc.	TT-1013	No publications anymore.
Samsung Electronics Co., Ltd.	S3CR650X	No publications anymore.
NXP (form. Philips)	ISP3582 (MAC)	Only MACs, use RealTek PHYs.
NEC Electronics	–	Only MACs, use RealTek PHYs.
STEricsson (STMicroelectronics)	–	No mention of UWB or WUSB on their site.
Infineon	–	Only LNAs.
Synopsys	–	Not a vendor of discrete ICs, only IP.
Nokia	–	Not a vendor of discrete ICs.
Olympus	–	Not a vendor of discrete ICs.
Texas Instruments, Inc.	–	Left WiMedia alliance in May 2008.
TZero Technologies	–	Quit.
WiQuest Communications	–	Quit.

Table A.2: Published indoor UWB localisation projects in January 2010.

project	partners	status
Europcom [78; 56]	TU Delft, Thales UK, IMST, TUG	Realised a co-operative localisation unit with an RMS error of 25 cm in indoor environments. No reaction to e-mail.
Pulsers2, WP3b	IMST, IHP, Sennheiser, EADS	A European Ultra-WideBand (EUWB) project: transmitter/receiver chipset in experimental phase. Expect to have an integrated module by February 2011.
UCAN [79]	GWT Forschung	Predecessor of Pulsers2 in EUWB context.
WPAN UWB [77; 46; 45]	IMEC	A bits-in bits-out demonstrator platform is realised, there is a working ranging demonstrator with offline processing.

ranging is performed using UWB radio signals with a tranceiver in development by IMEC. This transceiver can currently perform ToA ranging with 9 cm accuracy, but there are plans to develop TDoA capabilities in the future. The transmitter uses 650 μW when it is on. An on-PCB elliptical pseudo-monopole antenna is used.

B Transmitter Output Power

In this appendix, measurements on the transmitter's output power are presented and related to the regulatory limits.

The goal of these measurements is to determine the power at the balun output of the transmitter. Two methods (time-domain and frequency domain) are compared to be sure about the results.

B.1 Measurement Set-up

The Ultra-Wideband (UWB) low-band Transmitter (Tx) V1 module was set-up to repeatedly transmit a payload containing all zero bytes in 2.2 mode. That is, the digital controller is programmed with the register setting of `mode-2.2sjoerd.csv`, with the maximum `AMP` parameter (SPI register 11[4:2]) of 7.

The balun output is connected to the measurement device using a 1 m Radiall cable and a BLK-89-S+ DC-block. The insertion loss of the cable is only specified to be 1.81 dB maximum at 3 GHz [80]. Therefore, we use the average of our measurements on this type of cable for 4 GHz to 5 GHz, that is 2.10 dB (Figure B.3). The insertion loss of the DC-block is specified to be 0.35 ± 0.10 dB for the 4-5 GHz range [81]. The insertion loss thus totals 2.45 dB.

B.2 Time-domain Results

The first measurement is taken in the time domain, using the LeCroy SDA816 Zi 40 GS s^{-1} oscilloscope. An isolated pulse from the pre-amble and a burst of four positive pulses from the payload were both captured and presented in Figure 4.16, which is repeated here in Figure B.1.

We read the RF output amplitude of the current transmitter (without Power Amplifier (PA)) to be 62 mV$_{\mathrm{pk}}$ at the oscilloscope, for four subsequent 2 ns pulses (Figure B.1). We calculate the pulse amplitude of the RF signal at the balun output to be:

$$v_{\mathrm{pk,balun\ out}} = v_{\mathrm{pk,scope}} \cdot 10^{\left(\frac{\mathrm{IL}}{20\,\mathrm{dB}}\right)} = 62\,\mathrm{mV_{pk}} \cdot 10^{\left(\frac{2.45\,\mathrm{dB}}{20\,\mathrm{dB}}\right)} = 82\,\mathrm{mV_{pk}}. \tag{B.1}$$

According to [49, Table II], the maximum pulse amplitude of a 2 ns pulse in 802.15.4a mode 2.2 is 575 mV$_{\mathrm{pk}}$ in a 50 Ω load, assuming a 0 dBi antenna. The used circular pseudo-monopole microstrip antenna has a reported azimuth gain of 3.02 dBi to 3.63 dBi for 4 to 5 GHz [50, Table 2]. If we use the average (3.33 dBi) for our calculations, the maximum signal amplitude at the antenna feed is:

$$v_{\mathrm{pk,antenna\ in}} = v_{\mathrm{pk,0\ dBi}} \cdot 10^{\left(\frac{-A_{\mathrm{antenna}}}{20\,\mathrm{dBi}}\right)} = 575\,\mathrm{mV_{pk}} \cdot 10^{\left(\frac{-3.33\,\mathrm{dBi}}{20\,\mathrm{dBi}}\right)} = 392\,\mathrm{mV_{pk}}. \tag{B.2}$$

The gain of the internal PA can therefore be at maximum:

$$A_{\mathrm{PA}} = 20\log_{10}\left(\frac{v_{\mathrm{pk,antenna\ in}}}{v_{\mathrm{pk,balun\ out}}}\right) = 20\log_{10}\left(\frac{392\,\mathrm{mV_{pk}}}{82\,\mathrm{mV_{pk}}}\right) = 13.6\,\mathrm{dB}, \tag{B.3}$$

which should equal the measurement gain.

B.3 Frequency-domain Results

The second measurement is taken in the frequency domain, using an Agilent E4440A spectrum analyser. The UWB compliance measurement procedure common at IMEC was used [82]. The spectrum analyser readings are plotted in Figure B.2.

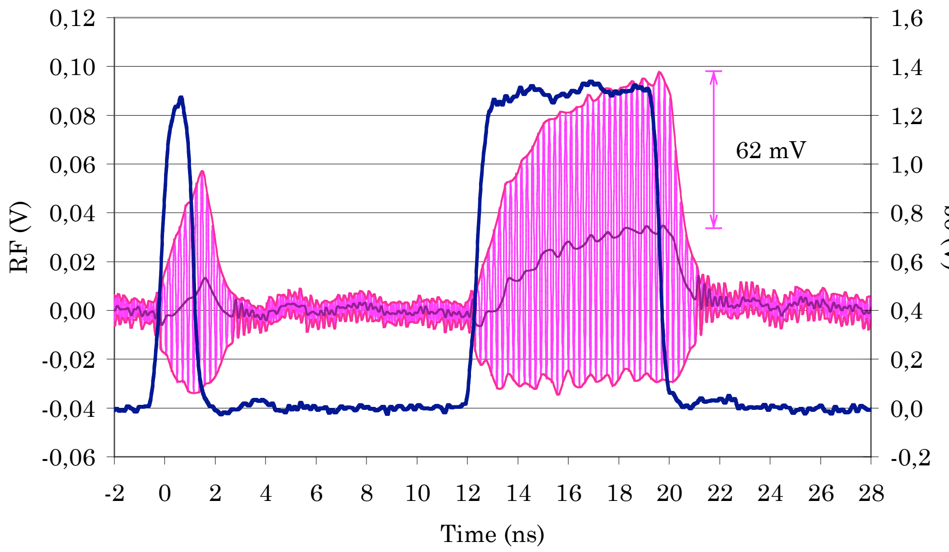


Figure B.1: Transmitter RF output for one and for four positive pulses. The single pulse is the first from the preamble, the burst of four positive pulses is taken from the payload (there is a hiatus in the time scale). Measured with a LeCroy SDA816 Zi 40 GS s^{-1} oscilloscope, connected to the balun's differential output with a 1 m Radiall R284 cable and a Mini-Circuits BLK-89-S+ DC-block. A baseline is added at the middle between the upper and the lower envelope.

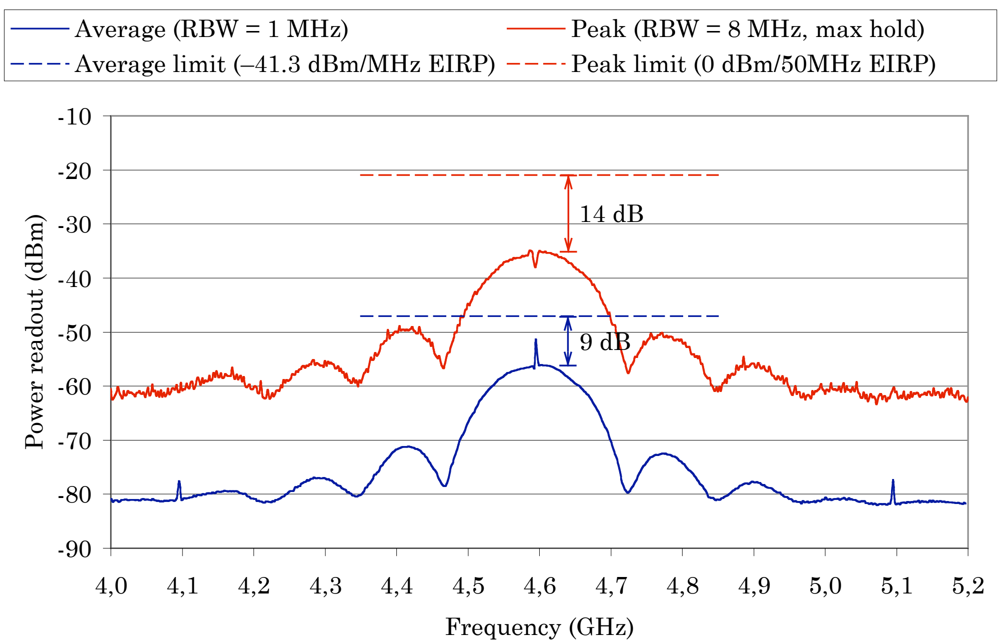


Figure B.2: Readings of the Agilent E4440E spectrum analyser, fed with the balun's output through a 1 m Radiall cable and a BLK-89-S+ DC-block. The limits are compensated for the insertion loss, antenna and spectrum analyser's RBW.

The Federal Communications Commission (FCC) specifies that the UWB Equivalent Isotropically Radiated Power (EIRP) does not exceed 0 dBm/50 MHz by peak nor −41.3 dBm/MHz on average. The average power limit translates to the spectrum analyser's readings using only the insertion losses and antenna gain:

$$P_{\text{lim avg,readout @RBW = 1 MHz}} = -41.3 - IL - A_{\text{antenna}} = -41.3 - 2.45 - 3.33 = -47.1\,\text{dBm}. \tag{B.4}$$

As for the peak power limit, some compensation is necessary: we measure using a RBW of 8 MHz, whereas the limit is specified for 50 MHz. A linear compensation term is used (8/50) plus a term to compensate for the saturation of the band pass filter used in the spectrum analyser (β^8_{50}, [83]). For the 2.2 mode, this amounts to:

$$\begin{aligned} P_{\text{lim pk,readout @RBW = 8 MHz}} &= 0\,\text{dBm} + 20\log_{10}\left(\frac{RBW}{50\,\text{MHz}}\right) + \beta^8_{10} - IL - A_{\text{antenna}} \\ &= 0\,\text{dBm} + 20\log_{10}\left(\frac{8}{50\,\text{MHz}}\right) + 0.75\,\text{dB} - 2.45\,\text{dB} - 3.33\,\text{dB} \\ &= -20.9\,\text{dBm}. \end{aligned} \tag{B.5}$$

These limits are added to the plot of Figure B.2 and the margins of the signal power with respect to these limits are indicated. Note that the spectral peak in the average curve at the Digitally Controlled Oscillator (DCO) frequency is ignored. The peak is caused by mixer leakage that we expect to be fixed in the next revision of the transmitter. Therefore we deem it not to be representative for the transmitter design.

The smallest margin occurs for the average power limit, so the signal seems to be effectively average power-limited. This is not in accordance with our expectation: mode 2.2 should be peak power-limited. On the other hand, the data (all zero bytes) is not very representative, which means that the measured spectrum may not be representative as well.

B.4 Conclusions

The time-domain measurement seems to indicate that the transmitter's PA may amplify by 13.6 dB for the radiation of a 3.33 dBi antenna to be exactly legal.

The frequency-domain measurement gives an allowed amplification of 14 dB, assuming the signal to be peak power-limited. Note, however, that the average power limit only allows an amplification of 9 dB. We do not completely trust the frequency-domain measurement.

We deem the time-domain measurement to be the most trustworthy method, so we conclude there is an amplification margin of 13.6 dB, with respect to the current output level of the balun.

B.5 Extra: Cabling

To find out what cabling is acceptable, various cables at hand in IMEC's laboratory were examined with a Rohde & Schwartz ZVL network analyser. The results are plotted in Figure B.3; the losses vary between 0.28 and 1.6 dB/m.

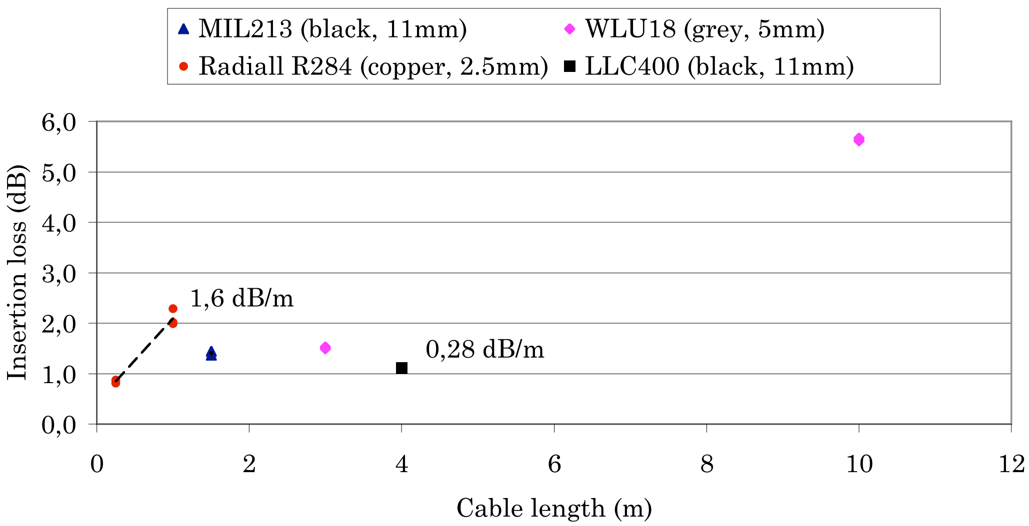


Figure B.3: Insertion loss for 4-5 GHz of 12 cables at hand in IMEC's laboratory. The value was found by averaging the minimum and maximum values of the S_{21} parameter in a 4-5 GHz sweep. Cables with N-connectors are connected without calibration, cables with SMA-connectors are connected with an N-connector cable and SMA-adapters, for which is calibrated or compensated.

C SPI Improvements

This appendix describes the modifications made to the USB-SPI interface firmware, as well as the created MATLAB driver.

C.1 Firmware

Below, the main file of the new USB-SPI interface firmware is printed; sections in red roughly represent the added code. For this public release, most existing code is replaced by an ellipsis (...).

```
// ***************************************************************************
//            IMEC-NL CONFIDENTIAL PROPRIETARY UNPUBLISHED SOURCE CODE
//            Copyright 2007, Stichting IMEC Nederland
//            ALL RIGHTS RESERVED
// ***************************************************************************
//----------------------------------------------------------------------------
//
// MODULE NAME:    usbrx.c
//
// MODULE FUNCTION:
//   USB-to-SPI converter
//
// MODULE AUTHOR(S) :
//   Jef van de Molengraft (jef.vandemolengraft@imec-nl.nl)
//
// ...
//
// History :
//   25-02-2008  Initial version
//   22-06-2010  Added hardware flowcontrol (Sjoerd Op 't Land)
//----------------------------------------------------------------------------

...

//***************************************************************************
// CTS defines (negative logic)
#define CTS_TRUE    P2OUT &= ~0x04
#define CTS_FALSE   P2OUT |=  0x04


...

#pragma vector=UART1RX_VECTOR
__interrupt void usart1_rx(void)
{
  // Short interrupt handler, just register the incoming byte and return
  rxWritePointer += 1;
  rxWritePointer &= 0x07; // wrap around ring buffer
  rxBuffer[rxWritePointer] = RXBUF1;
}

//***************************************************************************
void InitHardware()
{
  ...
  // CTS Support
  P2DIR|=0x04; // enable CTS output...
  P2SEL&=~0x04;
  CTS_TRUE; // ... and drive low (negative logic)

```

```
  ...

}

//*****************************************************************************

int main(void)
{
  InitHardware();
  while(1){
    // Main event loop processes the bytes in the ring buffer
    // Note: this loop may be interrupted by incoming bytes, so either (1) this
    // loop should be interruptible, or (2) we should trust that setting CTS
    // to false avoids the loop being interrupted.
    if (rxReadPointer == rxWritePointer)
    {
      // last byte received is already processed
      CTS_TRUE; // welcome new bytes
      // idle...
    } else {
      // at least one more byte needs to be processed
      CTS_FALSE; // ask the sender to stop

      unsigned char b; // to be eliminated
      unsigned      i; // for iterating through strings

      rxReadPointer += 1;
      rxReadPointer &= 0x07; // wrap around ring buffer

      b = rxBuffer[rxReadPointer];

      ...

      }

    } // else(rxReadPointer == rxWritePointer)

  } // while(1)
} // main
```

C.2 Driver

The Graphical User Interface (GUI) made in Visual C++ formerly used to program the modules, was ported to MATLAB. A class `SpiInterface` was made to be the agent of the USB-SPI interface, regardless of the actual firmware. Using this firmware, agents of the modules could be made, of which two examples are included: `ImecModule` and `ImecUwbTxModule`.

C.2.1 SpiInterface

```
classdef SpiInterface < handle
    %> API for IMEC's SPI interface via serial-over-USB.
    %>
    %> Constructing this object opens the serial port and destructing it
    %> closes the port again. It is therefore important to take care that
    %> the object is properly delete()'d, also when something goes wrong
    %> between construction and destruction.
    %>
    %> @author Sjoerd Op 't Land (maintainer)
    %> @version 1.0
    %> @since 2010-06-16
    %> @todo Add readback support to verify succesful writing.

```

```
    properties % note that SpiInterface is a subclass of handle, so mutable
        serialPort % the serial port the SPI interface is connected to
        writeBuffer % to buffer write data
        spiIdentification % name and version of the SPI interface firmware
        hardwareFlowControl % whether we use hardware flow control
    end % properties

    properties (Constant)
        baudRate = 1000000
    end

    methods
        %% Opening and closing of the device
        function obj = SpiInterface(varargin)
            %> Construct an SpiInterface agent to the SPI interface at a
            %> (virtual) serial port. Whether or not we use hardware flow
            %> control is automatically detected.
            %>
            %> @param portName Name of the serial port (e.g. 'COM5'). If
            %>        omitted, we try to automatically find an interface.
            if numel(varargin) >= 1
                portName = varargin{1};
            else
                portName = '';
            end

            %% Clean up formerly created or even opened serial port objects
            % Find all serial ports objects created in the MATLAB workspace.
            % If a portName is given, we will only (close and) clean up serial port
            % objects for that port.
            if isempty(portName)
                foundPorts = instrfind('Type','serial');
            else
                foundPorts = instrfind('Type','serial','Port',portName);
            end
            for foundPort = [foundPorts]
                if strcmp(foundPort.Status,'closed')
                    SpiInterface.logInfo(['Cleaning up ' foundPort.Name '.']);
                else
                    fclose(foundPort); % close the port
                    SpiInterface.logInfo(['Closed and cleaning up ' foundPort.Name '.']);
                end
                delete(foundPort); % clean up the serial port object
            end

            %% Automatically try and detect an SPI interface
            % A port is 'Available' when it is not in use by MATLAB. When it's in use
            % by another application, it will show up as 'Available', but fail to open.
            serialInfo = instrhwinfo('serial');
            availablePorts = serialInfo.AvailableSerialPorts;
            if isempty(portName)
                multiplePorts = true;  % for adequate error messages
            else
                % check that this portName is actually available
                if ismember(portName,availablePorts),
                    availablePorts = {portName};
                    multiplePorts = false; % for adequate error messages
                else
                    error([portName ' does not exist.']);
```

```
            end
        end


        remainingAvailablePorts = numel(availablePorts);

        for tryPortNumber = 1:numel(availablePorts)
            tryPort = availablePorts{tryPortNumber};
            SpiInterface.logInfo(['Trying␣to␣find␣interface␣at␣' tryPort ' ...']);

            remainingAvailablePorts = remainingAvailablePorts - 1;

            obj.serialPort = serial(tryPort,'BaudRate',obj.baudRate,'DataBits',8,'FlowControl','hardware');
            try
                % try to open the port, may fail if it is opened by
                % another application
                fopen(obj.serialPort);
            catch saveError
                delete(obj.serialPort);

                if remainingAvailablePorts == 0; % last or only port
                    if multiplePorts,
                        SpiInterface.logInfo(['...␣failed␣to␣open␣(' ...
                                              saveError.message ...
                                              '),␣last␣port,␣so␣throw␣error.']);
                    end
                    rethrow(saveError); % last port, failed
                else % by definition multiple ports
                    SpiInterface.logInfo(['...␣failed␣to␣open␣(' ...
                                          saveError.message ...
                                          '),␣but␣let''s␣continue␣with␣other␣ports.']);
                    continue; % there are more ports to try
                end
            end

            % Detect hardware flow control support
            if strcmp(obj.serialPort.PinStatus.ClearToSend,'on'),
                % we assume the interface supports hardware flow control
                obj.serialPort.FlowControl = 'hardware'; % turn hardware flow control on
                obj.hardwareFlowControl = true; % save this conclusion
            else
                % we assume the interface does not support hardware
                % flow control
                obj.serialPort.FlowControl = 'none'; % turn hardware flow control off
                obj.hardwareFlowControl = false; % save this conclusion
            end



            % Check for SPI interface
            obj.serialPort.Terminator = 85; % ASCII 'U'
            availableBytes = obj.serialPort.BytesAvailable;
            if availableBytes > 0
                dontCare = fscanf(obj.serialPort,'%c',obj.serialPort.BytesAvailable);
            end
            if obj.hardwareFlowControl
```

```
                    % ask for identification ('?') and for
                    % termination ('M') with 'U' (decimal 85)
                    fwrite(obj.serialPort,'?M');
                else
                    % do not rely on an input buffer, so wait
                    % a little between both bytes
                    fwrite(obj.serialPort,'?');
                    %pause(5);
                    fwrite(obj.serialPort,'M');
                    %pause(5);
                end

                % Unfortunately, 'U' (the termination character) also occurs
                % in the identification. The identification is at least 17
                % characters long (date and time), so we at least read 17
                % characters without waiting for 'U'.
                obj.spiIdentification = transpose(char(fread(obj.serialPort,17))); % read the first 17 characters
                obj.spiIdentification = [obj.spiIdentification fscanf(obj.serialPort,'%c')]; % read until 'U'
                obj.spiIdentification = obj.spiIdentification(1:end-1); % strip the trailing 'U'
                if numel(obj.spiIdentification) > 17
                    if obj.hardwareFlowControl
                        flowControlText = 'yes';
                    else
                        flowControlText = 'no';
                    end

                    SpiInterface.logInfo(['...␣found␣' obj.spiIdentification ...
                                         '␣(hardware␣flow␣control:␣' flowControlText ').']);
                    break; % escape from the tryPort loop
                else
                    fclose(obj.serialPort);
                    delete(obj.serialPort);
                    if remainingAvailablePorts == 0; % last or only port
                        if multiplePorts,
                            SpiInterface.logInfo(['...␣failed␣to␣identify,␣last␣port,␣so␣throw␣error.']);
                        end
                        error(['No␣self-identifying␣device␣found␣among' strcat(availablePorts{1:end})]); % last port, failed
                    else % by definition multiple ports
                        SpiInterface.logInfo(['...␣failed␣to␣identify,␣but␣let''s␣continue␣with␣other␣ports.']);
                        continue; % there are more ports to try
                    end
                end % identification?
            end % tryPort loop

            obj.writeBuffer = '';
        end

        function delete(obj)
            % Note that this destructor may be called upon multiple times
            % while obj.serialPort is the same, because copies of an
            % SpiInterface instance can be made. Therefore, we first check
            % that the serialPort is still valid
            if isvalid(obj.serialPort)
                serialPortName = obj.serialPort.Name;
                fclose(obj.serialPort);
```

```
                delete(obj.serialPort);
                SpiInterface.logInfo(['Closed␣serial␣port␣' serialPortName '.'
                    ]);
            end
        end
        %% Burst read/write functions (efficient)
        function writeMultiple(obj, startRegister, registerValues)
            %> Efficiently writes a list of SPI register values starting at
            %> startRegister.
            %>
            %> @param startRegister First SPI register number to write.
            %> @param registerValues 1D array of register values in [0,255].


            for registerCount = 1:numel(registerValues)
                registerNumber = startRegister + registerCount - 1;
                obj.writeNotYet(registerNumber,registerValues(registerCount));
            end
            obj.flushWriteBuffer();
        end

        function readValues = readMultiple(obj, startRegister, stopRegister)
            %> Efficiently read a contiguous sequence of registers.
            %>
            %> @param startRegister First SPI register number to read.
            %> @param stopRegister Last SPI register number to read.

            for registerNumber = startRegister:stopRegister
                obj.readNotYet(registerNumber)
            end
            obj.flushWriteBuffer;
            readValues = obj.readBytes(stopRegister-startRegister+1);
        end

        %% Reading and writing individual registers (less efficient)
        function readValue = read(obj,registerNumber)
            %> Less efficiently read one register
            readNotYet(obj,registerNumber);
            obj.flushWriteBuffer();
            readValue = obj.readBytes(1);
        end

        function write(obj,registerNumber,dataByte)
            %> Less efficiently write one register
            obj.writeNotYet(registerNumber,dataByte);
            obj.flushWriteBuffer();
        end

        %% Postponed reading/writing helper functions
        % These functions don't postpone when hardware flow control is off.
        function readNotYet(obj,registerNumber)
            obj.writeBuffer = [obj.writeBuffer 'R' registerNumber];
            if not(obj.hardwareFlowControl)
                obj.flushWriteBuffer();
            end
        end

        function writeNotYet(obj,registerNumber,dataByte)
            assert(registerNumber <= 127,'Register␣number␣should␣be␣<=␣127.');
            obj.writeBuffer = [obj.writeBuffer 'R' registerNumber+128 dataByte
                ];
            if not(obj.hardwareFlowControl)
                obj.flushWriteBuffer();
```

```
            end
        end

        function flushWriteBuffer(obj,varargin)
            %> Write the contents of the write buffer at once to the serial
            %> port. Make sure that we do not exceed the OutputBufferSize
            %> of the serial port.
            while numel(obj.writeBuffer) > 0,
                numberOfBytesAtOnce = obj.serialPort.OutputBufferSize;

                if obj.hardwareFlowControl
                    % wait until the CTS is on again, to be sure that another write
                    % command will not block
                    waitCount = 500; % (milliseconds)
                    while strcmp(obj.serialPort.PinStatus.ClearToSend,'off')
                        pause(0.01); % smallest pause (in seconds) supported on all platforms
                        waitCount = waitCount -10;
                        if waitCount <= 0
                            error('Timeout␣while␣waiting␣for␣CTS␣to␣go␣low␣(ON).');
                        end
                    end
                end %obj.hardwareFlowControl

                fwrite(obj.serialPort,obj.writeBuffer(1:min(end,numberOfBytesAtOnce)));
                obj.writeBuffer = obj.writeBuffer(min(end,numberOfBytesAtOnce)+1:end);
            end


        end % flushWriteBuffer

        function bytesRead = readBytes(obj,numberOfBytes)
            [bytesRead,readCount,errorMsg] = fread(obj.serialPort,numberOfBytes);
            if ~isempty(errorMsg)
                throw(MException('SpiInterface:readError',errorMsg));
            end
            bytesRead = transpose(bytesRead);
        end

    end % methods

    %% Logging facilities
    methods (Static = true, Access = private)
        function logInfo(infoText)
            fprintf(1,'SpiInterface:␣%s\n',infoText);
        end
    end
end % classdef
```

C.2.2 ImecModule

```
classdef ImecModule
    %> Agent of IMEC's SPI controllable devices, with a type-ID on SPI
    %> register 0.
    %>
    %> @version 0.9
    %> @author Sjoerd Op 't Land
    %> @since 2010-06-16

```

```
    properties
        interface % the SPI interface to talk with the module
        id % the device ID as read from SPI register 0
        type % type of IMEC module (string)
    end % properties

    methods
        %% Discovery of the device
        function obj = ImecModule(spiInterface)
            obj.interface = spiInterface;
            obj.id = obj.readId();

            switch obj.id
                case 1
                    obj.type = 'UWB␣Tx';
                case 2
                    obj.type = 'UWB␣Rx␣Building␣blocks';
                case 3
                    obj.type = 'NB␣Tx';
                case 4
                                        obj.type = 'UWB␣conventional␣Rx';
                case 5
                    obj.type = 'UWB␣feedback␣Rx';
                case 6
                    obj.type = 'UWB␣Tx␣module';
                case 7
                    obj.type = 'UWB␣Rx␣module';
                otherwise
                    obj.type = sprintf('<unknown␣device␣ID␣%d>',obj.id);
            end
                        obj.interface.write(0,1); % don't know why, but found this ritual
            % in the Visual C++ code

            ImecModule.logInfo(sprintf('%s␣found',obj.type));
        end

        %% Convenience functions
        function id = readId(obj)
            id = obj.interface.read(0);
        end

        function registerListing = readFirstRegisters(obj)
            %> Reads out SPI registers [1,15]
            registerListing = obj.interface.readMultiple(1,15);
        end

        function writeFirstRegisters(obj,firstRegisterValues)
            %> Writes list of register values, starting from SPI register 1
            %>
            %> @param firstRegisterValues 1D array or list of register values in [0,255].

            obj.interface.writeMultiple(1,firstRegisterValues);
        end

    end % methods


    %% Logging facilities
    methods (Static = true, Access = private)
        function logInfo(infoText)
            fprintf(1,'ImecModule:␣%s\n',infoText);
```

```
        end
    end
end % classdef
```

C.2.3 ImecUwbTxModule

```
classdef ImecUwbTxModule < ImecModule
    %> Agent of IMEC’s UWB Tx module (device ID 7), with support for
    %> programming the TX controller using SPI registers 13, 14 and 15.
    %>
    %> @author Sjoerd Op ’t Land
    %> @since 2010-06-16


    methods
        %% Discovery and assertion of the device
        function obj = ImecUwbTxModule(imecModuleArguments)
            obj = obj@ImecModule(imecModuleArguments);

            assert(strcmp(obj.type,’UWB␣Tx␣module’),’Did␣not␣find␣UWB␣TX␣module␣(ID=6)’)
            ImecUwbTxModule.logInfo(’found!’);
        end

        %% Writing to the TX controller
        function writeByteToTxController(obj,address,dataByte,varargin)
            %> @param address TX controller register address
            %> @param dataByte TX controller register value
            %> @param spi15AlreadyCleared If passed and true, we assume
            %>        that SPI register 15 is already zero
            %> @param dontFlushBuffer If passed and true, we do not yet
            %>        flush the write buffer.

             obj.interface.writeNotYet(13,address);
             obj.interface.writeNotYet(14,dataByte);
             if (size(varargin,2) == 0) || varargin{1} == false
                 obj.interface.writeNotYet(15,0);
             end
             obj.interface.writeNotYet(15,1);
             obj.interface.writeNotYet(15,0);
             if (size(varargin,2) <= 1) || varargin{2} == false
                 obj.interface.flushWriteBuffer();
             end
        end

        function writeFileToTxController(obj,filePath)
            %> @todo Eliminate loop by proper matrix/cell operations
            obj.interface.write(15,0); % make sure that WriteEnable is low

            dataFile = fopen(filePath,’r’);
            nextLine = fgetl(dataFile);
            while(nextLine ~= -1)
                % read and interpret one CSV (hex) line of the file
                %ImecUwbTxModule.logInfo(nextLine);
                hexStrings = regexpi(nextLine,’0x([0-9A-F]{1,2})\s*,\s*0x([0-9A-F]{1,2})’,’tokens’);
                address = hex2dec( hexStrings{1,1}(1) );
                dataByte = hex2dec( hexStrings{1,1}(2) );

                % write to TX controller via SPI
                obj.writeByteToTxController(address,dataByte,true,true)

                nextLine = fgetl(dataFile);
            end
```

```
            fclose(dataFile);
            obj.interface.flushWriteBuffer();
        end

        function stop(obj)
            %> Stop the TX by clearing the programmed bit
            writeByteToTxController(obj,0,0)
        end

        function start(obj)
            %> Start the TX by setting the programmed bit
            writeByteToTxController(obj,0,1)
        end
    end % methods




    %% Logging facilities
    methods (Static = true, Access = private)
        function logInfo(infoText)
            fprintf(1,'ImecUwbTxModule:␣%s\n',infoText);
        end
    end
end % classdef
```

D Measurement and Analysis Code

We start describing the measurement and analysis code by presenting coding conventions and file organisation. Then, we will describe how to perform measurement and analysis tasks. The MATLAB code is not included, because there was no time left to conform all comment to produce correct HTML/LaTeX markup.

D.1 Conventions and File Organisation

These are the conventions that the MATLAB code files should adhere to. The was no priority/ time to conform all comments.

D.1.1 Conventions

All names, directories and variables alike, try to be descriptive of the meaning of their contents. This applies to all levels of the hierarchy, so even

```
for i=1:10
        listOfNumbers = [listOfNumbers i];
end
```

is deprecated, because `i` has a more particular meaning, that could be expressed with a proper name (for instance, `newNumber`).

Furthermore, all names are written in camelCase, capitalizing the first letter of all words that would be separated by spaces in common text. Abbreviations should be avoided, because that reduces doubt about the spelling used. (For example, did I use `propVal`, `propertyVal` or `propValue`?). If commonly used abbreviations are used, they are capitalized in camelCase, e.g. `getSpiInterface()`. Using numbers should be avoided, as they cannot be capitalised. The first letter of a variable or function name is lower case, whereas the first letter of a class name is upper case. Examples: `variableName`, `propertyValue`, `getHttpProperty()` and `ClassName`.

Constant names have no exceptional style, so they are written just like any variable name. For what is not defined in this subsection, refer to [84].

D.1.2 Directory structure

All conclusions, great and small, drawn during (this) empirical research, are interpretations of subjective observations. However, we can make the artificial distinction between interpretations that we plan to doubt and interpretations that we deem obvious. In the sequel, we will call the former ‘interpretations’ and the latter ‘measurements’.

For example, we choose to trust that the oscilloscope display that we seem to see with our eyes, really is a time-domain representation of the voltage at its terminals. Therefore, I will call the oscilloscope reading a measurement. Conversely, I use an experimental algorithm to estimate a channel impulse response from the oscilloscope reading. This algorithm still leaves room for improvement, so its outcomes are open to discussion. Therefore, I will call the estimated channel impulse responses ‘interpretations’.

The directory tree is as follows:

- `measurement` Scripts, functions and class definitions to perform measurements.
- `measurements` Contains all measurements and is ‘holy’; after taking the measurements, it’s contents should not change anymore.
- `interpretation` Scripts, functions and class definitions to perform interpretation.

- `interpretations` Stored interpretations, which can be re-generated by running scripts in `/interpretation`.
- `documentation` MATLAB-extracted documentation from all source code.
- `utilities` Miscellaneous utility functions, both used by measurement and interpretation scripts.

D.1.3 Scoping

With the proposed directory structure, all useful functions and class definitions are scattered over the repository. A quick-and-dirty way to find everything, is to add all (sub)folders to the path. This means that function and class names should be unique in all of the repository.

A utility function `recursiveAddPath(rootDirectory)` is created in `utilities/`, so all folders can be added by calling `rootDirectory = recursiveAddPath('..')` from within `utilities`. Scripts can avoid unnecessarily calling this function by checking the existence of workspace variable `rootDirectory`:

```
%% Fix MATLAB's path, if necessary
[herePath, ~,~,~] = fileparts(mfilename('fullpath')); cd(herePath); % force that we are in the directory of this M-file
if not(evalin('base','exist(''rootDirectory'')')), addpath('../../utilities'); assignin('base','rootDirectory',recursiveAddPath('../..')); end
```

This solution is elegant nor is it strict, and a the future of the repository may call upon a better way of scoping.

D.1.4 SVN specifics

To ignore particular files, we would like to set `global-ignores`. Unfortunately, we cannot, because the repository settings are out of our hands. Therefore, we use the script `recursivelySetSvnProperties.bash` to set the `svn:ignore` property on all directories. As soon a a repository becomes available where `global-ignores` can be set, the bash script should be removed and all `svn:ignore` properties should be set to inherit from the repository.

D.1.5 Documentation

MATLAB has built-in facilities for both online help and publishable, formatted documentation and demos. Unfortunately, these two goals require separately maintained code.[1] This is deemed impractical; in an experimental and dynamic set of source files, nobody will be motivated to keep both sources of documentation in sync.

We decide to design a default header that displays acceptably using MATLAB's `help` and `doc` commands, while easy to convert to a format that is publishable. For functions, a file looks like this:

```
function acceptedReward = rewardPerson(personName, bonusMoney)
%rewardPerson Reward an employee both emotionally and financially.
% As is good practice, employees should be rewarded from time to time for their
% efforts. Of course, if this rewards become his sole motivation, we speak of
% extrinsic motivation. In modern times, we deem this suboptimal and should
% try, together with the employee, to regain intrinsic motivation as well.
% Good old times...
%
%% Inputs
```

[1] From MATLAB 2010b Documentation (after describing formatting source code for online help): "You can create a reference page in an HTML authoring environment by importing the help text for a function and formatting the text. For example, you need to remove the percent sign (%) character from the beginning of each line of text, and make sure that spaces separate words." This implies having to maintain two sources of truth.

```
%
% * *personName* _string_: Name of the person, who should be rewarded for his
%     hard work for the institution.
% * *bonusMoney* _double_: The quantifiable part of the reward (EUR).
%
%% Outputs
% * *acceptedReward* _logical_: Whether or not the person accepted the reward.

%% History
% Created: optland, _2010-09-10_
%
% Modified: $$ $$ LastChangedBy $$ $$, _$$ $$ LastChangedDate $$ $$_
%
% _$$ $$ URL $$ $$_
%
%% Code walkthrough
```

...and from there the real code starts. Note the dollar signs at the end, these are magic SubVersioN (SVN) tags that are expanded to the true values while checking out the SVN.

D.1.6 Revisions

17 Working ranging set-up with V1 LowBand UWB hardware.

28 Repository cleaned up to contain only the useful files and to be consistent with this appendix.

D.2 Common Measurement and Analysis Tasks

We will present how-to's for common measurement and analysis tasks. Guided by these descriptions, we introduce the object model used by the MATLAB code.

D.2.1 Taking Measurements

Connect all measurement equipment and run `measurement/rangingUsingScopeV1.m`. This script initialises the measurement set-up, takes measurements (b0, b1, I and Q waveforms) and feeds them to the ranging algorithm.

The following objects are worth noticing. `transceiverParameters` is a struct that contains all settings necessary for the transmitter and receiver to communicate, such as mode and frequency. `setup` is a `MeasurementSetup` object, which is an agent of all measurement equipment (USB-SPI interfaces and the oscilloscope). `algoritm` is a `RangingAlgorithm` object, which is a wrapper for (mostly Dries') ranging functions. (Indeed, the ranging algorithm is performing interpretation, so its class definition can be found in `interpretation/ranging`.)

Note the experimental `measurement/rangingUsingScopeErrorEstimationV1` script, that also invokes the error estimation, to demonstrate error estimation in a live fashion.

D.2.2 Saving Multiple Measurements

In the measurement campaign, it is important to quickly perform many measurements and save the results. According to the philosophy of the artificial measurement/interpretation distinction, we try only to save measurements and no interpretations.

A measurement performed is called a *take*. A set of one or more takes performed under the same apparent conditions is called a *try* (such as antenna positions, obstacles, time and transceiver parameters). These conditions can be stored in a `TryParameters` object. Practically, every try is stored as a directory in `measurements/`, containing one or more `take_nn.mat` files, a `tryParameter.mat` file and a `webcamPhoto.jpg`.

The latter is a photo taken by a webcam hanging from the ceiling. YawCam[2] is used to interface between MATLAB and the webcam. That is, YawCam presents a webserver at `http://localhost:8080/out.jpg`. Furthermore, YawCam can be set to keep a record of the 15 most recent time-lapse photos in a directory. These photos can be dumped to create a timelapse movie later on.

A script to efficiently perform multiple measurements is `multiMeasurement.m`. It first generates the try parameters of all the measurements that we want to take. Then, it guides the user through taking these measurements. Practically, the `TryParameters` constructor can take swept parameters, like obstacle partiality, and calculates antenna and obstacle positions and orientations from them.[3] The try parameters are generated by sweeping one or more parameters. By running the script with `rehearsal = true;`, no actual measurements are taken, but the try parameters are shown quickly in succession. This way, the user can verify that he performed the correct parameter sweep. If he is satisfied, he sets `rehearsal` to false and runs the script. The measurement procedure described in Section 5.4 can then be performed.

D.2.3 Calculating the Power Delay Profiles

The next step in analysis is calculation the power delay profiles of the measurements taken with `multiMeasurement.m`. This is done by `interpretation/powerDelayProfile/interpretMeasurements.m`. For every take found in `measurements`, a `TakeSummary` object is created, containing the power delay profile. All the takes belonging to one try are grouped in a `TakeSummarySet` object and stored to `interpretations/powerDelayProfiles`, each in a directory and together with a low-quality version of the webcam photo. The script automatically skips measurements that are already interpreted, so (1) the user can quickly run the script if only a few measurements were added and (2) if the algorithm changes, the user should delete all interpretations and rerun `interpretMeasurements`.

D.2.4 Perform Ranging

The ranging algorithms are implemented as methods of `TakeSummary` objects. For example, one can load a `takeSummarySet.mat` file, and type:

```
>> takeSummarySet.takeSummaries(1).extrapolatedEdgeTime
ans =
   3.7138e-08
>> takeSummarySet.takeSummaries(1).extrapolatedEdgeTimeRange
ans =
     1.2277
```

To show a certain time's (`extrapolatedEdgeTime`) relation with the distance, the user can run `interpretation/ranging/plotRangingFeature.m`. This should result in a screen like Figure D.1.

Reading through this script, one observes the analysis steps that are similarly seen in other interpretation scripts:

1 Call `queryTries()` to select a subset of the tries. For example, one could want to only select tries without an obstacle, or only tries with a partiality higher than 1.0.

2 From every try (`TakeSummarySet`), collect all the `TakeSummary` objects. This results in one array `selectedTakes`.

3 For every selected take, calculate the requested τ.

4 Fit $\tau(x)$ curves to the data points and report the average absolute error and standard deviation (accuracy and precision).

[2] `http://www.yawcam.com`

[3] Note that only positions and orientations are stored, as the partiality is an interpretation itself.

5 Plot the data points and the curves, together with a callback implementation to allow the user to click on data points.

A similar ranging script `plotRanging` shows the range estimate as function of the distance. A different script, `signalStrength` allows to fit an inverse quadratic curve to the received energy as function of the distance. To verify that obstacles really induce ranging errors, the user can run `visualiseTriesPartiality.m`, which plots the ranging error against the obstacle partiality, for the different obstacles.

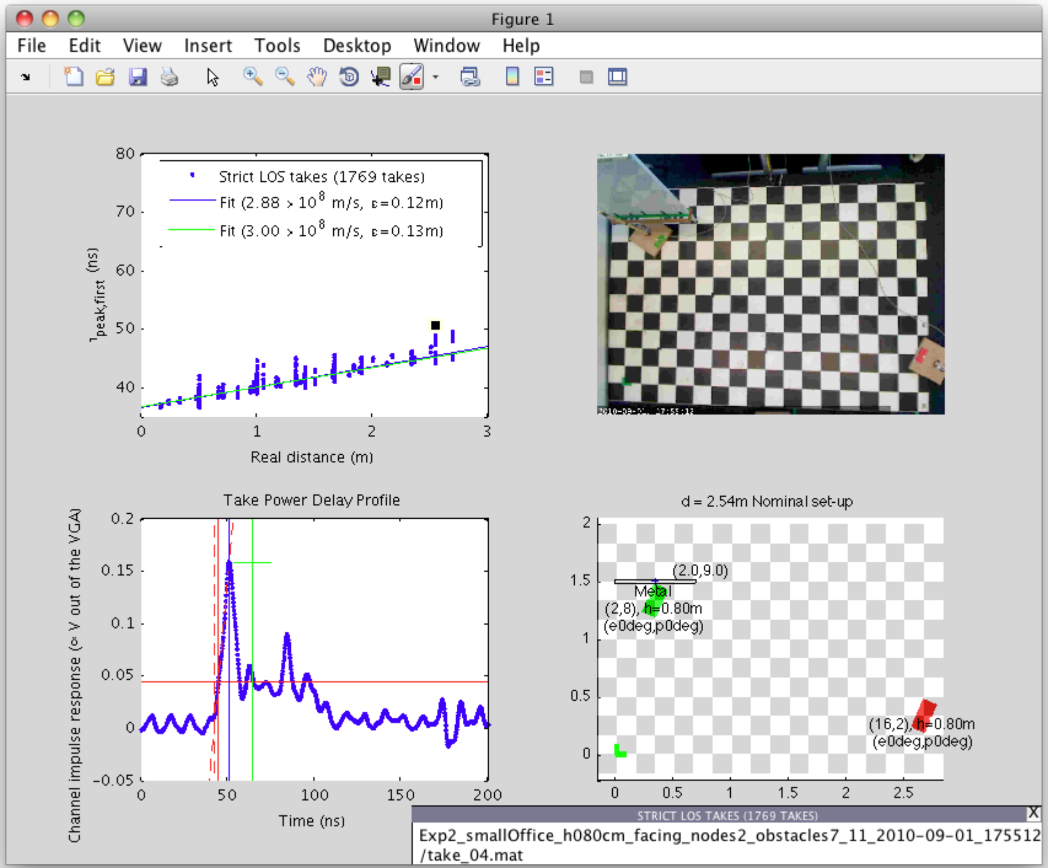


Figure D.1: Screenshot of `plotRangingFeature.m`. Every point in the top-left scatter plot represents a take. Click a point to show the power delay profile, photo and try parameters visualisation of the take.

D.2.5 Inspect Feature Values

The features are implemented as methods of the `TakeSummary` class, just like the ranging functions. The advantage is that one can easily change the implementation of a feature calculation and rerun analysis scripts. The disadvantage is that, when the feature calculations are finalised, the user has to await feature evaluation for every analysis script.

A next logical step in analysis would be inspecting the feature values, to study the relation with the ranging error. This can be done with the scripts in `interpretation/featureValues/`. For example, `calculateHistograms` gives a bivariate histogram, a 2D and a 3D scatterplot. Using `errorAgainstFeature`, we can show scatterplots and simple histograms for different series (for example, to compare line-of-sight (LOS) and non-line-of-sight (NLOS) tries).

The two scripts mentioned above evaluate the features using the implementation in `TakeSummary`. Assuming that the feature implementations are finalised by now, the next two scripts use the feature values cached in `/interpretations/featureValues/allFeaturesAndErrors.ma` to speed up analysis. To fill the cache, run `calculateFeatureValues`. Consequently, this script needs to be rerun if the feature implementations in `TakeSummary` change.

To estimate the ranging error based on feature values, one can define fitted functions in `fitFunctionDefinitions`. The user can verify the fit for each feature separately with `fitIndividualErrorEstimates`: a histogram of the feature is plotted, overlayed with the fit. The correlation between the individual error estimates and the real ranging error can be studied with `fitErrorEstimate`. The latter scripts uses multilinear regression to obtain combination coefficients (`predictorWeights`. Using these coefficients, the joint error estimate is calculated and plotted against the real ranging error. Furthermore, the false alarm and false accept rates are plotted as function of the threshold.

Important: when the user is satisfied about the fitted functions and the predictor weights, they should be copied to the estimator functions in `TakeSummary`, to be used in the sequel.

D.2.6 Perform Localisation

The actual localisation scripts are located in `interpretation/localisation`. To perform LMSE ToA localisation, the range measurements taken in the same environment, with the same obstacle positions and orientations, have to combined to one *scene*. That is, the `Scene` class is a container of many tries (`TakeSummarySets`). All scenes, given the measurements, can be grouped with the `splitScenes`. This results in `interpretations/localisation/allCompleteScenes.mat`, containing 13 `Scene`s. The nodes included in a `Scene` can be viewed like this after loading `allCompleteScenes.mat`:

```
completeScenes(6).visualise()
```

As can be seen, scene 1 is void of obstacles, mostly filled with node positions from experiment 2. Scenes 2 up and until 10 belong to experiment 2, each containing one obstacle and 7 nodes. Scenes 11 up and until 13 belong to experiment 3, containing different obstacle types. For convenience, scenes 1–10 can be copied to `interpretations/localisation/experiment2Scenes.mat` and scenes 11–13 to `experiment3Scenes.mat`.

Let a *localisation problem* mean estimating the location of one node (the *target*), given the position of other nodes (the *anchors*) and the range measurements between the target and each anchor. Given one scene which contains measurements, generally, many localisation problems can be generated by selecting different targets and selecting a subset of the remaining nodes as anchors. The `SingleLocalisation` represents a localisation problem. `Scene` has a method that returns all localisation problems that localise a given target:

```
>> problems = completeScenes(6).targetLocalisation(4)
problems =
  42x1 SingleLocalisation
>> problems(40).visualise(true)
```

This example shows the 40th problem that can be extracted from scene 6, with node 4 as a target.

The localisation problem can be solved by calling the `localise` method:

```
>> problems(40).localise('simpleLocalisation',true)
rangeErrorEstimates =
    [5]    [0.0438]
    [6]    [0.0352]
```

```
    [1]    [0.0834]
    [3]    [0.1914]
    [7]    [0.4081]
ans =
    0.6946
```

Apparently, this solution has an absolute error of 69 cm. Other localisation strategies can be applied:

```
>> problems(40).localise('correctedLocalisation',true)
ans = 0.3311
>> problems(40).localise('combinedLocalisation',true)
ans = 0.1131
>> problems(40).localise('discardingLocalisation',true)
ans = 0.0300
```

To test the all localisation strategies, on all problems, for all targets in all scenes, run `localiseAll`. Note that this generally takes a while, so guard the resulting `localisationResults.mat` carefully. Histograms of and comparisons between the localisation results can be plotted with `evaluateLocalisationPerformance`.

Bibliography

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, October 2007. [Online]. Available: http://dx.doi.org/10.1109/TSMCC.2007.905750

[2] "IEEE standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–305, 2006.

[3] "Revision of part 15 of the commission's rules regarding ultra-wideband transmission systems," *ET Docket 98-154*, no. 02-48, April 2002.

[4] F. Chiti and R. Fantacci, "Wireless sensor network paradigm: Overview on communication protocols design and application to practical scenarios," *EURASIP Newsletter*, vol. 17, no. 4, pp. 6–27, December 2006. [Online]. Available: http://www.eurasip.org/newsletter/newsletter-17-4.pdf

[5] Localisation in smart dust networks. Project Description. [Online]. Available: http://www.utwente.nl/ewi/te/projects/SRR/Smart_dust/

[6] N. Patwari, J. N. Ash, S. Kyperountas, A. O. H. III, R. L. Moses, and N. S. Correal, "Locating the nodes," *IEEE Signal Processing Magazine*, July 2005.

[7] SiRFstarIV™ GSD4e product brochure. [Online]. Available: http://www.csr.com/products/35/sirfstariv-gsd4e

[8] GPS accuracy - how accurate is it? [Online]. Available: http://www.maps-gps-info.com/gps-accuracy.html

[9] Time Domain. How does UWB differ from conventional RF technologies? [Online]. Available: http://www.timedomain.com/differences.php

[10] H. Nikookar and R. Prasad, *Introduction to Ultra-Wideband for Wireless Communications*. Springer, 2009.

[11] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 404 –426, feb. 2009.

[12] L. G. Hector and H. L. Schultz, "The dielectric constant of air at radiofrequencies," *Physics*, vol. 7, no. 4, pp. 133–136, 1936. [Online]. Available: http://link.aip.org/link/?JAP/7/133/1

[13] E. Karapistoli, F.-N. Pavlidou, I. Gragopoulos, and I. Tsetsinas, "An overview of the IEEE 802.15.4a standard," *Communications Magazine, IEEE*, vol. 48, no. 1, pp. 47 –53, january 2010.

[14] H. B. Lee, "A novel procedure for assessing the accuracy of hyperbolic multilateration systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-11, no. 1, pp. 1–15, January 1975.

[15] "IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirement part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)," *IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006)*, pp. 1 –203, 2007.

[16] J.-Y. Lee and R. Scholtz, "Ranging in a dense multipath environment using an UWB radio link," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 9, pp. 1677 – 1683,

Dec. 2002.

[17] D. Dardari, A. Conti, J. Lien, and M. Z. Win, "The effect of cooperation on localization systems using UWB experimental data," *EURASIP J. Adv. Signal Process*, vol. 2008, pp. 1–11, 2008.

[18] J. Schroeder, S. Galler, K. Kyamakya, and K. Jobmann, "NLOS detection algorithms for ultra-wideband localization," March 2007, pp. 159 –166.

[19] I. Guvenc, C.-C. Chong, and F. Watanabe, "NLOS identification and mitigation for UWB localization systems," mar. 2007, pp. 1571 –1576.

[20] S. Venkatesh and R. Buehrer, "Non-line-of-sight identification in ultra-wideband systems based on received signal statistics," *Microwaves, Antennas Propagation, IET*, vol. 1, no. 6, pp. 1120 –1130, dec. 2007.

[21] S. Maranò, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 7, pp. 1026 –1035, sep. 2010.

[22] W. Gao, K. Veeramachaneni, G. Kamath, and L. Osadciw, "A novel ultrawide band locationing system using swarm enabled learning approaches," mar. 2009, pp. 129 –136.

[23] J. M. Huerta, A. Giremus, J. Vidal, and J.-Y. Tourneret, "Joint particle filter and UKF position tracking under strong NLOS situation," aug. 2007, pp. 537 –541.

[24] J. González, J. Blanco, C. Galindo, A. O. de Galisteo, J. Fernández-Madrigal, F. Moreno, and J. Martínez, "Mobile robot localization based on ultra-wide-band ranging: A particle filter approach," October 2008, preprint submitted to Elsevier.

[25] A. Hugine, H. Volos, J. Gaeddert, and R. Buehrer, "Measurement and characterization of the near-ground indoor ultra wideband channel," vol. 2, apr. 2006, pp. 1062 –1067.

[26] A. Molisch, J. Foerster, and M. Pendergrass, "Channel models for ultrawideband personal area networks," *Wireless Communications, IEEE*, vol. 10, no. 6, pp. 14 – 21, dec. 2003.

[27] M. Peter, R. Felbecker, W. Keusgen, and J. Hillebrand, "Measurement-based investigation of 60 GHz broadband transmission for wireless in-car communication," sep. 2009, pp. 1 –5.

[28] T. Rappaport, S. Seidel, and K. Takamizawa, "Statistical channel impulse response models for factory and open plan building radio communicate system design," *Communications, IEEE Transactions on*, vol. 39, no. 5, pp. 794 –807, may. 1991.

[29] T. Rappaport, "Characterization of uhf multipath radio channels in factory buildings," *Antennas and Propagation, IEEE Transactions on*, vol. 37, no. 8, pp. 1058 –1069, aug. 1989.

[30] A. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *Selected Areas in Communications, IEEE Journal on*, vol. 5, no. 2, pp. 128 – 137, feb. 1987.

[31] Y. Jeong, "A new method for calibration of NLOS error in positioning systems," *IEICE Transactions on Communications*, vol. 85, no. 5, 2002. [Online]. Available: http://csp.yonsei.ac.kr/publications/files/A%20new%20method%20for%20calibration%20of%20NLOS%20error%20in%20positioning%20systems.pdf

[32] E. Genender, C. Holloway, K. Remley, J. Ladbury, G. Koepke, and H. Garbe, "Use of reverberation chamber to simulate the power delay profile of a wireless environment," in *Electromagnetic Compatibility - EMC Europe, 2008 International Symposium on*, 8-12 2008, pp. 1 –6.

[33] M. Schack, M. Jacob, and T. Kiirner, "Comparison of in-car UWB and 60 GHz channel measurements," apr. 2010, pp. 1 –5.

[34] E. W. Weisstein, "k-statistic," MathWorld–A Wolfram Web Resource. [Online]. Available: http://mathworld.wolfram.com/k-Statistic.html

[35] L. Greenstein, D. Michelson, and V. Erceg, “Moment-method estimation of the Ricean K-factor,” *Communications Letters, IEEE*, vol. 3, no. 6, pp. 175 –176, jun. 1999.

[36] C. Holloway, D. Hill, J. Ladbury, P. Wilson, G. Koepke, and J. Coder, “On the use of reverberation chambers to simulate a Rician radio environment for the testing of wireless devices,” *Antennas and Propagation, IEEE Transactions on*, vol. 54, no. 11, pp. 3167 –3177, nov. 2006.

[37] T.-K. Liu, D. Kim, and R. Vaughan, “A high-resolution, multi-template deconvolution algorithm for time-domain UWB channel characterization,” *Electrical and Computer Engineering, Canadian Journal of*, vol. 32, no. 4, pp. 207 –213, 2007.

[38] W. Yang and Z. Naitong, “A new multi-template CLEAN algorithm for UWB channel impulse response characterization,” in *Communication Technology, 2006. ICCT ’06. International Conference on*, 2006, pp. 1 –4.

[39] M.-S. Choi, G. Grosskopf, and D. Rohde, “Statistical characteristics of 60 GHz wideband indoor propagation channel,” vol. 1, sep. 2005, pp. 599 –603.

[40] J. Romme, “UWB channel fading statistics and transmitted-reference communication,” Ph.D. dissertation, TU Graz, 2008.

[41] B. Denis and N. Daniele, “NLOS ranging error mitigation in a distributed positioning algorithm for indoor uwb ad-hoc networks,” in *Wireless Ad-Hoc Networks, 2004 International Workshop on*, 31 2004, pp. 356 – 360.

[42] R. Casas, A. Marco, J. J. Guerrero, and J. Falcó, “Robust estimator for non-line-of-sight error mitigation in indoor localization,” *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 156–156, uary.

[43] W.-K. Chao and K.-T. Lay, “NLOS measurement identification for mobile positioning in wireless cellular systems,” sep. 2007, pp. 1965 –1969.

[44] V. Brethour, “Ranging with IEEE 802.15.4a draft 2,” Presentation, May 2006.

[45] G. Dolmans, O. Rousseaux, L. Huang, T. Fu, B. Gyselinkx, S. d’Amico, A. Baschirotto, J. Ryckaert, and B. van Poucke, “UWB radio transceivers for ultra low power and low data rate communications,” sep. 2007, pp. 152 –157.

[46] J. Ryckaert, G. Van der Plas, V. De Heyn, C. Desset, G. Vanwijnsberghe, B. Van Poucke, and J. Craninckx, “A 0.65-to-1.4nj/burst 3-to-10ghz UWB digital TX in 90nm cmos for IEEE 802.15.4a,” feb. 2007, pp. 120 –591.

[47] H. Schantz, “Planar elliptical element ultra-wideband dipole antennas,” in *Antennas and Propagation Society International Symposium, 2002. IEEE*, vol. 3, 2002, p. 44.

[48] Representative. [Online]. Available: http://www.britannica.com/EBchecked/topic/424706/Ockhams-razor

[49] H. W. Pflug, D. Neirync, J. Romme, K. Philips, and H. de Groot, “UWB pulse amplitude estimation method for ieee 802.15.4a,” March 2010, to be published at ICUWB 2010.

[50] H. Visser, “Low-cost, compact UWB antenna with frequency band-notch function,” in *Antennas and Propagation, 2007. EuCAP 2007. The Second European Conference on*, 11-16 2007, pp. 1 –6.

[51] R. Wilson, “Propagation losses through common building materials,” Master’s thesis, University of Southern California, March 2002.

[52] *Agilent 83000A Series Microwave System Amplifiers*, Agilent Technologies, October 2009.

[53] M. Ingram, “Tutorial ECE4606: Wireless communications,” Presentation. [Online]. Available: http://www.ece.gatech.edu/research/labs/sarl/tutorials/ECE4606/19-Diffraction.pdf

[54] Fresnel zones and their effect. [Online]. Available: http://www.zytrax.com/tech/wireless/fresnel.htm

[55] I. Naqvi and G. El Zein, “Time domain measurements for a time reversal SIMO system in reverberation chamber and in an indoor environment,” in *Ultra-Wideband, 2008. ICUWB 2008. IEEE International Conference on*, vol. 2, 10-12 2008, pp. 211 –214.

[56] D. Harmer, A. Yarovoy, N. Schmidt, K. Witrisal, M. Russell, E. Frazer, T. Bauge, S. Ingram, A. Nezirovic, A. Lo, L. Xia, B. Kull, and V. Dizdarevic, “An ultra-wide band indoor personnel tracking system for emergency situations (europcom),” 2008.

[57] P. Holland and R. Welsch, “Robust regression using iteratively reweighted least-squares,” *Communications in Statistics-Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[58] Robust regression – MATLAB. [Online]. Available: http://www.mathworks.com/help/toolbox/stats/robustfit.html

[59] S. Gezici and H. V. Poor, “Position estimation via ultra-wideband signals,” *Computing Research Repository (CoRR)*, vol. abs/0807.2730, 2008.

[60] V. Krasnopolsky, “Elements of nonlinear statistics and neural networks,” Presentation, April 2006. [Online]. Available: http://polar.ncep.noaa.gov/mmab/people/vladimir/nntutor/Nonlinear_Statistics_and_NNs.pdf

[61] G. Orwell, *Nineteen eighty-four: a novel*. Penguin Books Harmondsworth, 1956.

[62] J. E. E. D. Acton, *Essays on freedom and power*. Beacon Press, Boston, Mass., 1948.

[63] P. Jansen. (2010, March) iDNA manifest 4.1. [Online]. Available: http://www.dotindividual.com/?p=17

[64] A. Avizienis, J.-C. Laprie, and B. Randell, “Fundamental concepts of dependability,” LAAS-CNRS, Research Report 1145, April 2001.

[65] J. Jones, P. Brett, G. Bryan-Brown, A. Graham, and E. Wood, “Zenithal bistabile displays,” ZBD Displays Ltc., Malvern Hills Science Park, Malvern, Worcs., WR14 3SZ, UK, Tech. Rep., March 2001. [Online]. Available: http://www.zbdsolutions.com/resources/downloads/white_papers/pdf10.pdf

[66] Msp430 ultra-low-power microcontrollers. [Online]. Available: http://focus.ti.com/lit/sg/slab034s/slab034s.pdf

[67] A. Abrahami Saba, H. Abolhassani, and M. Ghodsi, “Range-free passive acoustic localization,” dec. 2008, pp. 37 –42.

[68] Euclid of Alexandria, *Elements*. Self-published. [Online]. Available: http://aleph0.clarku.edu/~djoyce/java/elements/elements.html

[69] A. Hatami, K. Pahlavan, M. Heidari, and F. Akgul, “On RSS and TOA based indoor geolocation - a comparative performance evaluation,” vol. 4, April 2006, pp. 2267 –2272.

[70] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, “Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks,” *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 70–84, July 2005.

[71] D. Dardari, C.-C. Chong, and M. Win, “Improved lower bounds on time-of-arrival estimation error in realistic UWB channels,” in *Ultra-Wideband, The 2006 IEEE 2006 International Conference on*, 2006, pp. 531 –537.

[72] M. Pinsky, *Introduction to Fourier Analysis and Wavelets*. Brooks/Cole, 2002.

[73] R. A. Saeed, S. Khatun, B. M. Ali, and M. A. Khazani, “Performance of ultra-wideband time-of-arrival estimation enhanced with synchronization scheme,” *ECTI Transactions on Electrical Eng., Electronics, and Communications*, vol. 4, pp. 78–84, February 2006.

[74] N. Alsindi and K. Pahlavan, “Cooperative localization bounds for indoor ultra-wideband wireless sensor networks,” *EURASIP J. Adv. Signal Process*, vol. 2008, pp. 1–13, 2008.

[75] “ECMA-369 MAC-PHY standard.”

[76] WiMedia. (2010, January) List of certified PHYs. [Online]. Available: http://www.wimedia.org/en/ecosystem/certified_phy.asp?id=eco

[77] G. Dolmans and K. Philips, “An UWB transceiver for IEEE 802.15.4a WPAN,” oct. 2008, pp. 710 –712.

[78] D. Harmer, “Europcom: UWB radio for rescue services,” Rise 2008.

[79] IST. Overview IST-2001-32710. [Online]. Available: ftp://ftp.cordis.europa.eu/pub/ist/docs/ka4/mob_lobster_bxl1202_ucan.pdf

[80] *R284C0351053 Technical Data Sheet*, Radiall, No. 390, Yong He Road, Shanghai 200072, China, May 2007.

[81] *BLK-89-S+ SMA DC Block Typical Performance Curves*, Mini-circuits, P.O. Box 350166, Brooklyn, New York 11235-0003.

[82] “Measuring UWB spectral properties,” Holst Centre, Technical Note TN-07-WATS-TPX-XXX, 2010.

[83] H. W. Pflug, “Method to estimate impulse radio ultra-wideband peak power,” Holst Centre, Technical Note TN-10-WATS-TP2-047, March 2010.

[84] R. Johnson, “MATLAB programming style guidelines,” Datatool, Tech. Rep., 2002.

[85] C. Rizos, *Principles and Practice of GPS Surveying*. University of New South Wales, 1999. [Online]. Available: http://www.gmat.unsw.edu.au/snap/gps/gps_survey/principles_gps.htm

[86] “IEEE standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems,” *IEEE Std 802.16-2004 (Revision of IEEE Std 802.16-2001)*, pp. 1–857, 2004.

Abbreviations

AoA Angle of Arrival

ASIC Application Specific Integrated Circuit

AWGN Additive White Gaussian Noise

BPM Burst Position Modulation, information is conveyed in the timing (position) of a burst.

BPSK Binary Phase Shift Keying, information is conveyed in the polarity of the signal.

CIR Channel Impulse Response

COTS Commercial Of The Shelf

CSS Chirp Spread Spectrum

CTS Clear To Send

DC Direct Current, often used informally to indicate 0 Hz.

DCO Digitally Controlled Oscillator

DDP Dominant Direct Path, [18].

DFT Discrete Fourier Transform

DP Direct Path

EIRP Equivalent Isotropically Radiated Power

EOP End Of Preamble

EUWB European Ultra-WideBand, a European R&D project team on UWB.

FCC Federal Communications Commission

FCF Frequency Correlation Function

FFT Fast Fourier Transform, a fast implementation of the Discrete Fourier Transform (DFT).

FoM Figure of Merit

FPGA Field Programmable Gate Array, a popular SOPC.

GDoP Geometric Dilution of Precision, the standard deviation of the distance error, a metric for robustness. In the literature, GDoP is sometimes used to indicate RGDoP.

GPS Global Positioning System

GUI Graphical User Interface

HTML HyperText Markup Language

IC Integrated Circuit

IEEE Institute of Electrical and Electronics Engineers

IMEC Interuniversitair Micro-Elektronica Centrum, Inter-universitary centre for micro-electronics.

LCD Liquid Crystal Display

LFSR Linear Feedback Shift Register

LMedS Least Median of Squares, an error criterion more robust to outliers than LMSE.

LNA Low Noise Amplifier, one of the first subsystems in an RF receiver chain.

LO Local Oscillator

LOS line-of-sight

LR-WPAN Low-Rate Wireless Personal Area Network

MAC Medium Access Control

MATLAB MATrix LABoratory

LMSE Least Mean Squared Error

MPC MultiPath Component

NDDP NonDominant Direct Path, [18].

NF Noise Figure, the ratio of the output SNR and the input Signal to Noise Ratio (SNR), also expressed in dB.

NLOS non-line-of-sight, as opposed to LOS.

ODP Obstructed Direct Path, which can result in a NDDP or UDP channel condition.

OFDM Orthogonal Frequency Division Multiplexing

PA Power Amplifier, one of the last subsystems in an RF transmitter chain.

PC Personal Computer

PCB Printed Circuit Board

PDF Probability Density Function

PDP Power Delay Profile

PDU Protocol Data Unit, the type of unit exchanged at the boundary with the lower layer (see Figure D.2).

PHR PHY HeadeR

PHY PHYsical layer

PRF Pulse Repetition Frequency

PSDU PHY Service Data Unit

PSK Phase Shift Keying, information is conveyed in the phase of a carrier.

RBW Resolution Bandwidth

RDEV Ranging-capable Device

RF Radio Frequency

RFRAME Ranging Frame, a frame with the ranging bit set in the PHY HeadeR (PHR).

RGB Red Green Blue

RGDoP Relative Geometric Dilution of Precision, the ratio between the positioning error standard deviation and the ranging error standard deviation [85, sec. 1.4.9], a measure for the quality of the geometry. In the literature, RGDoP is often denoted Geometric Dilution of Precision (GDoP).

RMS Root Mean Squared

RSS Received Signal Strength

RSSI Received Signal Strength (RSS) Indication

RToA Return Time of Arrival, or RTT, or TWR.

RTT Round Trip Time

SDS-TWR Symmetric Double-Sided Two-Way Ranging

SDU Service Data Unit, the type of unit exchanged at the boundary with the upper layer (see Figure D.2).

SFD Start of Frame Delimiter

SHR Synchronisation HeadeR

SMA SubMiniature version A, a screw-type coaxial connector for DC-18 GHz.

SNR Signal to Noise Ratio, a power ratio, often expressed in dB.

SPI Serial Peripheral Interface Bus, a full duplex serial bus.

SRR Short Range Radio

SVM Support Vector Machine

SVN SubVersioN, a version control system.

TDoA Time Difference of Arrival

TE Telecommunication Engineering

ToA Time of Arrival

ToF Time of Flight

trit TeRnary digIT, valued −1, 0 or +1.

TWR Two-Way Ranging

Tx Transmitter

UART Universal Asyncronous Receiver/Transmitter

UDP Undetected Direct Path, [18].

ULP Ultra Low Power

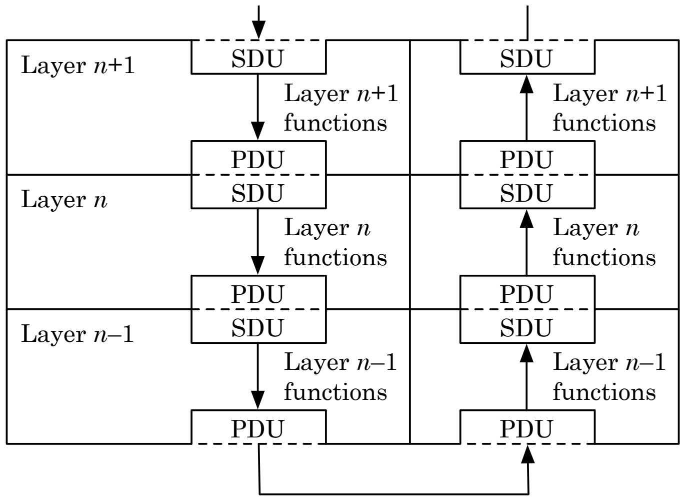


Figure D.2: The layer-relative definitions of SDU and PDU (after [86, Fig. 2]). What is an SDU from a certain layer's perspective, is a PDU from its upper layer's perspective.

USB Universal Serial Bus

UWB Ultra-Wideband, see also definitions.

VDP Voltage Delay Profile

VGA Variable Gain Amplifier

WSN Wireless Sensor Network