Acoustic TDOA Propagation Time Estimation

BSc Thesis

Jelle Tams & Nuriel Rozsa

Technische Universiteit Delft

TU Delft

Acoustic TDOA Propagation Time Estimation

BSc Thesis

by

Jelle Tams & Nuriel Rozsa

Student number:	4469704 (Jelle Tams)	
Student number:	4459040 (Nuriel Rozsa)	
Thesis committee:	Dr. ir. R. Heusdens,	TU Delft, supervisor
	Dr. ir. R. C. Hendriks	TU Delft, supervisor
	Dr. J. Martinez,	TU Delft, supervisor
	Dr. ir. J. J. A. Baselmans	TU Delft
	Dr. O. A. Krasnov	TU Delft

TU Delft

Abstract

In this report, the design of a subsystem within a localization system for the Bosch DICENTIS wireless conference system will be presented. The localization system will function by means of using acoustic Time Difference Of Arrival (TDOA) measurements to determine the location of each unit connected to the DICENTIS conference system. By unit, the system on the desk of each attendee in the conference, that contains a microphone and speaker is meant. The task of the subsystem presented is to estimate propagation times of transmitted signals between speakers of each unit in the conference system and the microphones of each unit.

The design choice for the type of localization method implemented is based on the gathered information from an initial literature study, the hardware specifications of the Bosch DICENTIS system and the demands for the localization system that were imposed by Bosch. The subsystem will function by transmitting a set of pseudo-random codes, modulated using a type of Frequency Shift Keying (FSK), where two On Off Keying (OOK) signals, modulated at different frequencies, are superimposed. The received and demodulated pseudo-random codes are then correlated with multiple different peak detectors that will correlate with multiple different sets of the transmitted string of pseudo-random codes to gain a higher robustness for the estimated propagation times and a higher accuracy for these estimates. Results show that the the use of multiple different sets of transmitted codes indeed improves the propagation time estimation. The overall system as presented, concerning accuracy and robustness, meets the requirements made by Bosch. However, in future work, optimalization of the system with regard to computation time is required.

Preface

Jelle Tams & Nuriel Rozsa
Delft, June 2018

This document is the final report for the Bachelor Graduation Project (BAP). The project described in this report is part of a larger project intended to provide Bosch with a localization system for the DICENTIS wireless conference system, but also serves as preliminary research for the potential implementation of said localization system into other conference systems designed by Bosch. The overarching BAP project commissioned by Bosch consists of three subgroups, of which dr.ir. Richard Heusdens, dr.ir. Richard Hendriks and dr. Jorge Martinez are the supervisors.

For this project, a DICENTIS conference system is made available by Bosch for use in this project. However, due to issues with getting access to the software stack of the DICENTIS conference system in a timely matter, the prototyping of the localization system is done in part with the Bosch DICENTIS wireless conference system and in part with a set of microphones and a sound card belonging to the Circuits And Systems department at the TU Delft. In this project, the microphones borrowed from the Circuits And Systems department were used as a replacement for the DICENTIS microphones in order to gain separate audio streams from each conference unit, given that these streams are not separable without gaining access to the software stack for the DICENTIS system. Although different microphones are used from the microphones in the DICENTIS system, the design choices are based around the hardware of the DICENTIS system, where for testing purposes, given that the used microphones have less limiting characteristics (such as frequency response and directivity) than the microphones used in the DICENTIS system, they are considered suitable replacements for prototyping purposes.

Contents

1 Introduction **1**
1.1 Project Context . . . 1
1.2 Project Definition . . . 1
1.3 System Description . . . 2
1.4 Propagation Time Estimation . . . 2
1.5 State Of The Art Analysis Of Ranging Techniques . . . 2
1.6 Document Structure . . . 3

2 Program Of Requirements **4**
2.1 Project Based Program Of Requirements . . . 4
2.2 Subgroup Based Program Of Requirements . . . 5

3 Design Of Propagation Time Estimator **7**
3.1 System Overview . . . 7
3.2 Creation of inter unit transmission signals . . . 8
3.2.1 Determination of Signal Coding Method . . . 8
3.2.2 Modulation method: Transmission Type and Modulation Technique . . . 11
3.2.3 Channel Generation . . . 13
3.2.4 Multiple Codes and Signal Repetition . . . 14
3.3 Filtering and Demodulation of Received Audio Signals . . . 15
3.3.1 Filter and Window Type . . . 15
3.3.2 Demodulation of Filtered Signals . . . 18
3.4 Signal Detection Method . . . 19
3.4.1 Matched Filter . . . 19
3.4.2 Peak Detection Method . . . 20
3.5 Estimation of Inter-Unit Propagation Times . . . 23
3.5.1 Determining Propagation Times . . . 23
3.5.2 Error Analysis . . . 23

4 Results **25**
4.1 Measurement Setup . . . 25
4.2 Filtering and Demodulation . . . 25
4.3 Signal Detection . . . 27
4.3.1 Matched Filter vs Gcc-Phat . . . 27
4.3.2 Implemented Matched Filter . . . 27
4.3.3 Peak Detection Method . . . 29
4.4 Determining Propagation Times . . . 29
4.4.1 Error Analysis . . . 30
4.5 Estimation Improvement by means of Multiple Estimators . . . 32
4.6 SNR of Total System . . . 33

5 Discussion and Conclusion **35**
5.1 Discussion . . . 35
5.2 Conclusion . . . 36

A Appendix **37**
A.1 Bandwidth of UniPolar and Polar Line Codes . . . 37
A.1.1 General PSD of binary signals . . . 37
A.1.2 Unipolar NRZ . . . 37
A.1.3 Polar NRZ . . . 39

A.2 Demodulating two OOK signals to original polar NRZ signal 39
A.3 Discrete Gcc-Phat Derivation 40

Bibliography **42**

1
Introduction

1.1. Project Context

In conference rooms such as at a UN headquarter, there is an integrated conference system in each conference room, where a microphone and speaker is available for each attendee. However, for many conferences, there is no integrated conference system available. Instead, a portable conference system is placed at the location where such a conference is held. Typically, when a conference is recorded in video, there are two camera views available; one where the entire conference is put in view and one which focuses on the speaker(s). In order for the camera to be able to focus on a speaker, the location of the speakers is pre-programmed such that once a speaker indicates he/she is about to speak, the camera automatically pans to that speaker. Currently the locations of each speaker is programmed by panning with the camera to each location where the speakers will be, and then setting those locations as the speaker locations. This is a reasonably fast solution if there is a small amount of speakers present at a conference, but during a large conference, this solution becomes less efficient. Because of this, Bosch requires a type of auto-locating system for their wireless conference system ”DICENTIS”.

1.2. Project Definition

This BAP group is tasked with designing an auto-locating system, such that it can be integrated into the Bosch DICENTIS hardware with minimal additional hardware requirements. Auto-localization can be done in various ways. In this case, the amount of available options is limited to the hardware specifications of the DICENTIS system. This means that either a localization method utilizing audio or WiFi needs to be designed [1]. Furthermore, depending on more specific specifications of the DICENTIS system, various localization techniques could be implemented such as Time Difference Of Arrival (TDOA), Time Of Arrival (TOA), Angle Of Arrival (AOA) or Received Signal Strength Indication (RSSI). Also, the type of transmitted codes to be detected between each DICENTIS unit of each attendee in a conference can be optimized based on the type of location the conference system is used in and the sending medium chosen to do localizations with. The fact that the DICENTIS system is synchronized also plays an important role in determining which kind of localization method is used.

In order to implement the system required by Bosch, the system design is split up into three sub-designs for each subgroup in group K, where a localization system will be implemented assuming a synchronized system. The following block diagram indicates the system as a whole with the data flow through each subsystem:

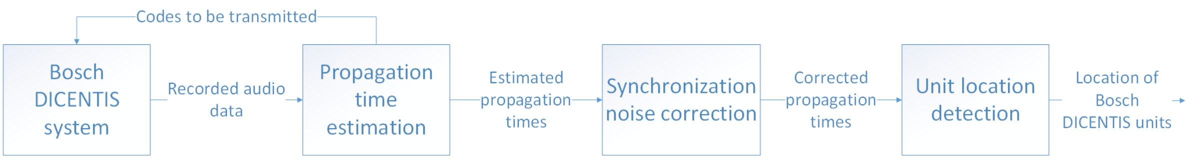


Figure 1.1: Total system showing the data flow through each subsystem.

In the propagation time estimation subgroup, assuming synchronized recorders, the propagation time between speakers and microphones of different units is estimated. In the synchronization noise correction subgroup, the offsets in the estimated propagation times due to the Bosch DICENTIS system not being fully synchronized are (partially) corrected. In the unit location detection subgroup, based on the measured and corrected estimated propagation times, the location of each Bosch DICENTIS unit is determined. In this report, the design of the propagation time estimation block will be covered.

1.3. System Description

The Bosch DICENTIS system consists of a set of units which can be placed in front of each speaker. Each unit contains a speaker and a microphone. Figure 1.2 shows three of these DICENTIS units.

In the back of Figure 1.2, a router is shown. This router acts as a hub for all connected units to the conference system. Audio data recorded at a specific unit is first transmitted to the router, after which this data can be relayed to all other units in the system via a WiFi protocol. The hub also has an analog input and output port. This allows for audio data to be relayed to each unit from a remote computer or audio data to be relayed to that remote computer which was recorded from the connected Bosch DICENTIS units. A remote computer can also be connected to the hub via a WiFi connection. Once a remote computer is connected to the hub, multiple things can be configured from a web interface like audio filtering, allowed speakers, volume of the speakers of each unit, etc. In order to avoid echoing, Bosch has reduced latencies in the system in part by optimizing the DICENTIS system such that each unit is synchronized within 100 μs.

Figure 1.2: Bosch DICENTIS wireless conference system.

1.4. Propagation Time Estimation

In this document, the propagation time estimator of the localization system will be covered. The challenge in this subsystem is to make the ranging between different DICENTIS units as accurate and as robust as possible, provided the hardware in each DICENTIS unit. As stated before, each DICENTIS unit contains a speaker, microphone and WiFi module. Therefore, a suitable ranging technique for either audio or WiFi needs to be chosen in coherence with the Bosch DICENTIS hardware. Once a suitable ranging technique is chosen, a method needs to be implemented that is most suitable for increasing the accuracy and robustness of the chosen technique. For this, types of transmitted signals and methods of parallelizing the localization technique should be considered. Once a suitable implementation of a chosen ranging technique is found, this method can be prototyped in order to confirm if it does in fact meet the system requirements.

1.5. State Of The Art Analysis Of Ranging Techniques

Given the available transmission methods with the Bosch DICENTIS system, various transmission methods using WiFi or audio were looked into. In source [2] and [3], multiple transmission media and techniques were compared. For WiFi as a sending medium, typically TDOA or TOA measurements are done, where it is required that the time stamps of the transmitting and receiving WLAN modules are adequately synchronized. For WiFi transmission, this can in practice prove difficult, due to the fact that the point of transmission does not have to be synchronous to the point where a time stamp is made. Even if two WiFi modules are perfectly synchronized, if the behaviour of a used WiFi module is not perfectly defined or controllable, this can lead to inaccuracies [4]. As covered in source [5] it is also possible to combine RSSI with TOA measurements for UWB signals to gain measurement accuracy. Given that a standard WLAN 802.11 module automatically does RSSI measurements [6][7], the addition of RSSI measurements in a localization system using WiFi would require no extra hardware, however using RSSI as a measurement method in and of itself does not produce a significant accuracy to be applicable for this project [7][8]. Although WiFi is not typically called a UWB signal, it is possible to significantly increase the effective available bandwidth of a signal sent via WiFi by means of frequency hopping [9][10]. Due to the increase in effective bandwidth, higher accuracy measurements can be done. By fingerprinting a region where localization is applied, the accuracy of localization in that region can also be increased. With this technique, by measuring signal strengths throughout a region where localization is required, better signal matching can be achieved. However, this does require the environment in which localization is required to be unchanging. This is something that is not the case in the application of localization for this project.

Concerning audio localization, source [2] and [3] mainly compared methods using ultrasonic signals. For this project, only audio signals in the audible range can be used, given that the speakers and microphones used in the Bosch DICENTIS system are optimized to an extent to only produce sound in those frequencies effectively. However, most of the techniques mentioned for ultrasonic acoustics also apply for acoustic signals in the audible range given that both signals share similar properties. Typically, acoustic measurements are also done via TDOA or TOA measurements with similar (or higher [11]) accuracies as the most accurate RF localization methods found (that don't use fingerprinting) without having as strict synchronization requirements. However, room temperature and pressure influence the measurement results [2] given that the speed of sound is reliant on both these parameters. A mentioned localization method in source [2] and [3] is the use of a Time Of Flight (TOF) measurement where a RF signal is used to synchronize a transmitter and receiver and the actual localization is done by means of a ultrasound signal, where millimeter accuracy can be achieved. The use of AOA is also mentioned, where an array of sensors is used to determine the angle between two sensors such that less sensors are required to determine the location of any single sensor. Because of this, this method is used with TDOA, TOA or TOF measurements. Here, synchronization using an RF signal can also be applied. Furthermore, source [2] mentions an achieved cm level accuracy using this method.

1.6. Document Structure

In this report, firstly the requirements imposed by Bosch will be covered for the localization system to be implemented as a whole. Also, the allowed assumptions by Bosch will be mentioned as well as the assumptions based on ambiguity in the situation under which the system will be used. Based on these requirements and assumptions, a set of requirements and assumptions will be presented for the design of the subsystem which is covered in this report. After the system requirements are defined, the design of the system will be covered, where the choices made and their reasoning is explained. After which, the results of the tests for the designed system will be presented, and based on that, a conclusion will be drawn on the viability of the designed system and recommendations will be made concerning the design itself and potential alternative options for the system that were not implemented in the current version.

2 Program Of Requirements

2.1. Project Based Program Of Requirements

The project based program of requirements are the requirements and assumptions that are imposed over the overlapping BAP group project. This concerns the system that can be presented to project supplier, which is Bosch.

Assumptions

When discussing the system requirements with the contact person at Bosch, there were a set of assumptions that were allowed to be made. Furthermore, our group also added a set of assumptions based on things that were not discussed at the meeting with the Bosch contact person. The made assumptions for the overarching project are as follows:

1. As indicated by the representative at Bosch when discussing the system requirements, there is line of sight between the Bosch DICENTIS units and between any given unit and the access point.
2. Minimal distance between units is 75 cm.
3. Microphones on the Bosch DICENTIS units can accurately capture audio from a distance of at least 30 m.
4. There is at least one unit per 15 m^2.
5. It is possible to upgrade the existing software on the Bosch DICENTIS system to accommodate the to be designed localization system functionalities.
6. It is assumed that the runtime of a MATLAB program run on system with recommended requirements is comparable to the runtime of the same program written for the Bosch DICENTIS system.

Mandatory requirements

These are criteria of which the system should always, at the very least, comply with. These can be subdivided into functional and non-functional requirements. Functional requirements being requirements of what the system must do, and non-functional requirements being attributes that the system must have.

1. **Non-Functional requirements**
   (a) The localization system must be within 10 cm accurate.
   (b) The localization system must be scalable up to 120 units.
   (c) A 3D localization method is necessary.
   (d) The localization should work in conference rooms with dimensions up to 30×30 m.
2. **Functional requirements**
   (a) The localization speed should have a maximum duration of 15 minutes for systems of more than 100 units.

(b) The localization speed should have a maximum duration of 5 minutes for systems of less than 20 units.

(c) The localization algorithm has to comply with the current Bosch DICENTIS conference system hardware characteristics.

(d) The localization procedure can only be initiated remote interface, so no one accidentally starts it during a conference.

(e) The system as a whole should not exceed the maximum allowed sound pressure level.

(f) The addition to the system of Bosch should not bypass any safety precautions taken by Bosch.

Trade-off requirements

These are criteria of which it is preferable to comply with as much as possible:

1. Minimize the number of manually configured anchor points.
2. Minimize the number of units required for the system to work.
3. The localization should work with the least possible additional hardware.
4. The localization speed should be as fast as possible.
5. The localization should be as accurate as possible.
6. A purely 2D localization method is usable besides a 3D localization method.

2.2. Subgroup Based Program Of Requirements

In each subgroup, a set of the project based program of requirements is relevant. Some requirements require stricter margins, given that multiple subgroups require a margin of error for certain requirements. For instance, all three subgroups in this project will influence the margin of error of the location estimation of each unit, this means that the margin of error for each subgroup concerning this requirement must be stricter than the project based requirement. Given that this subgroup is tasked with determining propagation times between speakers and microphones of the used Bosch DICENTIS speaker/microphone units, for this subgroup, the following assumptions and requirements are made.

Assumptions

When discussing the system requirements with the contact person at Bosch, there were a set of assumptions that were allowed to be made. Furthermore, our group also added a set of assumptions based on things that were not discussed at the meeting with the Bosch contact person. The made assumptions for the overarching project are as follows:

1. As indicated by the representative at Bosch when discussing the system requirements, there is line of sight between the Bosch DICENTIS units and between any given unit and the access point.
2. Microphones on the Bosch DICENTIS units can accurately capture audio from a distance of at least 30 m.
3. It is possible to upgrade the existing software on the Bosch DICENTIS system to accommodate the to be designed localization system functionalities.
4. It is assumed that the runtime of a MATLAB program run on system with recommended requirements is comparable to the runtime of the same program written for the Bosch DICENTIS system.
5. There is significantly little noise in the area where the Bosch DICENTIS system is used such that it is possible to create an audio signal with the DICENTIS speakers on each unit that has a significantly high signal energy to theoretically be able to be detected from a 30 m distance.

Mandatory requirements

1. **Non-Functional requirements**
   (a) The propagation time estimation system must be accurate enough such that the standard deviation of the error translates to 3cm.
   (b) The propagation time estimation system must be scalable up to 120 units.
   (c) The propagation time estimation system should work in conference rooms with dimensions up to 30 $\times$ 30 m.
2. **Functional requirements**
   (a) The propagation time estimation speed should have a maximum duration of 10 minutes for systems of more than 100 units.
   (b) The propagation time estimation speed should have a maximum duration of 3 minutes for systems of less than 20 units.
   (c) The propagation time estimation system has to comply with the current Bosch DICENTIS conference system hardware characteristics.
   (d) The propagation time estimation system as a whole should not exceed the maximum allowed sound pressure level.
   (e) The addition to the system of Bosch should not bypass any safety precautions taken by Bosch.

Trade-off requirements

1. The propagation time estimation should work with the least possible additional hardware.
2. The propagation time estimation speed should be as fast as possible.
3. The propagation time estimation should be as accurate as possible.

3

Design Of Propagation Time Estimator

3.1. System Overview

In this chapter, the design of the propagation time estimator is covered. The estimator functions by having all units in the Bosch DICENTIS system transmit a Frequency Shift Keyed (FSK) modulated repeated stream of unique pseudo-random codes. The acoustic signal of each unit is then received by each other unit in the network. After this, an audio signal from each unit is presented to a demodulator. The demodulated signal is then correlated by multiple peak detectors correlating a varying set of pseudo-random codes transmitted by each unit. Based on the sets of propagation times estimated by each peak detector, an analysis is done for each unit what the most likely audio propagation times are from that unit to all other units. In Figure 3.1, the top down view of the system in relation to the Bosch DICENTIS wireless conference system is given. In Figure 3.2, a block diagram is shown of the localization system.

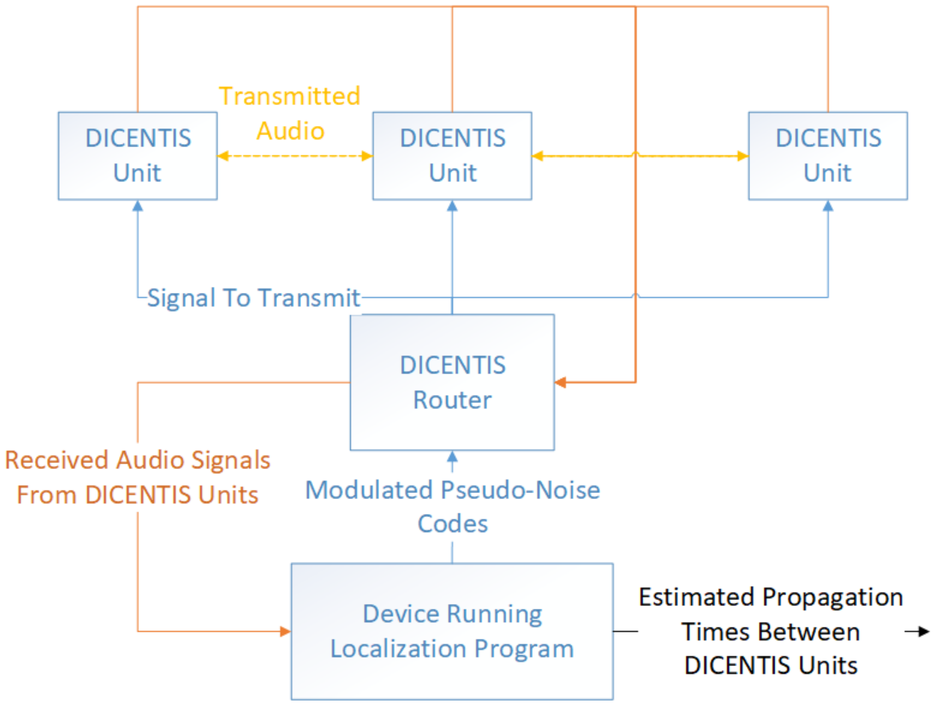


Figure 3.1: Block diagram of localization system in relation to the Bosch DICENTIS wireless conference system.

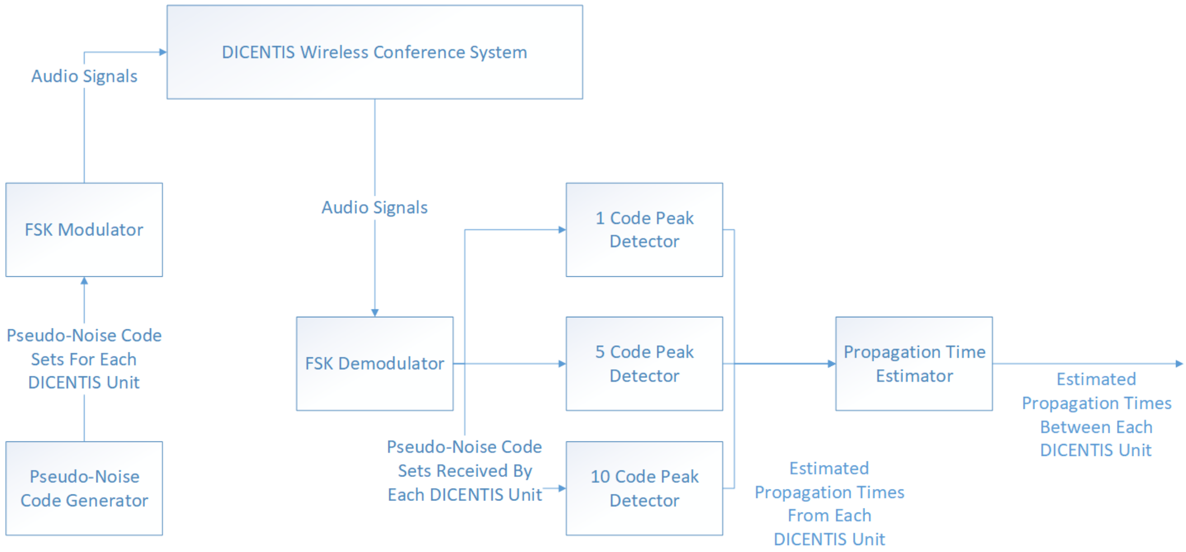


Figure 3.2: Block diagram of designed localization system.

3.2. Creation of inter unit transmission signals

To determine the propagation time between two different units of the Bosch DICENTIS system, a signal has to be sent between different units. Signals for localization are often coded signals, modulated with a certain modulation technique. Because different signals contain different properties, the signal with the most promising properties for the given problem has to be chosen.
This process is completed within two phases. Within the first stage, the different possibilities for signal codes will be compared to each other. Within the second phase, the different possible modulation techniques are compared to each other.

3.2.1. Determination of Signal Coding Method

To determine which type of signal code has the most promising properties, the possible code properties have to be listed and compared to the situation in which the signal will be used.
The following code properties are considered:

- Average Bit energy E_b
- Bandwidth
- Auto correlation
- Inter code cross correlation

The average bit energy is mostly determined by which method the binary symbols are represented in the to be used code. There are mainly two different forms, listed in source [12]; UniPolar (symbols are represented with 1's and 0's) and Polar (symbols are represented with 1's and -1's). Assuming that the expectancy of the different symbols are the same, the average bit energy is twice as large for Polar codes. Given that a twice as large average bit energy results in a twice as large signal energy, while the noise in the environment will be kept at the same level, the Signal to Noise Ratio (SNR) will be twice as high for the Polar representation of symbols.

The bandwidth of Unipolar and Polar codes is equal to the bit rate at which the codes will be sent. This is shown in appendix A.1 The bandwidth for linecodes is fixed, which means that for the final product a suitable bit rate has to be chosen in order to provide enough bandwidth for the signals. The fixed bandwidth of the Bosch DICENTIS system has to be taken into account while choosing the bit rate.

From appendix A.1 the autocorrelation function of the UniPolar and polar signal is also derived. The autocorrelation functions are defined in equations (3.1) and (3.2), where $R_{\text{UP}}(k)$ is the discrete autocorrelation function of a Unipolar signal and $R_{\text{P}}(k)$ is the discrete autocorrelation function of a polar signal. From these equations it can be observed that the autocorrelation function of a Polar signal is a Dirac (delta) function. This is a very useful property, because it guarantees that a signal is easily detected when correlated with itself. This is the main reason why Polar signals are preferred above Unipolar signals.

$$R_{\mathrm{UP}}(k) = \begin{cases} \frac{1}{2}, & k = 0 \\ \frac{1}{4}, & k! = 0 \end{cases} \tag{3.1}$$

$$R_{\mathrm{P}}(k) = \delta(k) \tag{3.2}$$

The most commonly used codes for localization algorithms in the literature are listed in Table 3.1 [12]. It shows next to polar line codes also a chirp code modulation.

Table 3.1: Different types of coding

Code Type	First Null Bandwidth	Autocorrelation	Numerical approach of the cross correlation
Pseudo-Noise code of MATLAB:	Bit rate R	$\delta(k)$	Close to zero
Maximum-Length sequence:	Bit rate R	$\delta(k)$	Shows smaller $\delta(t-\tau)$ functions
Gold codes:	bit rate R	$\delta(k)$	Uniform and bounded, close to zero
Linear Chirp:	freely chosen	$\phi_{ss}(t) = \sqrt{BT}\frac{\sin(\pi Bt(1-\frac{\lvert t \rvert}{T}))}{\pi BT}\cos(2\pi f_0 t)$ [13]	Close to zero, if different sequences differ in size

To determine what code is best to be used for the localization of the Bosch DICENTIS system, it is important to look towards the program of requirements. It is stated in section 2.2, there is a time limit of 10 minutes to locate all units of the system. Therefore, it is important that signals can be send simultaneously. Otherwise the recording time will increase proportionally with an increasing number of units. Because multiple signals/codes can be send simultaneously they may not interfere with each other.

The codes in Table 3.1 can be categorized within 2 sets. The first contains the pseudo-noise codes, which includes the pseudo-noise codes generated in MATLAB, the Maximum-Length sequence (ML-sequence) and the gold codes. The second set contains the chirp codes. The second set differs from the first set in that it is deterministically determined what the signal looks like. Furthermore, it is a combined coding and modulation method. Chirp codes also have the property that the required signal bandwidth can be freely chosen within the specifications of the Bosch DICENTIS system, whereas the baseband bandwidth of line codes is fixed with the bit rate. It should be noted that all different codes approach the Kronecker Delta function ($\delta(k)$), which makes all codes applicable for single code transfer. The difference comes when multiple codes are to be sent simultaneously.

Interference of linear chirp codes within the same bandwidth can be minimized in two different manners. The first would be to differ the chirp rate. Opposite chirp rates give an almost orthogonal set of chirp codes, where the crosscorrelation would be close to zero. In other cases, the crosscorrelation is bounded, but not uniform as there is to some extent overlap in the different chirp signals. The second method to minimize the interference is to change the chirp duration. This is not preferred, because not every chirp signal will receive the same power, but also the total measurement time can increase. Because the crosscorrelation does not always tend to go to zero and because time measurements can increase in duration, chirp codes are not used.

Pseudo-random noise codes are often used for signal detection, because of their auto- and crosscorrelation properties. Random codes generated from gaussian noise have equal likelihood of occurrence of each binary signal, making the autocorrelation function equation (3.2), which is ideal for detecting the signals with correlating techniques. The crosscorrelation of two different random generated codes can be defined in equation (3.3). Within this equation it is known that there are four possibilities for the product $(a_n a_k)$ that all occur with the same probability, see Table 3.2.

$$R(k) = \sum_{i=1}^{4} (a_n a_k) P_i = 0 \tag{3.3}$$

Table 3.2: Possibilities of inter signal product $(a_n a_k)$

Possible symbol value	Signal A: 1	Signal A: -1
Signal B: 1	1	-1
Signal B: -1	-1	1

Pseudo-random noise codes are easily generated in MATLAB using a signed randn function. Within MATLAB the ziggurat method is used to create a pseudo random normally distributed sequence [14]. Sequences made with this method have a repetition length of about 2^{64} numbers, which is more than enough to guarantee that no repetition within used sequences will occur. The ziggurat method is one of the fastest methods of it's kind, but also is very accurate as shown in [14].

The crosscorrelation of ML-sequences shows more than a single Kronecker Delta function, which makes it increasingly difficult to detect the correct arrival time of the correct ML-sequence using correlating methods. Multiple peaks arise due to the small repetition period. Time aliasing is the result. The problem is solved by using gold codes, which are paired ML-sequences in such a way that the same properties of the ML-sequence are kept, but that the cross-correlation function approaches a uniform line close to zero, as presented in [15]. This would make the gold codes an equally viable option as the pseudo-noise codes generated by the signed randn function of MATLAB.

The main difference between ML-sequences and gold codes on one side and the pseudo-noise codes generated by MATLAB on the other side is that ML-sequences and gold codes guarantee that there is one more 1 than there are 0's, which is important for the signal power when it is coded with a Unipolar line code [16]. Since Polar signals are chosen, the signals power will always be the same, for any random code generated. It is for this reason that there is no benefit to using gold codes above using pseudo-random generated codes. Because pseudo-random generated codes do have (3.2) as the autocorrelation function, which is ideal for signal detection using correlation methods, and because there is no benefit to using gold codes, pseudo-random codes, generated by the randn function of MATLAB are used.

To achieve minimal computation and measuring time, small signals are preferred. This means that it is preferred to have pseudo-random noise codes with the least amount of bits possible. However, because the correlation function will be numerically computed, the computed correlation will only converge to the analytical correlation function in the limit of increasing amount of bits. It is for this reason that the length of pseudo-random codes was chosen to be 127 bits, which is as long as an ML-sequence generated with 7 bits. Simulations show that 127 bits are the trade-off between speed and accurate representation of the estimated correlation functions, as is shown in Figures 3.3 and 3.4.

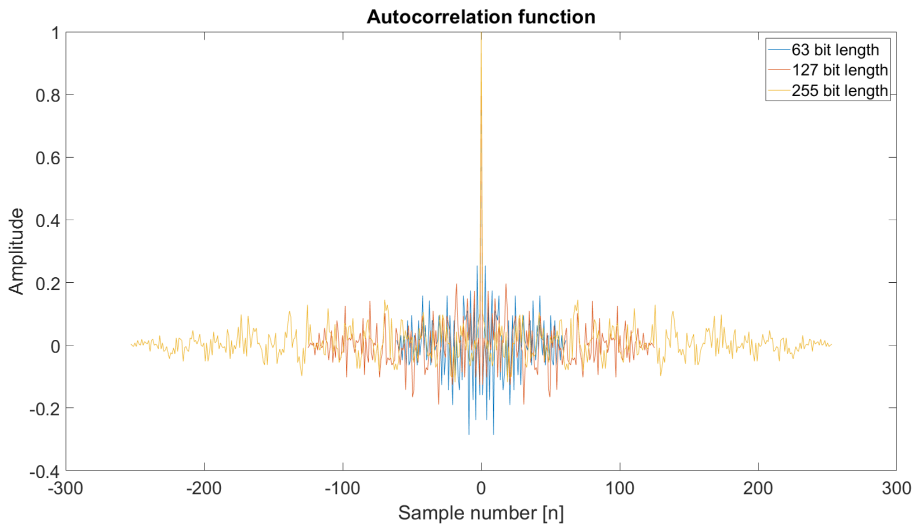


Figure 3.3: Numerical Autocorrelation for pseudo random noise codes

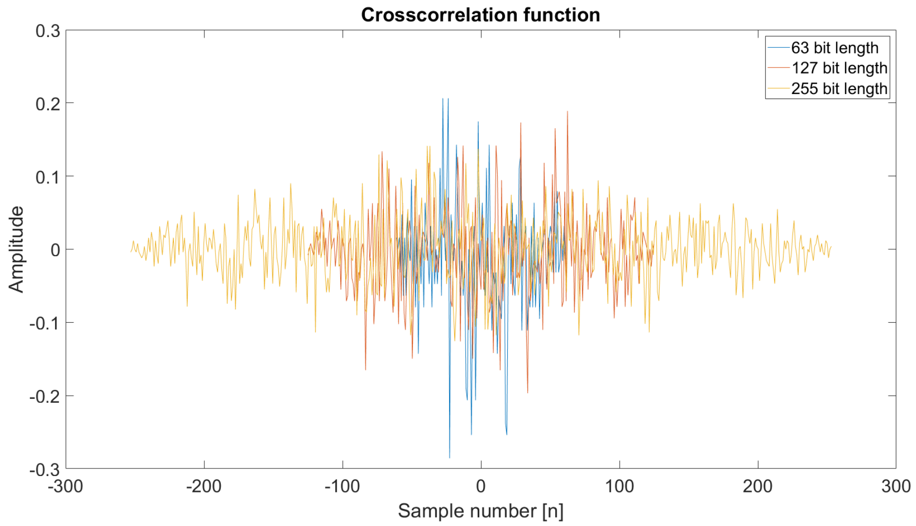


Figure 3.4: Numerical Crossorrelation for pseudo random noise codes

3.2.2. Modulation method: Transmission Type and Modulation Technique

Digital bit streams can not directly be transmitted and must therefore first be modulated onto a suitable carrier for a given sending medium. In this case, the medium to be considered is air in an indoor environment with a maximum size of 30 by 30 meters.

To obtain the best modulation method for the medium in which the Bosch DICENTIS system will work, first the different types transmission signals will be compared. After which, the different modulation techniques applicable for the codes to be transmitted will be discussed.

Transmission Type

Many different transmission signal types exist, such as audio, radio and optical signals. Within the program of requirements 2, it is stated that the localization system should work with as little additional hardware as possible. This means that only the different possible transmission types already available by the Bosch DICENTIS system will be covered.

The Bosch DICENTIS system provides for two different transmission signal types; audio signals and radio frequency (RF) signals. Audio signals can be sent from any DICENTIS unit speaker and can be detected by any other unit microphone. Concerning RF signals, the Bosch DICENTIS system uses a standard WLAN IEEE 802.11 WIFI module [1]. The way in which different units of the system communicate with each other is by means of a centralized router. This means that there is no direct WIFI communication between different units. This is sub-optimal for determining the direct propagation time between different units.

Apart from the indirect RF communication between different units, assuming that this problem can be resolved on the behalf of Bosch, the audio signals and RF signals are compared to each other in terms of performance.
In Figure 3.5 it can be observed that WLAN typically has an indoor resolution of larger than a meter. This is due to the non deterministic behaviour of WIFI signals when there is little control over the used WIFI modules. WIFI uses carrier sensing with a random backoff time. This makes the timing of the sending times of WIFI signals difficult when there is little control over the behaviour of the used WIFI modules, which is the case with the Bosch DICENTIS system. Small errors in timing can result in large distance errors due to the propagation speed of WIFI signals. Timing errors with audio signals result in smaller distance estimation errors, due to the fact that the speed of sound is approximately a million times slower than the speed of light, resulting in a million times smaller distance errors with the same timing error. In the literature, accurate distance measurements were conducted with audio signals. Source [17] shows that with acoustically modulated line codes, an accuracy of up to 2 cm can be achieved. Wide band chirp signals could even result in distance errors of less than 1 cm.[11]. Given that line codes are preferred over chirp signals because of their property to be able to send different signals in parallel, and that 2 cm is well within the accuracy requirements as stated in section 2, it is chosen to send line

codes with acoustic signals.

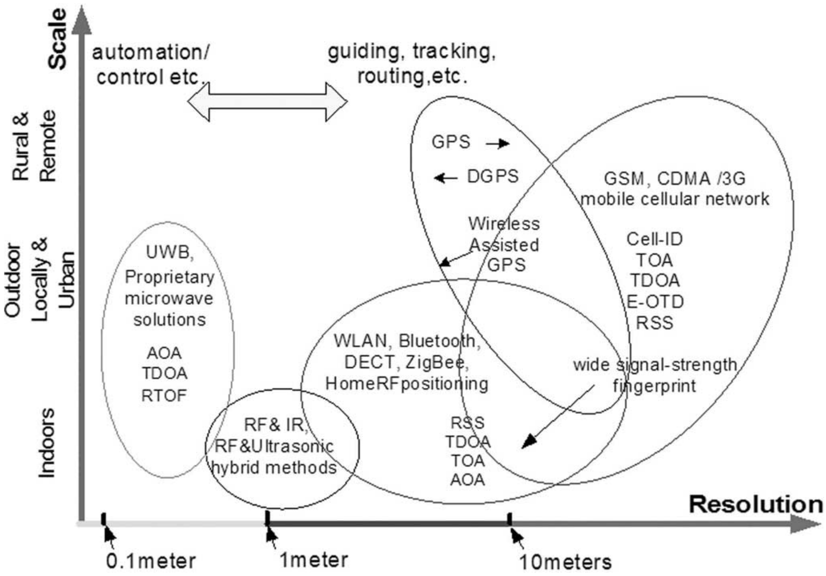


Figure 3.5: Outline of current wireless-based positioning systems[3]

Modulation Technique

With the transmission signal type chosen to be audio signals, the translation of line codes to audio signals still has to be made. This is done by means of modulating the line code onto an acoustic signal. Many types of modulation techniques exist of which some of the most common will be covered.

The book "Digital and Analog Communications" [12] mainly mentions three modulation techniques for modulating binary signals. These are:

- On Off Keying (OOK): A unipolar signal $m(t)$ is modulated on a single carrier. The modulated signal is: $s(t) = m(t)\cos(\omega_c t)$, where $s(t)$ is the modulated signal, $m(t)$ a unipolar signal and ω_c the angular carrier frequency.

- Frequency Shift Keying (FSK). There exist two different forms of FSK, namely continuous and discontinous phase FSK. Continuous phase FSK can be generated by Voltage Controlled Oscillators (VCO). Since the audio signal will be digitally created, only discontinous FSK will be covered. The modulated signal is of the form of equation (3.4) and can be seen as the linear superposition of two different OOK modulated unipolar signals. It should be noted that each symbol of the polar signal is modulated on a different carrier, resulting in more power in the signal, but using more bandwidth.

$$s(t) = \begin{cases} \cos(\omega_1 t + \theta_1), & \text{when a 1 is sent} \\ \cos(\omega_2 t + \theta_2), & \text{when a -1 is sent} \end{cases} \tag{3.4}$$

- Binary Phase Shift Keying (BPSK): A polar signal can be modulated on a single carrier using BPSK. The different symbols of the polar signal are modulated 180 degrees ($\Delta\theta$) out of phase. Equation (3.5) shows a polar signal modulated using BPSK on a single carrier.

$$s(t) = \cos(\omega_c t + \frac{1}{2}\Delta\theta m(t)) \tag{3.5}$$

Table 3.3 shows some properties of the mentioned modulation techniques. Only non coherent detection methods are compared. Due to the fact that, due to parallelism, multiple signals with different phases can be simultaneously received, only non coherent detectors are to be used. When no parallelism of signals would be applied, coherent detectors would result in smaller bit error rates, making BPSK the preferred choice. There is one case, mentioned in [18] where BPSK is non coherently detected. However, problems occur when multiple signals are received simultaneously, making it hard to track the correct phase of the signal, especially when different received signals are out of phase.

Table 3.3: Properties of Modulation Techniques

Modulation technique	Bandwidth	Non coherent detection	Average signal power	Bit error rate
OOK	2*bit rate	++	P_{OOK}	$\frac{1}{2}e^{-(\frac{1}{2}SNR)}, SNR > \frac{1}{4}$
FSK	4*bit rate	++	$2 * P_{\text{OOK}}$	$\frac{1}{2}e^{-(\frac{1}{2}SNR)}$
BPSK	2*bit rate	–	$2 * P_{\text{OOK}}$	Not applicable

Because BPSK signals are sensitive to multiple received signals, it is not a viable option when parallelism is an important aspect of the design. As mentioned in section 3.2.1, polar signals are preferred above unipolar signals. This leaves us with the choice of the use of FSK, because a single OOK signal typically modulates a unipolar signal. The FSK modulation will be created by the superposition of two OOK signals.

3.2.3. Channel Generation

A channel is a certain amount of bandwidth reserved for the transmission of a signal around a given carrier frequency. As long as two channels do not overlap, there exists a complete orthogonality between those two channels. Because of this, given a certain required bandwidth for the transmission of a signal and the available bandwidth, it is best to fit as many channels into the available bandwidth as possible, such that pseudo-random code sets which are transmitted can be orthogonalized more.

In order to determine the available bandwidth, the frequency response of both the Bosch DICENTIS microphones and speakers need to be considered. Given that only the specification sheet of the DICENTIS microphones was available, an optimal bandwidth will be determined based on the frequency response and directivity of the DICENTIS microphone. After which, an assumption will be made on the frequency response and directivity of the DICENTIS speaker in order to determine if more band may be used.

When analyzing the frequency response and directivity of the DICENTIS microphone from it's specification sheet, which is shown in Figure 3.6

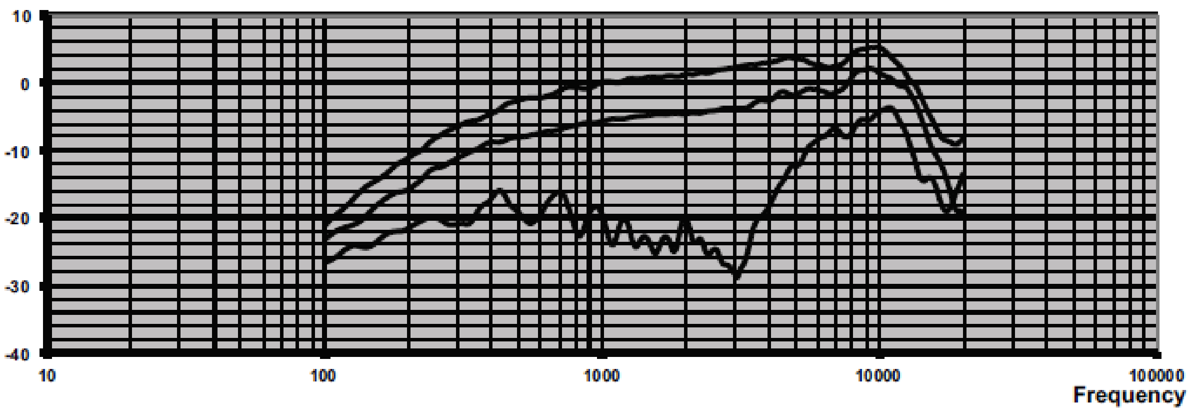


Figure 3.6: Frequency response and directivity of Bosch DICENTIS microphone from specification sheet [19] with an Opening angle: 0, 90 and 180 degrees.

it is found that the least directivity exists between 5 kHz and 12.5 kHz with a maximum directivity of 10 dB and a fluctuation in frequency response of at most 6 dB. Under 5 kHz, the directivity rapidly increases to at most 35 dB, however the frequency response directly in front of the microphone shows a slow decrease of at most -3 dB going down to 1 kHz with respect to the 5 kHz point. Above 12.5 kHz, the frequency response rapidly decreases with at most -20 dB, but the directivity remains relatively constant in this higher frequency range. With this, the best available band is between 5 kHz and 12.5 kHz, providing 7.5 kHz band.

Given that a higher bitrate increases the bandwidth requirements of a signal, while it is desirable to decrease the transmission time of a pseudo-random code as much as possible to avoid issues with echos overlapping, a trade-off has to be made for the used bitrate. Source [17] recommends a transmission time for a type of pseudo-random code of less than 50 ms as a trade-off between reverberation and noise cancelling. Given that in the application of this system, it is assumed that reverberation is a larger issue, the ideal transmission time is assumed to be 25 ms. In appendix A.1 it is stated that the baseband first null bandwidth is equal to the bit rate for a polar signal, which means that it is twice the bit rate for an OOK signal. In order fully receive the main lobe of the transmitted signal at the recommended bitrate to avoid issues with echos, the required bandwidth is two times the amount of bits sent in one code divided by the total recommended transmission time times the amount of OOK signals sent, which is 2*2*127/0.025 = 20160 Hz. This is more bandwidth than is plausibly available. Given that the ends of the main lobe of a received signal contain less energy than the main belly, it was chosen to receive 3/4 of the main lobe which is explained in more detail in section 3.3, which means that the ideal required bandwidth then becomes 2*1.5*127/0.025 = 15240 Hz, which is still significantly more than available. It was chosen to decrease the bit frequency of the transmitted OOK codes to 3000 Hz, requiring 2*1.5*3000 = 9 kHz band, which is a more plausible bandwidth requirement.

Now considering the new required band. Based on the requirements for the filters to adequately separate the channels of both OOK signals, the choice was made to expand the used band to 10 kHz between 2.5 kHz and 12.5 kHz. The reasons for this design choice will be explained in more detail in section 3.3. This means that the used centre frequencies are 5 kHz and 10 kHz allowing for a maximum bandwidth of 5 kHz for each OOK signal.

3.2.4. Multiple Codes and Signal Repetition

So far in this section, the generation and modulation of a single code is described. This code to be transmitted is a 127-bit pseudo-noise polar NRZ code, which transmitted by considering this code as two transposed OOK signals and modulating each OOK signal with a different carrier frequency. However, there are multiple reasons to transmit a stream of different generated pseudo-noise code instead of one for every given unit.

The first reason is that when correlating the demodulated received audio data, a larger set of estimations can be done, thus the estimated propagation delays will converge. The second reason for sending multiple different codes, is due to the fact that multiple peak detectors can then be implemented which can make uncorrelated estimations towards each other on the propagation delays based on correlating one or multiple pseudo-noise codes. The benefit to this is explained in more detail in section 3.5.2. In order to ensure the first benefit for all peak detectors, not only are multiple different codes transmitted, but the stream of different codes is also repeated multiple times.

When deciding how many different codes to transmit and how many times to repeat the transmission of the stream of different codes, a trade-off arises between the following things:

- The total transmission time.
- The total amount of convergence of the set of estimated propagation times.
- The increased energy in the signal for the peak detector correlating multiple transmitted codes, increasing the chance of finding a correct peak under low reverberant conditions.

Given that finding an optimum between these factors is difficult to calculate, it is imposed that the transmitted signal will be a stream of 10 pseudo-random codes repeated three times. Furthermore, three peak detectors will be implemented to correlate single codes, sets of 5 codes and sets of 10 codes. Figure 3.7 gives a representation of the transmitted signal with indicated sets for the code set correlators:

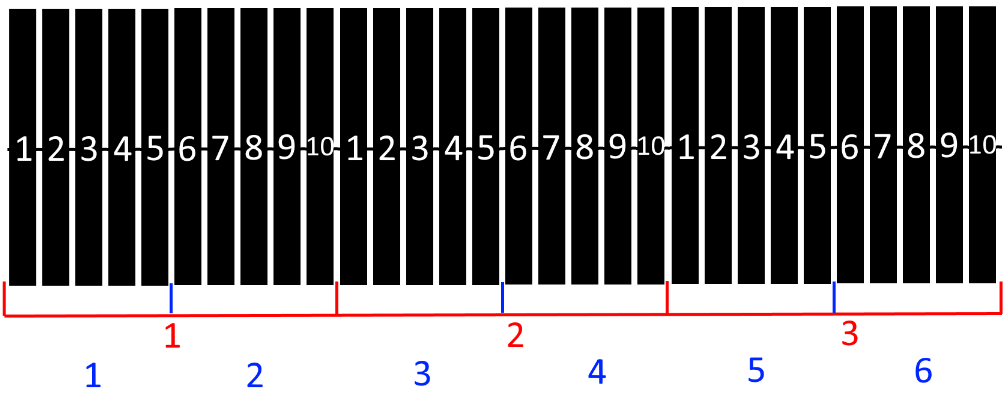


Figure 3.7: Visual representation of generated code with red and blue indicating the correlation sets of two different peak detectors.

In Figure 3.7, it is made clear that the first peak detector will produce a set of 30 propagation time estimations, the second peak detector will produce a set of 6 propagation time estimations and the third peak detector will produce a set of 3 propagation time estimations. Because each implemented propagation time estimator will produce a data-set of at least 3, each estimator has an error correcting capability of at least 1 if a significantly deviating error occurs, which is the reason that a repetition rate of three was chosen. A data stream of 10 random codes was chosen in a more heuristic fashion based on what produced reasonable results when developing the system. The reason that three propagation time estimators were chosen is for a similar reason that a code stream repetition rate of three is implemented. In the case that one estimator produces radically deviating propagation times, the results from the other estimators can outweigh the deviating estimator. In section 3.5.2, the use of these three estimators to estimate a most likely propagation time will be explained in more detail.

3.3. Filtering and Demodulation of Received Audio Signals

In order to separate and demodulate the sent signals from each used channel, a set of filters are required. The filtering can be done by multiple types of filters. To convert the signal back to a baseband signal, another signal conversion step is required. In this section, the design choices concerning these two points will be covered.

3.3.1. Filter and Window Type

Firstly, a choice is required for which type of filter to be used. Here, IIR and FIR filters can be distinguished. There is a critical benefit to using a FIR over an IIR filter. This benefit is the linear phase of a FIR filter, which a IIR filter does not possess [20]. It should be mentioned that it is possible to create local linear phase IIR filters, however these filters are sensitive to truncation errors and can therefore become unstable [21]. Given that it is unknown what kind of truncation errors the Bosch DICENTIS system introduces, it is undefined how stable an IIR filter is for the given application. Because of this, it was chosen to use a FIR filter.

Given that the received signal is a superposition of two OOK signals, a type of windowing is required to adequately separate the two frequencies of which the two OOK signals consist. These two filtered signals can then be superpositioned onto each other after filtering. For choosing the correct window type, three criteria need to be considered:

1. The main lobe in the time domain of the filter transfer function should be as thin as possible.
2. The ratio between the energy in the main lobe in the time domain and the side lobes after the first null-point should be as high as possible.
3. The filter time should be as low as possible for given filter specifications.

The first design criterium is important due to the fact that short-delay echos should be able to be separated from the direct-path signal as much as possible. Given that a wide main lobe of the filter can have almost overlapping signals appear as a single signal due to a spreading effect occurring, by using a filter with a thin main lobe, this kind of overlapping can be prevented more effectively.

The second design criterium is based on the fact that a finite length filter in the frequency domain has a infinite length in the time domain. In order to fully represent the filter transfer function in the time domain, infinite samples would be required. Given that the transfer function will not have an infinite length, it is best to have most energy of the transfer function of the filter in the time domain concentrated within the chosen window in the frequency domain. Different window types influence the trade-off between this property and for instance the 'sharpness' of the filter.

Because a higher order filter requires more computations when computing the convolution between an input signal and the filter transfer function than a filter with a lower order (due to the fact that each filter order introduces an extra delay which causes the filter transfer function to increase in length), there is a certain limit to the order which a filter may have. A design criterium is, that the maximum time for localizing all microphones (at most 120) is 15 minutes. Assuming 10 minutes for computing all distances on a single computing device and imposing that the computation time of a single filter may at most be 5 percent of the total available computation time, with at least 2 filters required to recover 1 channel, the maximum computation time for one filter becomes the total time in seconds times 0.05 divided by two times the maximum amount of units (two filters per audio stream from one unit), which is 60*10*0.05/(120*2) = 0.125 s. This computation time can vary based on the implementation of the filter in code, the programming language, computer, OS and other running applications. Because for prototyping purposes, MATLAB is used to develop the distance determining system, an average computation time for the MATLAB 'filter' function (when using the coefficients of the designed filter) will be used as reference after the other design steps are taken.

To design the required filters for a used channel, it is important to define the cut-off frequencies for the two windows of each channel. For this it is important to define the required frequency for each carrier, the frequency band available and the minimum amount of separation required to adequately be able to separate the two carriers within one channel. As was mentioned in section 3.2.3, the effective best band available is 7.5 kHz with frequencies between 5 kHz and 12.5 kHz. To determine the required band of one carrier, the required band for a modulated OOK signal can be considered, given that either a OOK signal for the -1's or 1's of the transmitted code need to be recovered. As is shown in appendix A.1, the band required for the main lobe of a unipolar NRZ signal is 2R, where R is the bitrate. As stated before, the chosen bitrate for the transmitted signal is 3 kHz. This means that, in order to fully window the main lobe of the received signal, the required band is 6 kHz. Given that two windows are required for one channel, it was chosen to use 2.5 kHz extra band between 2.5 kHz and 5 kHz. In this extra band, the directivity of the Bosch DICENTIS microphones increases with 30 dB. This is somewhat compensated due to the fact that the Bosch DICENTIS speaker can produce more power in the vocal audio range, given that the speakers are optimized to produce vocal audio. Although not optimal, this is deemed better than decreasing the bitrate, because as mentioned before, the estimated ideal transmission time is 25 ms. With a transmission time of 127*16/48000 = 42.3 ms, it is undesirable to decrease the bitrate further.

With a band betweeen 2.5 kHz and 12.5 kHz available, the windows of both filters would be centered around 5 kHz and 10 kHz with a window size of 4.5 kHz where 250 Hz band is used on both sides of each window as cut-off band. Given the equation for the power spectral density for a unipolar NRZ signal as described in appendix A.8, considering the baseband signal, the fraction of power in a frequency range $2f_b$ is:

$$P_{frac} = \frac{\int_{-f_b}^{f_b} \left(\frac{sin(\pi f T_b)}{\pi f T_b}\right)^2 \left[1 + \frac{1}{T_b}\delta(f)\right] df}{\int_{-\infty}^{\infty} \left(\frac{sin(\pi f T_b)}{\pi f T_b}\right)^2 \left[1 + \frac{1}{T_b}\delta(f)\right] df} \tag{3.6}$$

where $\delta(f)$ is the Dirac delta function. Numerically solving this equation using WolframAlpha [22] with $T_b = 16/48000 = 1/3000$ for $f_b = 2250$ and $f_b = 3000$ results in $P_{frac} = 0.9444$ and $P_{frac} = 0.9514$ respectively. The difference with respect to the total power is 0.9514-0.9444 = 0.007 which is deemed an acceptable relative loss to justify the chosen window.

The next value to be determined for the chosen windows is power decrease P_s at stopband frequency f_s. For this, the worst case power difference between two measurements and the two frequencies of a channel should be determined. This power difference is expressed with the following function:

$$P_{diff} = P_d + P_{fm} + P_{fs} + P_{dm} + P_{ds} + P_{bs} \text{ [dBW]} \tag{3.7}$$

where P_d is the power difference caused by the distance between two speakers with respect to a microphone, P_{fm} is the power difference caused by the difference in sensitivity of the used microphones for the different modulated frequencies, P_{fs} is the power difference caused by the difference in power that the used speakers can deliver for the different modulated frequencies, P_{dm} is the directivity of the used microphones, P_{ds} is the directivity of the used speakers and P_{bs} is the difference in power that the speakers can emit audio with due to a different battery strength. With the project assumption that the maximum distance between two units is 30 m, $P_d = 10log(30^2) \approx$ 30 dBW. In Figure 3.6, $P_{fm} + P_{dm}$ = 35 dBW was found for a maximum frequency-directivity-power difference between 3 kHz and 10 kHz with 180 degree difference in direction. P_{fs}, P_ds and P_{bs} are unknown, because the frequency and directivity characteristics of the speaker of the Bosch DICENTIS system are unknown and because it is unknown how the battery charge affects the output power of the speaker. Because of this, these power differences are assumed based on the knowledge that the DICENTIS system was designed to optimize the relaying of the vocal audio range. It is also assumed that during the design phase of the Bosch DICENTIS system that the directivity of the microphone is more optimized than the speakers. With these two assumptions, it is assumed that 6 dB more power can be relayed around the 5 kHz centre frequency than the 10 kHz centre frequency used to modulate a pseudo-random code in a channel. Given that when determining $P_{fm} + P_{dm}$, at 5 kHz the least power could be received at a 180 degree angle from the microphone, P_{fs} = -6 dBW. Furthermore, it is assumed that the maximum directivity of the Bosch DICENTIS speaker is less than the directivity of the DICENTIS microphone, thus P_{dm} is less than 35 dBW. Lastly, it is assumed that the emitted power from a Bosch DICENTIS speaker will deviate at most 3 dBW due to different battery strengths. With all power differences determined, the worst case power difference between two different signals is P_{diff} = 30 + 35 - 6 + 35 + 3 = 97 dBW. Taking a margin of error of 3 dBW, P_s should be more than -100 dBW.

With all filter requirements defined, a suitable window type can be chosen based on the filter requirements and the defined criteria for the window type. Using the 'filterDesigner' tool in MATLAB, the most suitable window type found is the Kaiserord filter window type. Using this window type, the desired bandpass filters around a centre frequency of 5 kHz and 10 kHz were designed with a passband of 4.5 kHz, a stopband of 250 Hz on each side and a stopfrequency f_s power decrease P_s = -100 dBW. This requires a filter order of 1232. The typical shape of the filters is shown in Figure 3.8:

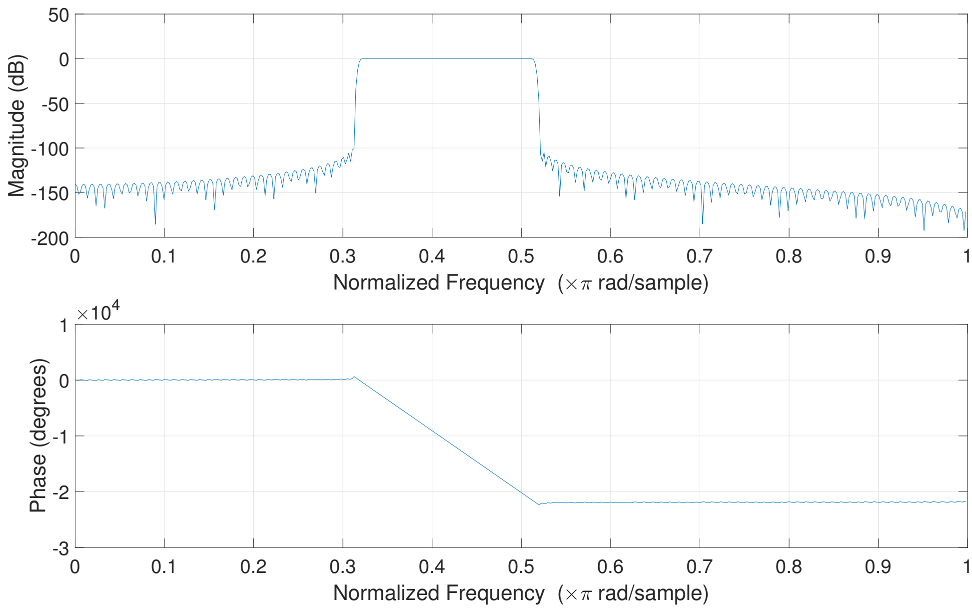


Figure 3.8: Frequency and phase response of the filter for a signal with a 10 kHz carrier frequency.

A mat file was then generated containing the coefficients for the designed filters. The filter speed was then tested by iterating the filter function 1000 times using the generated filter coefficients and a generated input stream of 300,000 samples with values equal to 1 at each sample by measuring the time difference of the time stamp before the filter function and after the filter function using the tic toc time measuring function in MATLAB. The average found time difference is 1.0e-04 s. This is significantly below the required of 0.125 s, therefore it can be stated that the designed filters meet all imposed demands.

3.3.2. Demodulation of Filtered Signals

Because a better auto-correlation can be gained from de-modulating the filtered audio signals, a de-modulation step is required that recovers a constant-amplitude signal irrespective of the phase in which the modulated signal arrives. In order to convert the two filtered OOK back into one polar NRZ signal, first the filtered modulated OOK signals should be described:

$$s_{\text{OOKm}}(t) = s_{\text{OOK}}(t)\cos\left(2\pi f_c t + \theta\right) \tag{3.8}$$

Here, θ is an unknown value and $s_{\text{OOKm}}(t)$ is the modulated $s_{\text{OOK}}(t)$ signal. There are multiple methods to convert this signal back to $As_{OOK}(t)$. One method is to multiply the signal with a complex number to regain a dc centre frequency in the signal and then filter all ac centre frequencies. Given that it is unknown how powerfull the computation unit is that will be used for the Bosch DICENTIS system is and it is unknown if the used software supports complex computations, it was decided that no complex computations should be used. In this case, the original polar NRZ signal can be recovered using Algorithm 1

Algorithm 1 Demodulation algorithm

procedure DEMODULATION($s_{\text{OOK1}}(n), s_{\text{OOK-1}}(n)$) ▷ Demodulation of 2 OOK signals
    for $i = 1$:length(s_{OOK}) **do** ▷ Multiply OOK signals with Sines and Cosines

        $s_1(i) = (s_{\text{OOK1}}(i)\cos(\omega_{c1}(i)))^2 + (s_{\text{OOK1}}(i)\sin(\omega_{c1}(i)))^2$ ▷ Square results and add
        $s_{-1}(i) = (s_{\text{OOK-1}}(i)\cos(\omega_{c2}(i)))^2 + (s_{\text{OOK-1}}(i)\sin(\omega_{c2}(i)))^2$

    $s_{\text{polar}} = s_1 - s_{-1}$ ▷ Subtract to gain Polar signal

    function LOWPASSFILTER(s_{polar})
        $Result$ = Output filter ▷ Subtracted OOK signals are filtered

    return $Result$ ▷ Return demodulated signal

The proof for this step plan is given in appendix A.2. In order to filter the non-baseband signal, a low-pass filter is used with equivalent characteristics as the filters covered in the previous subsection. This means that the cut-off frequency used is 2250 Hz and the stop-band frequency f_s is 2500 Hz. The following block diagram describes the total demodulation process:

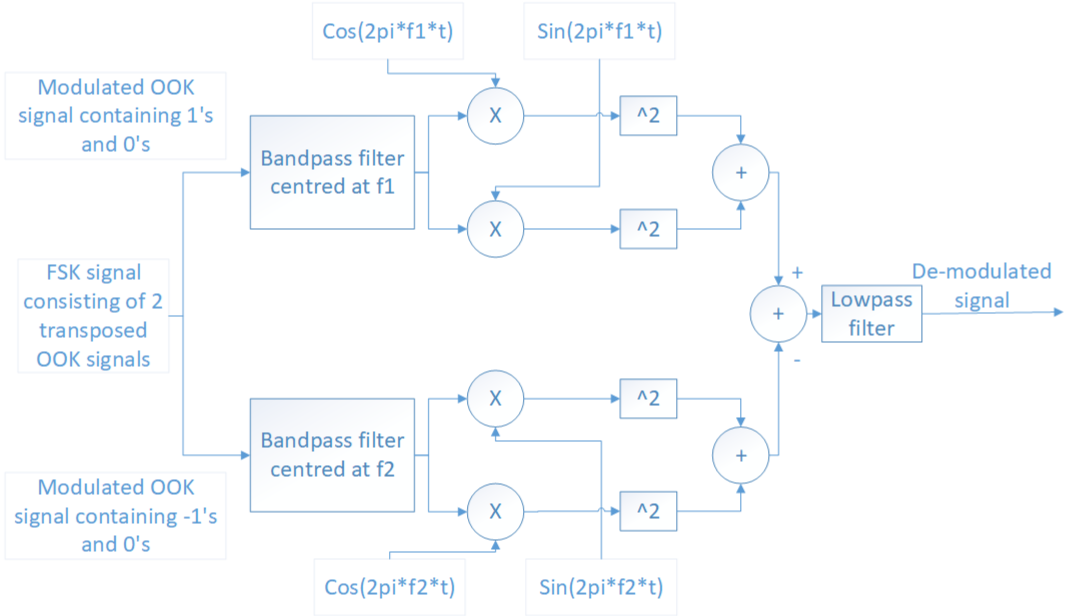


Figure 3.9: Block diagram of the designed demodulator.

3.4. Signal Detection Method

With the received signals demodulated, the line codes sent still have to be detected from the demodulated signals. Knowing that multiple signals can be simultaneously sent and can interfere with each other, a method has to be derived to detect the correct line code within the demodulated signal. There are multiple ways of detecting signals within the literature. Commonly used methods are by detecting signal energy [23] and by matched filtering. With the energy detection method it is assumed that the received signal power is larger than the received noise power. However, when multiple signals are received simultaneously, it is very hard to distinguish the different signals when using signal detection based on signal energy.
Matched filtering gives the opportunity to search for a reference signal. This is why multiple signals can be distinguished from any received signal and is the reason why matched filter detection is in this design preferred over signal strength detection. When a priori the to be detected signal is known, which is the case within this design, then the matched filter is the optimal detection method [24].
Within the following sections, such a matched filter detection method will be derived.

3.4.1. Matched Filter

Matched filters are linearly designed filters with the purpose to maximize signal strength, while minimizing the influence of noise on the signal [12]. It should be noted that a matched filter does not preserve the original shape of the signal, but transforms it to maximize the signal to noise ratio. The matched filter in case of gaussian noise is stated in equation (3.9), where C is a real positive constant (often used for normalization of the result), s(t) the known waveform of the signal to be detected and t_0 the time at which the output signal will result in a peak (often chosen to be 0).

$$h(t) = Cs(t_o - t) \tag{3.9}$$

Knowing that all line codes sent are pseudo-random gaussian noise codes, and assuming that the environmental noise is also gaussian, (3.9) maximizes the SNR for the demodulated signals. When applying matched filtering to the demodulated signal the following output of the matched filter can be found:

$$\begin{aligned}
h(n) &= s_{\text{ref}}(-n) && \text{discrete matched filter, assuming } C = 1 \text{ and } t_0 = 0 \\
s_{\text{out}(n)} &= s_{\text{in}}(n) \circledast h(n) = s_{\text{in}}(n) \circledast s_{\text{ref}}(-n) \\
s_{\text{out}(n)} &= \sum_{k=-\infty}^{\infty} s_{\text{in}}(k) s_{\text{ref}}(n+k) && (3.10)
\end{aligned}$$

where $\circledast$ is the convolution between two vectors. When the reference signal s_{ref} is the same as the demodulated polar signal, then the solution to (3.10) is equal to the autocorrelation function (3.2), which is a delta function ($\delta(t - t_0)$) shifted in time to the time at which the reference signal and the demodulated input signal completely coincide. When the reference signal is not present in the demodulated signal, then the solution to equation (3.10) is the crosscorrelation of two random signals, which is zero, as shown in (3.3). These observations mean that matched filtering the demodulated signals can be done in two ways. The first is convoluting the signal with a timeflipped reference signal. The second is numerically computing the cross correlation of the demodulated signal and the reference signal. Both methods will result in the same output with a maximized SNR.

It should be noted that equations (3.2) and (3.3) are analytically derived equations. This means that any computational approach to compute these equations will only approach the analytical solution in the limit for increasing sample sizes. This means that no matter how signals are chosen and what properties they have, there will always exist convolutive noise, due to the fact that the correlation functions will be numerically computed.

The generalized cross correlation phase transform

To decrease convolutive noise, source [25] provides for a generalized crosscorrelation using a phase transform, to minimize the convolutive noise. In appendix A.3 a derivation for the generalized cross correlation phase transform (gcc-phat) is given. It is an ideal function when the to be correlated signals are exactly shifted versions of each other. In reality the demodulated signals are not exactly shifted versions of the reference signals. This is due to the fact that the reference signals are the ideal polar line codes, while the demodulated signals have limited bandwidth and other artifacts due to the channel and modulation/demodulation. When the to be correlated signals are not exactly shifted versions of each other, then the gcc-phat might fail in producing a single delta function. This is because, when the to be correlated signals have limited bandwidth, large errors can be introduced by weighting each frequency with its amplitude. This is due to the fact that the phase in the frequency domain outside of the signals bandwidth can be random. When using gcc-phat, the random phase are weighted as important as the phases within the bandwidth of the signal. Within section 4.3.1 it will be checked whether the gcc-phat method works properly with the bandlimited signals of the designed system.

Implemented matched filter

As explained in section 3.2.4, the transmitted signal for each unit consists of a stream of 10 codes repeated 3 times. The first step in implementing a matched filter is creating the reference signal or signals. It was chosen to implement three different reference signals containing a single code, a batch of 5 codes and all 10 codes. The reasoning behind choosing multiple reference signals is given in section 3.5.2.

Theoretically the demodulated signals correspond to the polar line codes. However, due to filtering, not the whole bandwidth of the linecodes is preserved. This means that a distorted block wave is the output of the demodulator. The distorted block wave can be seen as noise superpositioned on a true block wave. This means that a matched filter with a polar block wave as reference signal will maximize the SNR of the distorted block wave.
To create the reference signal of a single code each bit of the linecode has to be repeated for $F_s/F_{\text{bit}} = 48\text{kHz}/3\text{kHz} = 16$ samples. Within this simple formula, the sample frequency is divided by the bit frequency to obtain the amount of samples in which a single bit is transmitted.
To create the reference signal consisting of a batch of codes or all codes, the reference signals of each single code has to be appended to each other with the correct amount of spacing samples between them.

With the reference signals created, the matched filter method can be implemented. As explained in section 3.4.1 the output of the matched filter can be found by correlating the demodulated signal with the reference signal, or by computing the convolution between the demodulated signal and the time inversed reference signal. Because in MATLAB there exists an optimized convolution function, optimized for minimum computation time. Since many matched filter operations have to be done for each unit in the system it is important that each matched filter operation takes as little computation time as possible. It is for this reason that the matched filter is implemented using the convolution technique.

3.4.2. Peak Detection Method

After the demodulated signals have been filtered with a matched filter, its output is ideally a signal with easy to detect delta peaks. However, due to non perfect matching due to convolutive noise and channel noise, the output of the matched filter does not only contain peaks, but also noise signals with a certain amplitude. It is the goal of a peak detection method to detect the correct peaks resembling the delta functions of an ideal matched filter. It should be taken into account, while designing the peak detection method, that the Bosch DICENTIS system can be placed in highly reverberant environments. This means that reflections of the to be detected signal can also be present in the received signal. It is even possible that, due to the directivity of the speakers and microphones of the different units, the peak resulting from a reflection is larger than the peak resulting from a direct line of sight path. In the next section a method will be explained, based on threshold determination, that theoretically will find the correct peak.

Threshold estimation

As stated in the previous section, it can not be assumed that the maximum values of the matched filters output equal the correct peaks to be detected. It is for this reason that it is common to choose a threshold value above which different peaks will be compared. The direct path between units is always the shortest path. This means that the peak, due to the direct path, is always the first peak. After the threshold is estimated, the first peak above the threshold is estimated to be the detected direct path. It is for this reason that the threshold value is carefully chosen. If all properties of the different forms of noise present in the signal are known, a theoretically optimal threshold for detecting peaks can be derived. When the noise is purely zero mean and gaussian the theoretically optimal threshold is known, as shown in [26].
However, due to the fact that the noise is a superposition of convolutive noise, channel noise and interference with other codes, it can not be assumed that the present noise after matched filtering is zero mean and gaussian. It is for this reason that a threshold will be chosen, based on the computed output of the matched filter.

There exist many methods to determine the threshold for the detection of signals. [27]. Many threshold estimation methods are derived for energy detection methods, because these detection methods are very sensitive to the properties of the present noise.
The method used to estimate the threshold is the quiet time approach, which is specifically designed for the matched filter. [27] With this approach the output of the matched filter, where it is known that no signal is present, is correlated with the reference signal. Equation (3.11) is solved numerically to obtain an estimation for the threshold. Within this equation $w(n)$ is the noise before the matched filter, where it is known that no signal is present. It should be noted that the estimated threshold is based on the maximum of the numerical output of the cross correlation with noise. This maximum value is multiplied with a factor λ_{f}, to set the estimated threshold a margin above the computed maximum value. This method can be seen as setting a threshold with a margin above the noise level and it assumes intrinsically that the SNR is positive on a dB scale. The value of the multiplication factor will be chosen for optimization of either maximizing the detection chance or minimizing the false alarm chance.

$$\lambda = \lambda_{\mathrm{f}} \max \left(\sum_{N} w(n) s_{\mathrm{ref}}(n+k) \right) \tag{3.11}$$

Implemented threshold estimation

To implement the threshold estimation, two approaches can be made.

1. Static threshold estimation, while not sending any information through the system.
2. Dynamic threshold estimation. Based on measurements and known characteristics of the signal the threshold can be estimated while doing the TDOA measurements.

The first approach is suboptimal for multiple reasons. The first, as stated in [27], static threshold estimations are not reliable, because of the uncertainty of noise present in the environment over time. By estimating the threshold, while not sending signals from other units, the noise due to other units is not taken into account even though this noise amplitude might be larger than the noise of the environment.

Dynamic estimation of the threshold is the better approach of the two, because it takes the differences of environmental noise in time and the noise due to other speaker units over time into account. This approach can be implemented, because the nature of the sent signal is known. As stated in section 3.2.4 the sent signal contains multiple codes, but more importantly a repetition.

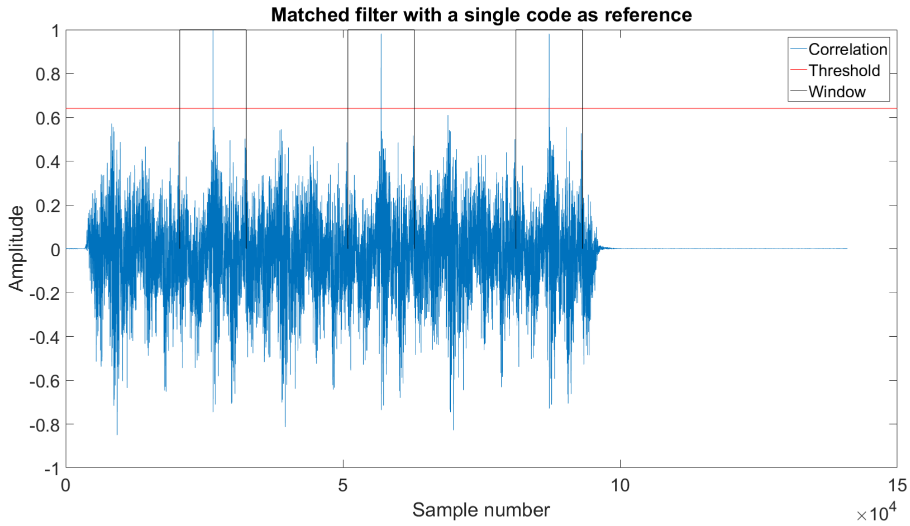


Figure 3.10: Implemented Threshold Estimation

The first step in the implementation of the dynamic threshold estimation is the determination of where there is no signal present in the output of the matched filter. Knowing that the sent signal repeats itself multiple times, intervals can be found in the output of the matched filter, where there is only noise present. Assuming that the highest peaks in the output signal are because of the correct signal sent (Assuming $SNR > 1$), these so called noise intervals can be determined. When such a high peak, whether it is a reflection or not, is found, a time window around this peak can be defined. The time window is chosen such that it is made sure that the direct path is included in this window. To determine the size of the window, the amount of time that can differ in arrival times of line of sight (LOS) signals and reflections have to be compared. In (3.12), a way to compute the window size (in sample count) is given. It computes the window size based on the maximum difference between the line of sight path and the reflection. It seems reasonable to assume that the maximum difference between different reflections and the line of sight signal within a $30 \times 30m^2$ room is around 40 m. Assuming that the speed of sound is $343\frac{\mathrm{m}}{\mathrm{s}}$ and a sample frequency of 48kHz, equation (3.12) results in a window size of about 6000 samples.

$$S_{\mathrm{w}} = \frac{[\max(s_{\mathrm{r}}) - s_{\mathrm{LOS}}]\, F_s}{c_{\mathrm{s}}} \tag{3.12}$$

Here, S_{w} is the window size in samples, s_{r} is the reflection distance, s_{LOS} is the line of sight distance and c_{s} is the speed of sound. When all windows with the to be detected signals are deleted from the output of the matched filter, then only the intervals with noise are left over. Simply taking the maximum of left over signal and multiplying it with a certain factor suffices equation (3.11). In Figure 3.10 the implemented threshold detection can be observed. In this case, the multiplication factor was chosen to be 1.05, implying that the threshold is estimated 5% higher than the noise amplitude.

As stated before, correlations with different reference signals will be computed. The different reference signals differ in the amount of codes they contain and effectively the amount of energy the reference signal has. The difference of these codes is used by the determination of the multiplication factor of the estimated threshold. The reference signals containing more codes have a larger signal energy resulting in a larger SNR after the matched filter. As stated in [27] a higher SNR increases the detection chance of a signal. This means that the multiplication factor for larger reference signals can be chosen to be higher than for smaller reference signals to minimize the false alarm chance. The smaller reference signals, having a smaller SNR, are still very useful, because they tend to have a higher time resolution, as is explained in section 3.2.3.
The chosen multiplication factors are:

- $\lambda_{\mathrm{f}} = 1.05$ for single codes, maximizing the detection chance.
- $\lambda_{\mathrm{f}} = 1.40$ for small batches of codes, making a tradeoff between maximizing the detection chance and minimizing the false alarm chance.
- $\lambda_{\mathrm{f}} = 2.0$ for all codes combined, minimizing the false alarm chance.

3.5. Estimation of Inter-Unit Propagation Times

Once a set of peaks are detected, this set of data needs to be converted into an appropriate set of arrival times. After this, an error analysis can be done on the found times from the generated data sets. This section will cover the generation of estimated time differences with respect to a reference node and the use of multiple estimators to indicate the change of the done estimations being correct.

3.5.1. Determining Propagation Times

Firstly, from the detected peaks, a set of appropriate times are estimated. For the estimation of the time differences between receiving a signal at a given microphone with respect to a reference microphone, the assumption is made that all microphones are synchronized. The audio card used to test the system uses the same clock for all connected microphones, thus in the test environment utilized, this is indeed the case. For application for the Bosch DICENTIS system, it is given that all units in the system are synchronized within 100 μs. Given a sampling frequency of 48 kHz, this means that the inaccuracy of the synchronization is $100 * 10^{-6}/48000$ = 4.81 audio samples. This, assuming a speed of sound of 343 m/s, leads to an inaccuracy of 343*4.81/48000 $\approx$ 3.5 cm. Given that another subgroup in the overarching project is tasked with decreasing this error, for the implementation on the Bosch DICENTIS system, for our purposes, this inaccuracy is assumed to be 0 cm.

Given that all microphones are effectively synchronized and the peak detection system produces a set of vector entries from the filtered audio data where a peak is estimated, the propagation delay with respect to a reference microphone is determined by subtracting the found vector entries from a given microphone recording from the vector entries of the reference microphone, gaining a vector of propagation delays. This principle is shown in the following equation:

$$\mathbf{d}_{\mathrm{pm}} - \mathbf{d}_{\mathrm{pr}} = \begin{bmatrix} s1+\Delta \\ s2+\Delta \\ . \\ . \\ . \\ sn+\Delta \end{bmatrix} - \begin{bmatrix} s1 \\ s2 \\ . \\ . \\ . \\ sn \end{bmatrix} = \begin{bmatrix} \Delta \\ \Delta \\ . \\ . \\ . \\ \Delta \end{bmatrix} \tag{3.13}$$

Where $\mathbf{d}_{\mathrm{pm}}$ is the vector containing the sample entries of the detected peaks of a random microphone in the system, $\mathbf{d}_{\mathrm{pr}}$ is the vector containing the sample entries of the detected peaks of the reference microphone and Δ is the propagation time of a signal to the random microphone in question with respect to the measured propagation times of the reference microphone. Given that the sample frequency is known to be 48 kHz, the propagation delays can simply be determined from the known sample delays (Δ) by dividing the sampling delays by 48000. With this, a set of propagation delays are found for each used microphone for each peak detection done.

3.5.2. Error Analysis

In the implemented peak detection system, three sets of peaks are generated:

- Peaks from each individual found code in a signal.
- Peaks from a set of found codes in a signal.
- Peaks from all found codes in a signal.

This means, after the propagation delay determining step is done, a set of propagation delays are generated for each microphone used for each of the peak detection systems implemented. Each peak detection system is optimized for a different purpose. The peak detector for individual transmitted codes is optimized to have a high detection rate, but as a result also has a high false alarm rate. The peak detector for the full transmitted set of transmitted codes is optimized to have a low false alarm rate, but as a result also has a low detection rate. The peak detector for a set of transmitted codes is optimized in terms of detection rate and false alarm rate to be in the middle of the other two. The reason for this optimization lies with the SNR of the detected signals for each peak set. Using only one code when doing a peak detection, the SNR is the lowest, however the resolution of the found peaks is the highest. With this it is meant that blurring due to interfering reflections is minimized, given that the amount of reflections overlapping decreases as the transmission time of the to be correlated code decreases. Because of this, single code autocorrelations can best be used to get a wider range of potential propagation times with a high resolution, then to search for a specific correct one with a larger chance of not detecting the right signal. The opposite argument can be given for the autocorrelation estimator for full sets of codes, because here the SNR is the highest (given more energy in the signal over which a correlation is done). However, in this case the resolution of the estimations is the lowest, given that the chance of the signal consisting of multiple codes overlapping with a reflection has increased, thus the chance of blurring occurring also has increased. Because of this, in the other extreme, this peak detection system is optimized to make use of the higher SNR to gain a better scope of where a correct peak occurred.

Given that the codes to be detected are an estimation of white noise and are therefore (to an extent) uncorrelated, a set of approximated white noise codes also form an approximated uncorrelated code with respect to the other code sets. Because of this, the peak detectors also are estimated to be uncorrelated. This means that with three detected sets of peaks, three uncorrelated estimations are done for the propagation delays for each used microphone. However, the three estimators are not independent, given that if one estimator finds a peak, the chance of the other two also finding that peak increases. This is the second core principle of why multiple peak estimators in the form of detectors are used.

To show why using multiple peak detectors in the way implemented also increases the chance of finding a correct peak, the peak detector correlating single codes is called p1, the peak detector correlating sets of codes is called p2 and the peak detector correlating all sent codes is called p3. The following table indicates a set of events that can occur related to peak detector p1, p2 and p3:

Table 3.4: Event Names

Detector Name	Event Where Detector Detects Peak(s)	Event Where Detector Detects Correct Peak
p1	A’	A
p2	B’	B
p3	C’	C

The estimation given by p1, p2 and p3 is weighted based on the amount of codes that are correlated. This means that an estimation given by p1 has a weight of 1 whereas an estimation given by p3 has a weight of 10, given that a correlation is done over 10 codes. Given the way p1, p2 and p3 are optimized, P(A’) > P(B’) > P(C’) and P(A) < P(B) < P(C). Because p1, p2 and p3 are not independent, P(B'_A) > P(B) and P(C'_B) > P(C). Because of this, the chance of finding a correct peak has increased because P(A’) > P(B’) > P(C’). Furthermore, Given that P($A_{\bar{B}'}$ > 0 and P($B_{\bar{C}'}$ > 0, the overall chance that a correct value is found among the three estimators is higher than that of the individual estimators. Given P($\bar{A}$) > P($\bar{B}$) > P($\bar{C}$), the weighing factor is given for results from p1, p2 and p3, such that the chance of a wrong detected peak outweighing a correct detected peak is minimized, but the chance of a correct peak being found is increased.

After p1, p2 and p3 detect a set of peaks correlating to a set of propagation times, the detected peaks are compared. Given that a set of 10 different codes is repeated 3 times in order to decrease the chance that any single measurement error dominates the list of the detected peaks from p1, p2 and p3, the three sets of estimated propagation times from peak detector p1, p2 and p3 will always contain a list of multiple estimations. The comparison occurs by first appending the found propagation times from p1 to the list of found propagation times for each microphone, after which the found propagation times from p2 are compared to p1, such that the resolution of the propagation times of p1 remain used, and if significantly differing propagation times are found from p2, these times are appended to the list. The same process occurs for the propagation times from p3, but now with respect to p1 and p2. The occurrence of a certain propagation time, as mentioned before is weighted based on occurrence and the weighing factor of the peak detector. Once all propagation times are listed, their weight relative to the total weight of all weighted propagation times is calculated. This is then presented as the estimated chance of each found propagation time being correct.

4
Results

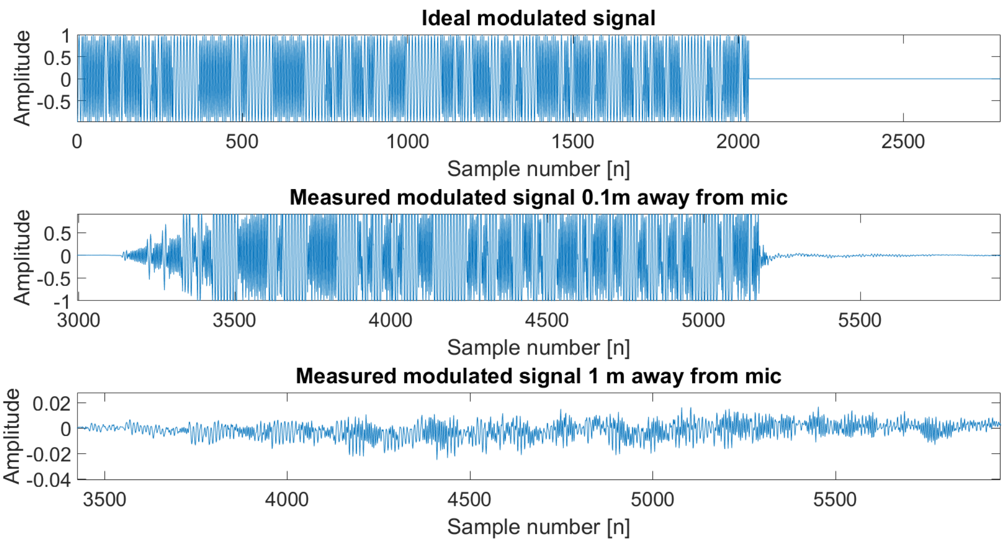
4.1. Measurement Setup

In every measurement, the setup consisted of the use of a Bosch unit speaker, where the router to which the unit was connected had it's analog input connected to a Focusrite 18i20 audio interface. This audio interface is connected via a USB connection to a laptop running the written localization program in MATLAB. Furthermore, a set of AKG C417 microphones are used, which are a replacement for the Bosch DICENTIS microphones due to practical reasons around the access of the software stack of the DICENTIS system. In total, 8 microphones are connected to the audio interface. All devices connected to the audio interface are synchronized, such that when a command is given in MATLAB to transmit and/or record, this occurs synchronous for all connected devices. Given that once audio is presented to the DICENTIS router, the data is not relayed with a fixed offset towards the speaker. Because of this, TDOA measurements are done between the connected microphones. One microphone in any given setup is then chosen to be the reference microphone, in which case all other microphone measurements are with respect to that reference microphone.

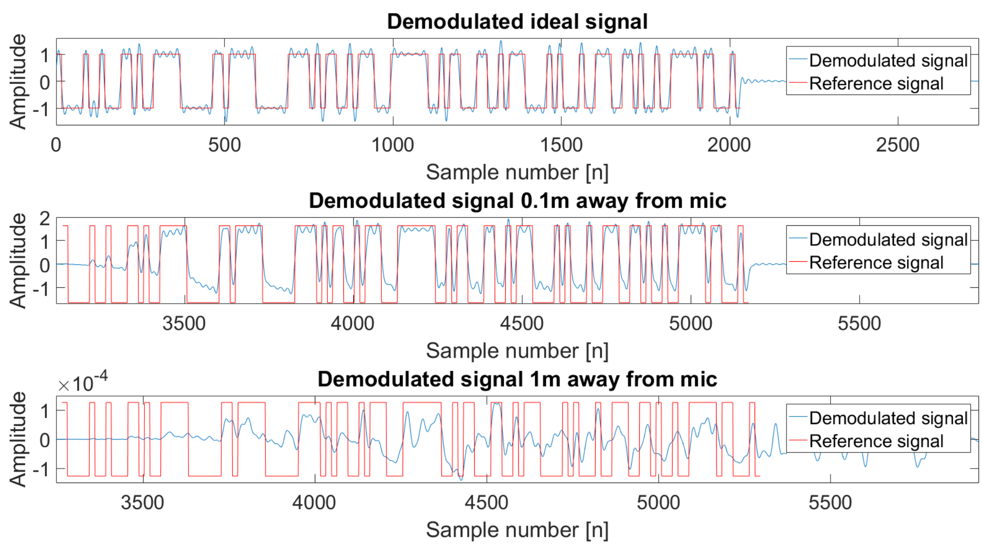
4.2. Filtering and Demodulation

In order to test the demodulator, a known audio signals are presented at the input. Figure 4.1a shows the recorded audio data that is presented at the input of the demodulator. When analyzing the output of the demodulator in the time domain, which is shown in Figure 4.1b, the original square wave signal is clearly visible in the output signal for close range audio measurements and reasonably well in longer range measurements. However, a rippling effect is also noticeable, which indicates that the high frequency components of the signal have been filtered out. This coincides with the expectation, given that the demodulator has filtered a band of 4.5 kHz, while the band of the transmitted signal was 6 kHz. This is shown more clearly in Figure 4.1c, where the recovered baseband signal is compared to the generated reference signal. In this figure it is clearly visible that in all cases, the signal is preserved within a 4.5 kHz bandwidth whereas the rest of the signal is filtered out. Given these results, it is clear that the demodulator functions as predicted.

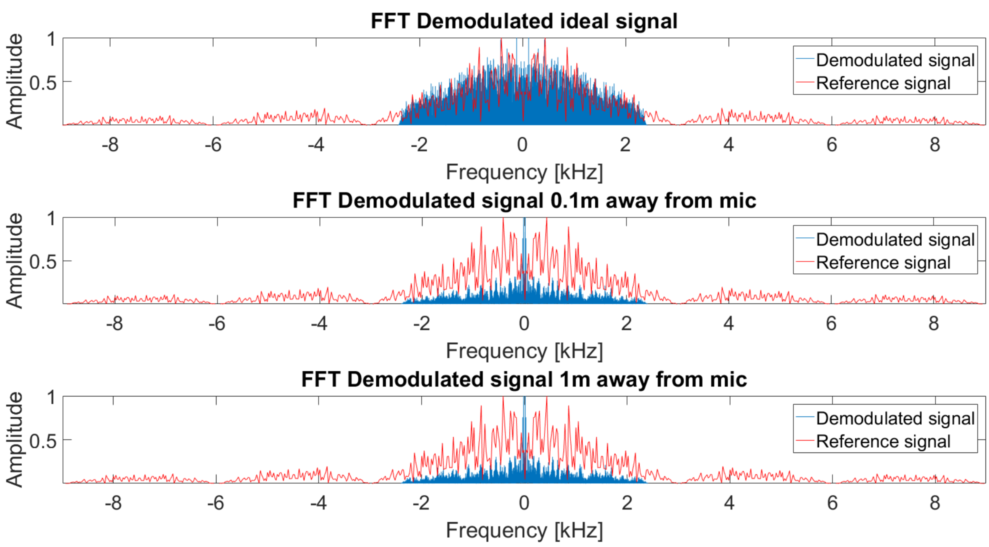
An interesting observation done, is the fact that in Figure 4.1c, the energy in the baseband signal is more concentrated towards DC than in the original signal. This can be explained by the fact that the speaker transmits less power for one of the carrier frequencies. Another explanation is that the microphones can be more sensitive to one of the carrier frequencies. This means that the demodulated signal will no longer have a zero-mean, thus an extra DC component is introduced into the signal. This zero-mean offset can also be observed at the output in the time domain, where a small positive offset is observable. This means that the carrier of the -1's in the signal was dampened more than the carrier of the 1's. This dampened carrier is the 10 kHz carrier, given that the OOK signal containing the -1's was modulated onto a 10 kHz carrier. This means that frequencies around 10 kHz are dampened more than frequencies around 5 kHz in the current setup.



(a) Time domain input of the demodulator



(b) Time domain output of the demodulator



(c) Frequency domain of the output of the demodulator

4.3. Signal Detection

4.3.1. Matched Filter vs Gcc-Phat

As stated in section 3.4.1, gcc-phat can be used in wide band signals to minimize the convolutive noise of a matched filter. To test whether this method can be used in the designed system, where limited bandwidth is available, two tests are done. Within the first test, ideal polar signals are used to compare the two methods. Within the second test, demodulated audio signals are used. As can be seen in Figure 4.2, the gcc-phat method is more accurate than the matched filter when ideal signals are used. This is because ideal signals contain a very large bandwidth and phase errors are not amplified. However, with the case of the demodulated audio signals, the gcc-phat method performs worse than the matched filter. This is because of the band-limited signals, due to the filtering process. For this reason, the implemented detection method is only a matched filter and no use is made of the gcc-phat method.

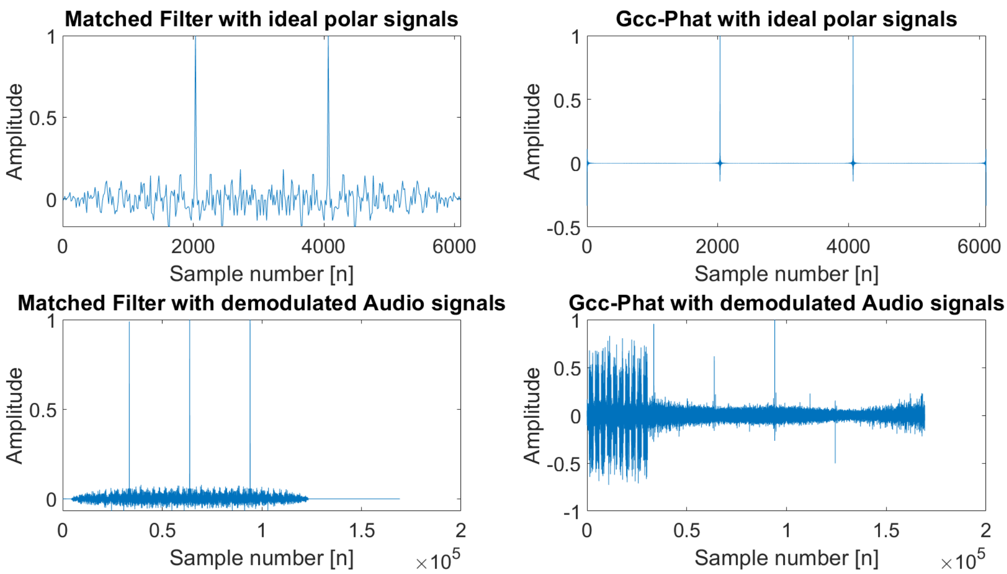
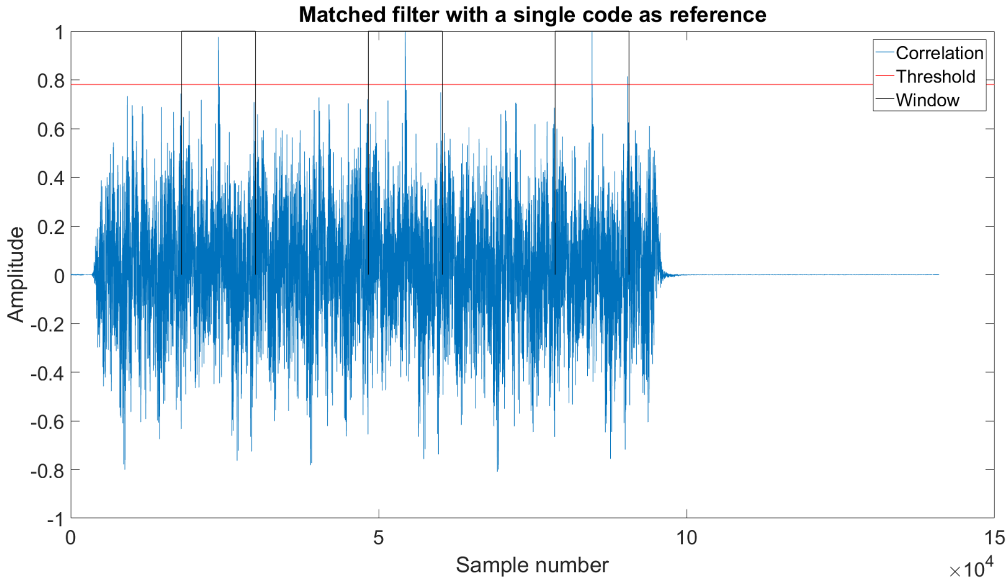


Figure 4.2: Matched Filter vs Gcc-Phat

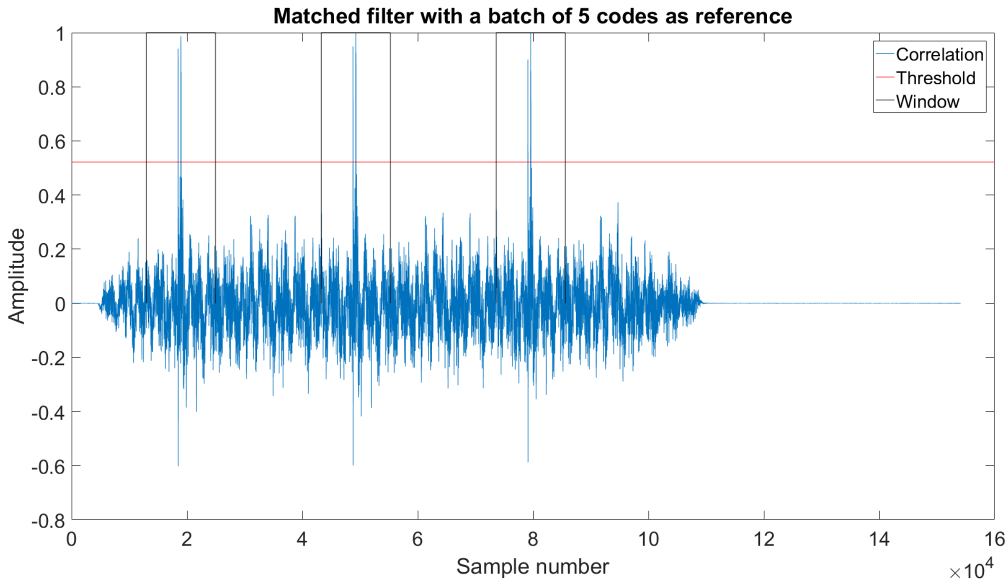
4.3.2. Implemented Matched Filter

In Figures 4.3a, 4.3b and 4.3c a correlation using the previously described three different peak detectors is done. In Figure 4.3a, a correlation is done using one pseudo-random code, in Figure 4.3b, a correlation is done using using 5 codes and in Figure 4.3c, a correlation is done using 10 codes. The first thing noticeable is that the relative noise with respect the auto-correlation peak decreases as the amount of correlation codes increases. This makes sense, given that the SNR increases due to the fact that more energy is in the correlated signal if more pseudo-noise codes are used as reference signal.

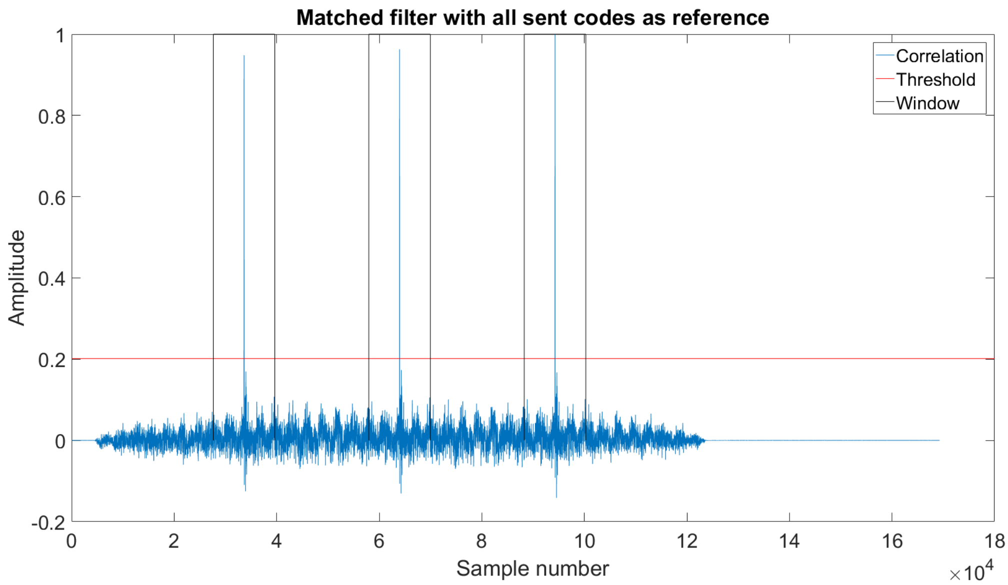
What is also noticeable, is that the observed noise mainly occurs in intervals, this is most likely due to the fact that the observed noise in part is the cross-correlation between other transmitted pseudo-noise codes. This noise is due to the fact that the cross correlation is numerically estimated.



(a) Matched filter output, reference signal is a single code



(b) Matched filter output, reference signal is a batch of 5 codes



(c) Matched filter output, reference signal is all codes combined

Figure 4.3: Matched filter outputs for different reference signals

4.3.3. Peak Detection Method

The peak detection method used, is based on dynamically estimating it's value above which peaks will be detected. In most of the cases the threshold estimation works properly, as can be seen in Figure 4.4. However, as is shown in Figure 4.5, the estimation of the threshold might fail. In this case the threshold is estimated to be larger than any sample in the signal and the three expected peaks can not be detected. Because of this, an alternative peak detection threshold is utilized for this scenario, where within the window of a local maximum, as shown in Figure 4.5, all peaks are considered which are above 75% of that local maximum. In this case, 75% is purely heuristically chosen.

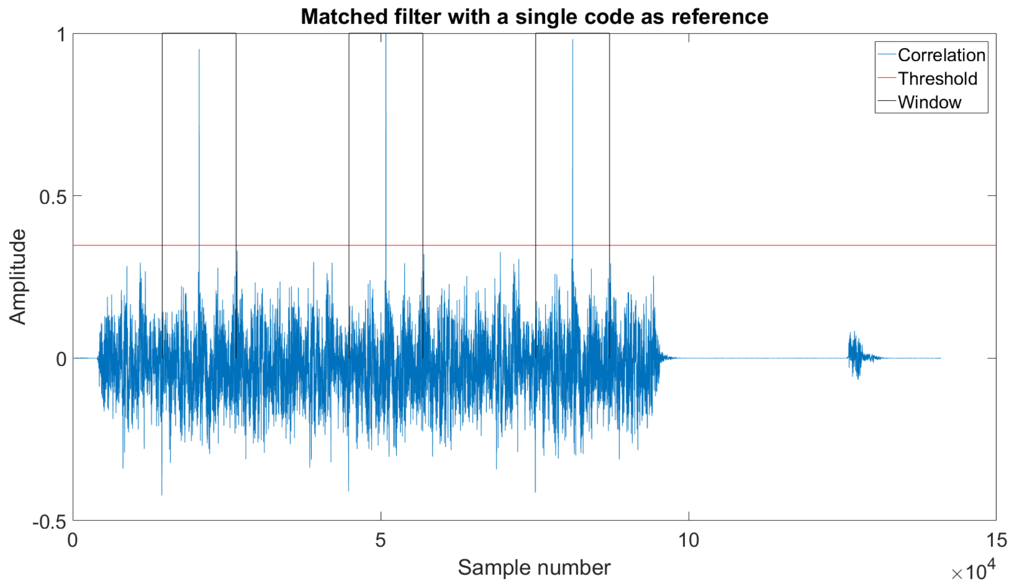


Figure 4.4: Correct threshold estimation

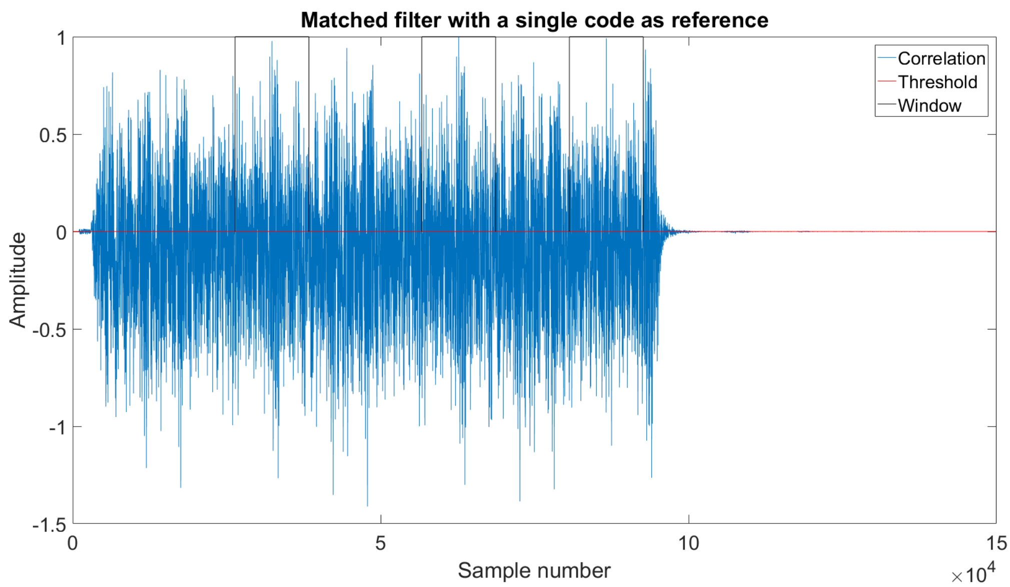


Figure 4.5: Failed threshold estimation

4.4. Determining Propagation Times

With the propagation times computed by the designed system, it is critical to investigate whether the computed propagation times are reliable. To verify the computed results, next to propagation times, estimated distances are computed. Knowing that time and distance are linearly proportional to each other by means of the speed of sound, measured distances give an accurate representation of the results. Mainly 2 different setups are used to determine the accuracy of the designed system:

1. A 1D problem. A line is drawn on the ground, to which the microphones of the system are attached. The TDOA timing measurement is directly proportional to the distance between the microphones. This setup is used to determine the direct line of sight performance for long distances and to investigate whether the system is reliable when multiple sources are sending simultaneously).

2. A 3D problem. A cube-like measurement setup is created, where the microphones are put on the corners of the cube with sides of roughly 1.5 m in length, as is shown in Figure 4.6. This setup is placed in a typical office environment and is used to determine if the directivity of the speaker has great impact on the overall performance of the system. As the exact microphone locations and the exact unit locations can not be accurately measured in the given setup, these measurements should show the same order of magnitude standard deviations, although they can exceed the required 3 cm.

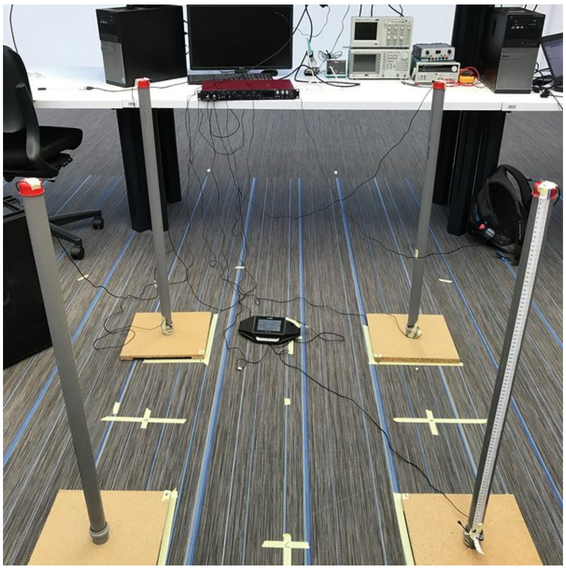


Figure 4.6: Measurement setup in the form of a cube

4.4.1. Error Analysis

1D Setup for multiple Distances

For the 1D setup, multiple measurements were done at different distances from the microphones. Within a straight line the Bosch DICENTIS unit was put either 5m or 25m away from the receiving microphones. In both cases, distance estimations were done and compared to the correct measured distance. Also the estimator for the reliability of the measurement is compared to the fraction of correct amount of measurements divided by the total amount of measurements. Table 4.2 shows that the estimated reliability of the measurement is almost equal to the fraction of total correct measurements. This gives a good indication that the found timing/distance values are accurate.

Table 4.1: Average estimation errors

Measurement Distance	Mean Position Error [m]	Variance Error [m]
5m	-0.0077m	$5.2 * 10^{-5}$
25m	-0.0280m	$1.91 * 10^{-4}$

The fact that the mean of the 25m distance measurement is not zero, can probably be attributed to the use of a constant speed of sound, which can differ from the real physical speed of sound. The variance of the error is at most $2*10^{-4}$, which results in a standard deviation of about 1.5cm. As stated in 2.2, the standard deviation should be less than 3 cm, as is the case. Overall it can be stated that the system, when under line of sight conditions, works within specified requirements up to at least 25 meters.

Table 4.2: 1D Distance measurements and estimator comparison

Measurement distance	5m Distance				25m Distance			
	Estimated chance of correct measurement	Fraction correct measured distances	Estimated Distance [m]	Correct Distance [m]	Estimated chance of correct measurement	Fraction correct measured distances	Estimated Distance [m]	Correct Distance [m]
Reference Mic 1	1	1	0	0	1	1	0	0
Mic 2	0.9181	0.9181	0.3379	0.34	0.7861	0.8003	0.3195	0.34
Mic 3	0.9250	0.9250	0.6458	0.65	0.7764	0.7907	0.6236	0.65
Mic 4	0.9222	0.9222	0.9106	0.91	0.7528	0.7572	0.8858	0.91
Mic 5	0.9083	0.9083	1.1663	1.18	0.7986	0.8086	1.1371	1.18
Mic 6	0.8597	0.8597	1.3901	1.4	0.7611	0.7463	1.3662	1.4
Mic 7	0.9389	0.9389	1.6818	1.7	0.8042	0.8129	1.6642	1.7
Mic 8	0.9472	0.9472	1.9859	2.0	0.7875	0.7900	1.9593	2.0

1D Setup for multiple sources

The propagation time estimation system is designed to have multiple sources sending their codes simultaneously. To test if multiple codes are indeed distinguishable, multiple units are used to send different codes simultaneously. In Table 4.3 the results of the use of multiple sources can be observed. Within this measurement two units were set on different distances from the microphones (3m,5m) and only one of the units is being detected by the system. Table 4.3 shows that the distance estimation is still accurate to about a centimeter and that the variance is less than $2 * 10^{-4}$. This means that the standard deviation of the errors is about 1.5cm and is less than the maximum 3cm standard deviation, required by section 2.2. It is therefore concluded that the system is able to distinguish the different simultaneously sent codes from multiple sources, which improves the total measurement time, as measurements can be done in parallel.

Table 4.3: Distance measurements with multiple sources sending simultaneously

Microphone	Average Distance [m]	Correct Distance [m]	Mean Position Error [m]	Variance Error [m]
Reference Mic 1	0	0	0	0
Mic 2	0.3293	0.34	-0.0107	$2.7858 * 10^{-5}$
Mic 3	0.6468	0.65	-0.0032	$5.322 * 10^{-5}$
Mic 4	0.9147	0.91	0.0047	$9.91 * 10^{-5}$
Mic 5	1.1681	1.17	-0.0019	$1.126 * 10^{-4}$
Mic 6	1.3951	1.4	-0.0049	$1.49 * 10^{-4}$
Mic 7	1.6975	1.7	-0.0024	$1.55 * 10^{-4}$
Mic 8	1.9812	2.0	-0.0188	$1.26 * 10^{-4}$

3D Setup

When it comes to the 3D setup, as is expected, slightly less performance has been achieved due to inaccurate position measurements. Because of the directivity of the speaker and the way the Bosch DICENTIS units are designed, there is not always a direct line of sight from speaker to microphone. In Figure 4.7, results are shown in which all errors of each measurement are listed. It is indicated that multiple measurements result in errors between 0.05 m and 0.1 m. Still, it is found that the mean error is -0.004m, but the variance is a bit larger, 0.01m, which results in a standard deviation of 10 cm. This standard deviation is larger than the required standard deviation of 3 cm. However, as stated before, this can partly be attributed to the inability of measuring the reference nodes with an accuracy of less than 4 cm with the equipment available. This means that the computed standard deviation of 10 cm is not accurate for the exact standard deviation of the exact position. Since most errors are still contained within 0.1 m of the inaccurately measured nodes and the errors are of the same magnitude as is the case with the 1D setup, this shows that the system is likely to perform according to the accuracy requirements 2.2 in a properly measured 3D setup.

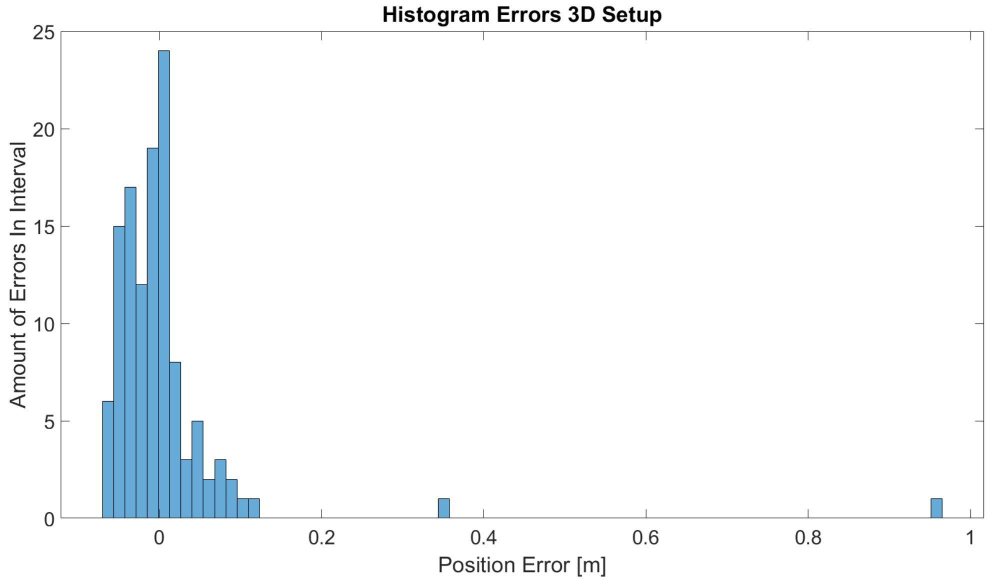


Figure 4.7: Histogram of errors in 3D measurement setup

4.5. Estimation Improvement by means of Multiple Estimators

Using the same generated data as was used in section 4.4.1, an analysis is done on the overall improvement of the propagation time estimation system due to the use of multiple uncorrelated propagation time estimators (with abbreviations mentioned in section 3.5.2) instead of just one. Here, the accuracy of the individual estimators is compared to the accuracy of the total estimator. From this, the error correction improvement can be calculated. From the 240 conducted measurements combining all microphone measurements and all distance measurements, the results are shown in the Table 4.4:

Table 4.4: Correct estimations from specific estimator sets

Results from sets of peak detection estimators	Amount of correct results from estimator set
Total Estimator	205
Only p1 and not p2 and p3	1
Only p2 and not p1 and p3	0
Only p3 and not p1 and p2	0
Only p1 and p2 and not p3	0
Only p1 and p3 and not p2	0
Only p2 and p3 and not p1	4
Both p1, p2 and p3	201

As is mentioned in section 4.4, p1 is the peak detection estimator where only single codes are correlated, p2 is the peak detection estimator where sets of 5 codes are correlated and p3 is the peak detection estimator where sets of 10 codes are correlated.

The final estimator estimates 205 propagation times correct within a 5 cm margin of error, although in only 201 cases, all three of the individual estimators estimate the correct propagation time. This means that 4 correct estimations in the total estimator were due to the fact that multiple individual estimators were weighted such that the correct result can still be found, even if a single estimator fails. The total estimation improvement E_i can be calculated using equation (4.1):

$$E_i = \frac{E_{mc}}{E_t} \tag{4.1}$$

where E_{mc} is the amount of wrong estimations (considering the three individual estimators) corrected in the total estimator and E_t is the amount of wrong estimations considering the case of all three individual estimators having to be correct. This results in an improvement of 4/39 = 0.1025, which is considered significant. These results prove that the use of multiple uncorrelated estimators improves the overall performance of the system, as is expected.

4.6. SNR of Total System

With the whole system implemented and tested, it is good to know the SNR gain of the complete system. To determine the system signal to noise ratio, a measurement is done, while white noise is imposed on the environment of the system, while a single code is repeatedly sent for three times. The signal to noise ratio will be numerically computed within three stadia of the system:

1. The input phase. In this phase the input SNR will be determined.
2. After the demodulator. The SNR gain of the demodulator will be determined.
3. The output of the matched filter. This is the output of the whole system. The gain in SNR with respect to the input SNR is the SNR gain of the complete system.

The different wave-forms in the different stadia of the system can be observed in Figure 4.8. It is easily observed that a significant SNR gain is achieved with the signal. To quantify this gain, the average noise power and average signal power are computed in each state of the signal.

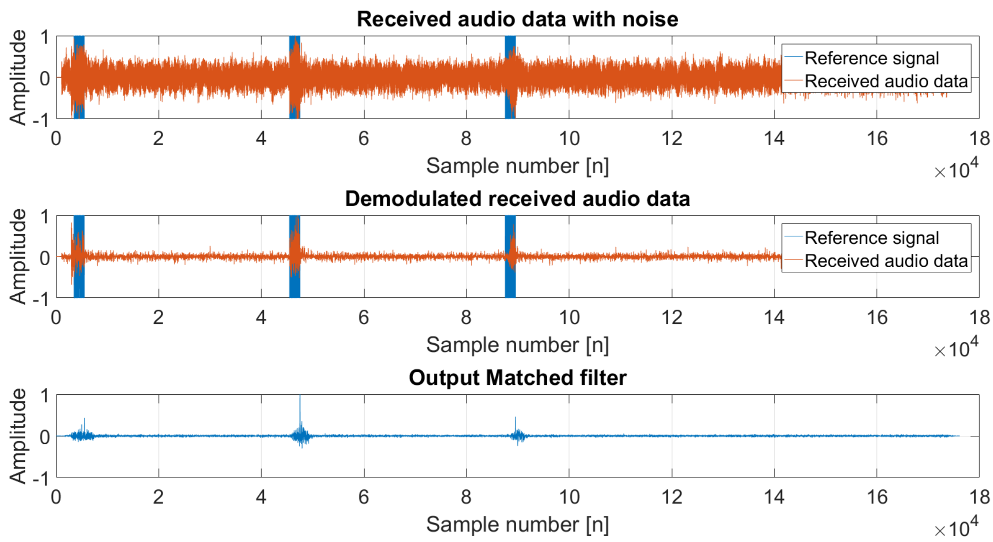


Figure 4.8: SNR determination, based on imposed white noise in the environment

In (4.2) the average signal power P is computed of signal $x(n)$. Knowing the reference signal, the intervals, in which noise is the only source, are easily recognized, as can be observed in Figure 4.8

$$P = \frac{1}{N_1 - N_0 + 1} \sum_{n=N_0}^{N_1} (|x(n)|)^2 \tag{4.2}$$

In Table 4.5 the computed signal to noise ratio's are shown. It is found that the designed system has a relatively large noise rejection with regard to the signal power. This large increase in SNR is to be expected from a matched filter, which maximizes the SNR given that the noise is Gaussian, as is the case with white noise.

Table 4.5: Computed SNR for different stadia of the system

Stadium of system	Computed SNR	Relative gained SNR with regard to the input
Input Signal	2.4 dB	—-
Output Demodulator	6.1 dB	3.7 dB
Output Matched filter	34 dB	31.6 dB

5 Discussion and Conclusion

5.1. Discussion

In the results, it was found that the estimator for the accuracy of measurements proves reliable 2.2. In most circumstances, a correct chance of a measurement being reliable with a standard deviation of 3 cm is given. This means that the produced weighing factor for the estimated audio propagation times can be taken into account when processing the presented data further. Furthermore, it is observed that using multiple codes with multiple peak detection estimators increases the amount of correct results by an acceptable margin. This means that in most cases, the system produces sufficiently accurate measurement results within the bounds of the program of requirements 2.2. However, as is mentioned in the results, under certain conditions where the speaker of a unit is not facing measuring microphones, the system proves less reliable due to a decrease in received signal power. Because of this, for the next iteration of this system, received signal power should be taken into account when determining the accuracy of a set of measurements. However, under typical conditions where a speaker is facing a recording microphone and there is sufficient line-of-sight, the requirements concerning the required accuracy and speaker-microphone distance are fulfilled 2.2.

Considering the required scalability of the system in the program of requirements. Given the limitations in implementation freedom that was available for the Bosch DICENTIS system, no significant tests were conducted to fully test this scalability. However, it was tested how well the system can function with multiple transmitted signals at multiple distances, and as is shown in the results 4.4.1, this is adequate for relatively small distances between different speakers. However, for larger relative distances, it is predicted based on figure 3.4, another form of signal separation is required. The reason that less attention is placed into this issue, is because it can not adequately be tested, thus optimizing performance of the system for single codes is prioritized. A potential method for removing dominating codes from a signal, is by subtracting that code from the signal after the distances for that code are analyzed. This is something that should be looked into for the next iteration of this system. However, given that the framework for the scalability of this system does adequately function, the requirement for this system to be scalable up to 120 units is considered to be fulfilled for the prototyping purpose of this first version.

With the non-functional requirements for the system covered, the functional requirements for the propagation time estimation system should be addressed. Firstly, the duration of the estimation system should be covered. Although this time is not specifically measured, the typical time duration for a measurement using one speaker and 8 microphones including computation times is roughly 8-10 seconds and the measurement time only is roughly 2-3 seconds. In a final implementation that fully functions in parallel consisting of 120 units, the measurement time of the system is constant, however the computation time for multiple units scales an a roughly linear fashion for the computation of the data from one unit, and therefore scales quadratically for the computation of data from all units. The current system does not scale to the required computation times. However, given that the system is prototyped in MATLAB and the code is also not optimized in any way for decreasing computation time, given that functionality for the currently implemented prototype is given priority, it is expected that the propagation time estimation system can be made significantly faster. Therefore the requirements concerning the speed of the system are expected to be able to be met once the system would be implemented in the Bosch DICENTIS system, but it should be noted that this is currently not the case.

Looking at the requirement for integrateability of the designed system into the Bosch DICENTIS hardware, the system is designed such that very little extra requirements are imposed on the current Bosch DICENTIS systems software. The propagation time estimation system can work with the current Bosch DICENTIS system if it is made possible to receive individual recorded audio signals from units and if it is made possible to send individual audio signals to DICENTIS units to be transmitted. Receiving individual recordings should be possible hardware wise, given that the router of the system also receives individual recorded audio signals, thus the router only needs to relay these signals further. Concering the ability to transmit different audio signals to each DICENTIS unit, with the currently implemented FPGA on each DICENTIS unit handling transmission of audio, it is assumed that the FPGA implementation can be altered such that individual audio streams can be stored on each unit, although it should be said that this is not fully confirmed by Bosch.

5.2. Conclusion

For the development of a propagation time estimation system subsystem, a prototype has been developed that produces acceptable results under the imposed conditions in the program of requirements section concerning accuracy and range. However, given that the currently implemented prototype is not yet a final version, certain requirements are taken more into account than others.

The requirements for the system aimed for in the currently implemented protype are fulfilled. Although this is the case, for the final version of the propagation delay estimator to fully meet all posed requirements, the main issues that still need to be addressed are speed and scalability. The currently implemented prototype is not sufficiently fast to meet the requirements concerning speed for a larger set of units. Also, for individual code streams to be fully separable under extreme conditions, the subtraction of received demodulated codes from an audio signal after being analyzed should be looked into.

The main requirement focused on in the design of the currently implemented prototype is accuracy. In order to achieve the required accuracy with a standard deviation of 3 cm, the design of the system is mainly focused around robustness. For this, uncorrelated peak detection estimators are implemented to not only increase the resolution of times related to estimated peaks, but also decrease the error rate of the system. In order for the peak detectors to produce higher estimated auto-correlation peaks, repeated pseudo-noise code streams are transmitted. The implementations, as is shown in the results, have indeed resulted in an acceptably robust and accurate system. With this, the designed prototype is considered a success.

A

Appendix

A.1. Bandwidth of UniPolar and Polar Line Codes

There exist different forms of line codes. Commonly used line codes are polar (binary symbols are 1's and -1's) and unipolar (binary symbols are 1's and 0's) non return to zero (NRZ) codes. NRZ means that during the symbol time the digital signal does not go to zero, as would be the case in return to zero codes (RZ). Within this section the power spectral densities ($\mathcal{P}_s$) of the different NRZ codes will be given. RZ codes are not considered, because they contain less power in their main lobe.

A.1.1. General PSD of binary signals

Every digital signal $s(t)$ can be represented by (A.1).

$$s(t) = \sum_{n=-\infty}^{\infty} a_n f(t - nT_s) \tag{A.1}$$

where $s(t)$ is the digital signal, the set a_n is the set of random data, f(t) the symbol pulse shape and T_s the time a single bit needs. As proven in [12] the Power Spectral Density (PSD) of the signal, as posed in (A.1), can be expressed as followed:

$$\mathcal{P}_s = \frac{|F(f)|^2}{T_s} \sum_{k=-\infty}^{\infty} R(k) e^{j2\pi k f T_s} \tag{A.2}$$

Where $F(f)$ is the Fourier transform of $f(t)$, f the frequency variable and $R(k)$ is the autocorrelation of random set a_n as defined in equation A.3. In (A.3) P_i is the chance that a certain product $a_n a_{n+k}$ can occur and I the total amount of different occurrences of the same product. Using equation A.2 the Power Spectral Density of unipolar and polar NRZ signals can be found.

$$R(k) = \sum_{i=1}^{I} (a_n a_{n+k})_i P_i \tag{A.3}$$

A.1.2. Unipolar NRZ

To compute the PSD of unipolar NRZ two properties have to be examined of the signal. First the symbol pulse shape ($f(t)$) and the autocorrelation of the random set a_n.

For NRZ signals the symbol pulse shape is defined as a rectangular pulse shape. The Fourier transform of the rectangular pulse shape can be defined as in (A.4), where T_b is the symbol send time.

$$
\begin{aligned}
F(f) &= \int_{-T_b/2}^{T_b/2} 1e^{-j\omega t}dt \\
F(f) &= \frac{e^{-j\omega T_b/2} - e^{j\omega T_b/2}}{-j\omega} \\
F(f) &= T\frac{\sin(\pi f T_b}{\pi f T_b} \\
F(f) &= T\mathrm{sinc}(\pi f T_b)
\end{aligned}
\tag{A.4}
$$

With the Fourier transform of the symbol pulse shape defined, the autocorrelation of the unipolar set a_n has to be computed. Assuming that both binary symbols are equally likely to occur and that the values of the symbols can only be 1's or 0's, the autocorrelation of the random set can be computed using equation A.3. Two different cases are compared, $R(0)$ and the general $R(k)$. For $R(0)$ two possible values can occur equally likely. A symbol can be either 1 or 0, so the autocorrelation can be defined as following, with the different possibilities defined in Table A.1:

$$
R(0) = \sum_{i=1}^{2}(a_n a_n)P_i = \frac{1}{2} \tag{A.5}
$$

Table A.1: Possibilities of inter signal product $(a_n(k)a_n(k))$

Possible symbol value	Signal A: 1	Signal A: 0
Signal B: 1	1	–
Signal B: 0	–	0

When k does not equal 0, then there are four possible combinations of symbols with equal chance of occurance, resulting in the following autocorrelation with the different possibilities defined in Table A.2:

$$
R(k) = \sum_{i=1}^{4}(a_n a_{n+k})P_i = \frac{1}{4} \tag{A.6}
$$

Table A.2: Possibilities of inter signal product $(a_1(n)a_2(n+k)$

Possible symbol value	Signal A: 1	Signal A: 0
Signal B: 1	1	0
Signal B: 0	0	0

Combining (A.5) and (A.6) results in the following autocorrelation function for the unipolar signal:

$$
R(k) = \begin{cases} \frac{1}{2}, & k = 0 \\ \frac{1}{4}, & k! = 0 \end{cases} \tag{A.7}
$$

Combining (A.2) and (A.7) the PSD of a unipolar signal can be defined as followed:

$$
\mathcal{P}_{\text{Unipolar NRZ}} = \frac{T_b}{4}\Big(\frac{\sin(\pi f T_b)}{\pi f T_b}\Big)^2\Big[1 + \sum_{k=-\infty}^{\infty} e^{j2\pi k f T_b}\Big]
$$

$$
\sum_{k=-\infty}^{\infty} e^{j2\pi k f T_b} = \frac{1}{T_b}\sum_{n=-\infty}^{\infty}\delta(f - \frac{n}{T_b})
$$

Since $\mathrm{sinc}(\pi f T_b) = 0$ at f = n/T_b for n !=0 we conclude:

$$
\mathcal{P}_{\text{Unipolar NRZ}} = \frac{T_b}{4}\Big(\frac{\sin(\pi f T_b)}{\pi f T_b}\Big)^2\Big[1 + \frac{1}{T_b}\delta(f)\Big] \tag{A.8}
$$

From (A.8) it can be concluded that the first Null Bandwidth of a unipolar signal is at $\frac{1}{T_b}$, which equals the bit rate of the signal.

A.1.3. Polar NRZ

The computation of the PSD of Polar NRZ signals is very similar to the computation of the PSD of Unipolar NRZ signals. The same symbol pulse shape is used, which means that equation (A.4) also holds for Polar NRZ signals. Only the autocorrelation function changes. Again the autocorrelation will be computed using (A.3). Two different cases are compared, $R(0)$ and the general $R(k)$. For $R(0)$ two possible values can occur equally likely. A symbol can be either 1 or -1, so the autocorrelation can be defined as following, with the different combination possibilities defined in table A.3:

$$R(0) = \sum_{i=1}^{2}(a_n a_n)P_i = 1 \tag{A.9}$$

Table A.3: Possibilities of inter signal product $(a_n a_k)$

Possible symbol value	Signal A: 1	Signal A: -1
Signal B: 1	1	–
Signal B: -1	–	1

When k does not equal 0, then there are four possible combinations of symbols with equal chance of occurrence, see Table A.4, resulting in the following autocorrelation:

$$R(k) = \sum_{i=1}^{4}(a_n a_{n+k})P_i = 0 \tag{A.10}$$

Table A.4: Possibilities of inter signal product $(a_n a_k)$

Possible symbol value	Signal A: 1	Signal A: -1
Signal B: 1	1	-1
Signal B: -1	-1	1

Combining (A.9) and (A.10) results in the following autocorrelation function for the polar signal:

$$R(k) = \begin{cases} 1, & k = 0 \\ 0, & k! = 0 \end{cases} \qquad = \qquad \delta(k) \tag{A.11}$$

Combining (A.2) and (A.11) the PSD of a polar signal can be defined as followed:

$$\mathcal{P}_{\text{Polar NRZ}} = T_b\left(\frac{\sin(\pi f T_b)}{\pi f T_b}\right)^2 \tag{A.12}$$

From (A.12) it can be concluded that the first Null Bandwidth of a unipolar signal is at $\frac{1}{T_b}$, which equals the bitrate of the signal.

A.2. Demodulating two OOK signals to original polar NRZ signal

Given two modulated OOK signals $s_{\text{OOKm1}}(t)$ and $s_{\text{OOKm2}}(t)$, where $s_{\text{OOK1}}(t) + s_{\text{OOK2}}(t) = s_{\text{original}}$, the following steps can be applied to regain the original signal:

Firstly the modulated OOK signals are considered as follows:

$$s_{\text{OOKmn}}(t) = s_{\text{OOKn}}(t)\frac{1}{2}\left[e^{j(2\pi f_{cn}t+\theta)} + e^{-j(2\pi f_{cn}t+\theta)}\right] \tag{A.13}$$

Multiplying this signal with a cos gives the following result:

$$s_{\text{OOKn}}(t)\frac{1}{4}\left[e^{j(2\pi f_{cn}t+\theta)}+e^{-j(2\pi f_{cn}t+\theta)}\right]\left[e^{j(2\pi f_{cn}t)}+e^{-j(2\pi f_{cn}t)}\right]$$

$= \mathrm{s}_{\text{OOKn}}(t)\frac{1}{4}\left[e^{j(4\pi f_{cn}t+\theta)}+e^{-j(4\pi f_{cn}t+\theta)}+e^{j\theta}+e^{-j\theta}\right]$

$= \mathrm{s}_{\text{OOKn}}(t)\frac{1}{2}\left[\cos\left(4\pi f_{cn}t+\theta\right)+\cos\left(\theta\right)\right]$ (A.14)

Squaring this signal gives the following result:

$$\frac{1}{4}s_{\text{OOKn}}(t)^2\left[\cos\left(4\pi f_{cn}t+\theta\right)^2+\cos\left(\theta\right)^2+2\cos\left(\theta\right)\cos\left(4\pi f_{cn}t+\theta\right)\right] \tag{A.15}$$

Given that $s_{\text{OOKn}}(t)$ is a random signal consisting of only 1's and 0's or -1's and 0's, the above equation can also be expressed as:

$$\frac{1}{4}|s_{\text{OOKn}}(t)|\left[\cos\left(4\pi f_{cn}t+\theta\right)^2+\cos\left(\theta\right)^2+2\cos\left(\theta\right)\cos\left(4\pi f_{cn}t+\theta\right)\right] \tag{A.16}$$

Now the same steps are repeated, however the modulated OOK signal is now multiplied with a sin:

$$s_{\text{OOKn}}(t)\frac{1}{j4}\left[e^{j(2\pi f_{cn}t+\theta)}+e^{-j(2\pi f_{cn}t+\theta)}\right]\left[e^{j(2\pi f_{cn}t)}-e^{-j(2\pi f_{cn}t)}\right]$$

$= \mathrm{s}_{\text{OOK n}}(t)\frac{1}{4}\left[e^{j(4\pi f_{cn}t+\theta)}-e^{-j(4\pi f_{cn}t+\theta)}-e^{j\theta}+e^{-j\theta}\right]$

$= \mathrm{s}_{\text{OOK n}}(t)\frac{1}{2}\left[\sin\left(4\pi f_{cn}t+\theta\right)-\sin\left(\theta\right)\right]$ (A.17)

Squaring this signal gives the following result:

$$\frac{1}{4}s_{\text{OOKn}}(t)^2\left[\sin\left(4\pi f_{cn}t+\theta\right)^2+\sin\left(\theta\right)^2-2\sin\left(\theta\right)\sin\left(4\pi f_{cn}t+\theta\right)\right] \tag{A.18}$$

Given that $s_{\text{OOK n}}(t)$ is a random signal consisting of only 1's and 0's or -1's and 0's, the above equation can also be expressed as:

$$\frac{1}{4}|s_{\text{OOKn}}(t)|\left[\sin\left(4\pi f_{cn}t+\theta\right)^2+\sin\left(\theta\right)^2-2\sin\left(\theta\right)\sin\left(4\pi f_{cn}t+\theta\right)\right] \tag{A.19}$$

Now, equation A.16 and A.19 are added resulting in the following:

$$\frac{1}{4}|s_{\text{OOKn}}(t)|\Big[\cos\left(4\pi f_{cn}t+\theta\right)^2+\cos\left(\theta\right)^2+2\cos\left(\theta\right)\cos\left(4\pi f_{cn}t+\theta\right)+$$

$\sin\left(4\pi f_{cn}t+\theta\right)^2+\sin\left(\theta\right)^2-2\sin\left(\theta\right)\sin\left(4\pi f_{cn}t+\theta\right)\Big]$

$= |\mathrm{s}_{\text{OOKn}}(t)|\frac{1}{4}\left[1+1+2\cos\left(\theta\right)\cos\left(4\pi f_{cn}t+\theta\right)-2\sin\left(\theta\right)\sin\left(4\pi f_{cn}t+\theta\right)\right]$

$= |\mathrm{s}_{\text{OOKn}}(t)|\frac{1}{2}\left[1+\cos\left(\theta\right)\cos\left(4\pi f_{cn}t+\theta\right)-\sin\left(\theta\right)\sin\left(4\pi f_{cn}t+\theta\right)\right]$ (A.20)

Filtering the sin and cos in the above equation, the absolute value of the original OOK signal is regained. Given that the original OOK signal consists of either 1's and 0's or -1's and 0's, and the fact that $s_{\text{OOK 1}}(t)$ + $s_{\text{OOK 2}}(t)$ = s_{original}, $|s_{\text{OOK1}}(t)|$ - $|s_{\text{OOK2}}(t)|$ = s_{original} if $s_{\text{OOK1}}(t)$ is the signal that contains 1's and 0's and $s_{\text{OOK2}}(t)$ is the signal that contains -1's and 0's.

A.3. Discrete Gcc-Phat Derivation

The generalized cross correlation phase transform (gcc-phat) is a method, given in source [25], that makes the cross correlation output of two shifted signals a Kronecker delta function. This method can be used to minimize convolutive noise, when cross correlations are computed numerically. Within this section a derivation of the discrete gcc-phat method will be given.

Starting with two time discrete signals $x_1(n)$ and $x_2(n)$ that are shifted versions of eachother, the cross correlation is given by:

$$R_{x_1x_2}(k) = \sum_{n=-\infty}^{\infty} x_1(n)x_2(k+n) \tag{A.21}$$

$$R_{x_1x_2}(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} G_{x_1x_2}(e^{j\omega})e^{j\omega k}d\omega \tag{A.22}$$

Equation A.22 is the inverse discrete Fourier transform of the discrete Fourier transformed cross correlation ($G_{x_1x_2}(f)$). Both equations A.21 and A.22 are equal to each other.
Next, the discrete Fourier transformed cross correlation $G_{x_1x_2}(f)$ will be weighed, sample for sample, by their amplitude, resulting in:

$$\frac{G_{x_1x_2}(f)}{|G_{x_1x_2}(f)|} \quad = \quad e^{j\theta(f)} \quad = \quad e^{j2\pi fD} \tag{A.23}$$

$$R_{y_1y_2}(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j2\pi fD}e^{j\omega k}d\omega \tag{A.24}$$

$$R_{y_1y_2}(k) = \delta(k-D) \tag{A.25}$$

It is shown in (A.23) that the result of weighting the Fourier transformed cross correlation with its amplitude is a linear phase within the frequency domain due to a delay D between the two correlated signals. Equations (A.24) and (A.25) show that inverse transforming this linear phase back to the time domain results in a Kronecker delta function shifted with exactly delay D. This is why the gcc-phat is so powerful, because it minimizes convolutive noise and can make any correlation between shifted signals a delta function.

Bibliography

[1] Bosch, “Dicentis draadloos conferentie systeem gebruikers handleiding.” https://nl.boschsecurity.com/nl/producten_1/conferencesystems_7/dicentiswirelessconferenc_7/dicentiswirelessconferenc_7_18132#section_18998258059, 2017.

[2] N. Iliev and I. Paprotny, “Review and comparison of spatial localization methods for low-power wireless sensor networks,” *IEEE Sensors Journal*, vol. 15, pp. 5971–5987, Oct 2015.

[3] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, pp. 1067–1080, Nov 2007.

[4] O. Shani, “Precise time synchronization over wlan.” http://www.ti.com/lit/an/swaa162a/swaa162a.pdf, 2015.

[5] C.-D. Wann and H.-C. Chin, “Hybrid toa/rssi wireless location with unconstrained nonlinear optimization for indoor uwb channels,” in *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pp. 3940–3945, IEEE, 2007.

[6] Cisco, “802.11 fundamentals.” https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Borderless_Networks/Unified_Access/CMX/CMX_802Fund.pdf, 2014.

[7] S. Mazuelas, A. Bahillo, R. M. Lorenzo, P. Fernandez, F. A. Lago, E. Garcia, J. Blas, and E. J. Abril, “Robust indoor positioning provided by real-time rssi values in unmodified wlan networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, pp. 821–831, Oct 2009.

[8] P. Kumar, L. Reddy, and S. Varma, “Distance measurement and error estimation scheme for rssi based localization in wireless sensor networks,” in *2009 Fifth International Conference on Wireless Communication and Sensor Networks (WCSN)*, pp. 1–4, Dec 2009.

[9] C. Chen, Y. Han, Y. Chen, and K. J. R. Liu, “Indoor gps with centimeter accuracy using wifi,” in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1–4, Dec 2016.

[10] C. Chen, Y. Chen, Y. Han, H. Q. Lai, and K. J. R. Liu, “Achieving centimeter-accuracy indoor localization on wifi platforms: A frequency hopping approach,” *IEEE Internet of Things Journal*, vol. 4, pp. 111–121, Feb 2017.

[11] H. Schweinzer and M. Syafrudin, “Losnus: An ultrasonic system enabling high accuracy and secure tdoa locating of numerous devices,” in *2010 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–8, Sept 2010.

[12] L. W. II, *Digital and Analog Communication Systems, Eighth Edition*. Pearson, 2013.

[13] A. Springer, M. Huemer, L. Reindl, C. C. W. Ruppel, A. Pohl, F. Seifert, W. Gugler, and R. Weigel, “A robust ultra-broad-band wireless communication system using saw chirped delay lines,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 46, pp. 2213–2219, Dec 1998.

[14] G. Marsaglia and W. Tsang, “A fast, easily implemented method for sampling from decreasing or symmetric unimodal density functions,” *SIAM Journal on Scientific and Statistical Computing*, vol. 5, no. 2, pp. 349–359, 1984.

[15] R. Gold, “Optimal binary sequences for spread spectrum multiplexing (corresp.),” *IEEE Transactions on Information Theory*, vol. 13, pp. 619–621, October 1967.

[16] "Pseudo-noise sequences." `https://paginas.fe.up.pt/~hmiranda/cm/Pseudo_Noise_Sequences.pdf`, accessed June 17, 2018.

[17] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "Beepbeep: A high accuracy acoustic ranging system using cots mobile devices," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, (New York, NY, USA), pp. 1–14, ACM, 2007.

[18] F. Zorzi, L. Mucchi, L. S. Ronga, P. Marcocci, E. D. Re, and M. Luise, "Non coherent demodulation of bpsk secure symbols," in *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 1–6, June 2013.

[19] Bosch, "Dicentis stem microphone." `http://resource.boschsecurity.us/documents/DCNM_MICx_Data_sheet_enUS_16949333515.pdf`, 2016.

[20] R. Pal, "Comparison of the design of fir and iir filters for a given specification and removal of phase distortion from iir filters," in *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pp. 1–3, Dec 2017.

[21] D. L. Jones, *Digital Filter Structures and Quantization Error Analysis*. Rice University, Houston, Texas, 2005.

[22] "Wolframalpha computational inteligence." `http://www.wolframalpha.com/`, accessed June, 2018.

[23] J. Luo, G. Zhang, H. Xu, Y. Zhou, and W. Yang, "An energy detection based signal detection method with multiple antennas in cognitive radio," in *2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 187–191, Dec 2015.

[24] P. S. Yawada and A. J. Wei, "Performance evaluation of matched filter detection based on non-cooperative spectrum sensing in cognitive radio network," in *International Journal of Computer Networks and Communications Security*, vol. 03, pp. 442–446, Dec 2015.

[25] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, pp. 320–327, Aug 1976.

[26] S. Kay, *Fundamentals of Statistical Signal Processing: Detection theory*. Prentice Hall Signal Processing Series, Prentice-Hall PTR, 1998.

[27] F. Salahdine, H. E. Ghazi, N. Kaabouch, and W. F. Fihri, "Matched filter detection with dynamic threshold for cognitive radio networks," in *2015 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1–6, Oct 2015.