Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ens.ewi.tudelft.nl/

CAS-2012-00

M.Sc. Thesis

Underwater Ultra-Wideband Fingerprinting-Based Localization

Siavash Shakeri

Abstract

In this work a new location fingerprinting-based localization algorithm is proposed for an underwater medium by utilizing ultra-wideband (UWB) signals. In many conventional underwater systems, localization is accomplished by utilizing acoustic waves. On the other hand, electromagnetic waves haven't been employed for underwater localization due to the high attenuation of the signal in water. However, it is possible to use UWB signals for short-range underwater localization. In this work, the feasibility of performing localization for an underwater medium is illustrated by utilizing a location-based fingerprinting approach. Existing algorithms for an indoor environment are evaluated in this project for an underwater medium. These algorithms are based on a neural networks or maximum likelihood estimator. Further, we also consider a classical k-nearest neighbors (KNN) approach. In addition, by employing the concept of compressive sampling, we propose a sparsity-based localization approach for which we define a system model exploiting the spatial sparsity. Moreover, a recently proposed grid mismatching algorithm is also adapted to the current localization framework and its performance is evaluated. Finally, the performance of the proposed methods is compared with the existing fingerprinting-based localization approaches.

TUDelft

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Underwater Ultra-Wideband Fingerprinting-Based Localization

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Siavash Shakeri
born in Tehran, Iran

This work was performed in:

Circuits and Systems Group
Department of Microelectronics & Computer Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics & Computer Engineering

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Underwater Ultra-Wideband Fingerprinting-Based Localization"** by **Siavash Shakeri** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: October 2012

Chairman: __________________________________
Prof.dr.ir. Alle-Jan Van der Veen

Advisor: __________________________________
Prof.dr.ir. Geert Leus

Committee Members: __________________________________
Dr.ir. Oleg Krasnov

____
Alexandro G.S. Mancusi

Abstract

In this work a new location fingerprinting-based localization algorithm is proposed for an underwater medium by utilizing ultra-wideband (UWB) signals. In many conventional underwater systems, localization is accomplished by utilizing acoustic waves. On the other hand, electromagnetic waves haven't been employed for underwater localization due to the high attenuation of the signal in water. However, it is possible to use UWB signals for short-range underwater localization. In this work, the feasibility of performing localization for an underwater medium is illustrated by utilizing a location-based fingerprinting approach. Existing algorithms for an indoor environment are evaluated in this project for an underwater medium. These algorithms are based on a neural networks or maximum likelihood estimator. Further, we also consider a classical k-nearest neighbors (KNN) approach. In addition, by employing the concept of compressive sampling, we propose a sparsity-based localization approach for which we define a system model exploiting the spatial sparsity. Moreover, a recently proposed grid mismatching algorithm is also adapted to the current localization framework and its performance is evaluated. Finally, the performance of the proposed methods is compared with the existing fingerprinting-based localization approaches.

Acknowledgments

Although, I am the only name on the cover of this thesis, this work could not be completed without the help and support of many individuals. First and foremost, I offer my sincerest gratitude to my supervisor, Prof.dr.ir. Geert Leus who has supported me throughout my thesis with his patience and knowledge. I am honoured to have worked with you and I have learned many things through these works.

I wish to express my sincere thanks to Koen Ruskamp for his help during the data measurements. Without his guidance and patience, I could not present the simulation results provided in this work.

It is a pleasure to thank all my friends that constantly encourage me through the completion of my work. Because of them, life was more amusing and joyful.

I am profoundly thankful of my parents and my sister Moozhan. They always were present through the rough times. To them, I express my endless consideration and respect and I would like to dedicate this work to them.

Siavash Shakeri
Delft, The Netherlands
October 2012

Contents

List of Figures

List of Tables

Introduction 1

The establishment of new regulations regarding the propagation of ultra-wideband (UWB) signals in 2002 by the Federal Communication Commission (FCC) and advances in digital and analogue circuitry have opened a new window to the design and implementation of new applications. These applications employ new features that are introduced by UWB signals such as high data rates and precise ranging. Numerous localization algorithms are proposed as a result of these new features.

Despite of the advances in indoor and outdoor localization, in an underwater medium, conventional acoustic waves or optical waves are still being used. However, employing acoustic or optical waves has some drawbacks. Optical systems are generally limited to extremely short distances. On the other hand, sonar systems which use acoustic waves for localization are suffering from the low speed of underwater wave propagation, limited data rates and severe distortion by multipath propagation in shallow waters. The main reason why UWB signals are neglected for underwater localization is due to the high attenuation of electromagnetic waves in water. This problem spurred many researchers to investigate the feasibility of underwater communication via electromagnetic waves and the design of suitable antennas to achieve this goal. In [3–12], you can see a few examples of these attempts that show the feasibility of short range underwater communication. However, these experiments and simulations are mostly done in seawater. Therefore, still waters like water in a tank which are fundamentally different from the seawater due to the small changes in the environment are not considered by these works. Besides, the main focus of the earlier works is on the establishment of a communication link and no effort has been made specifically for designing localization algorithms.

In this research, localization algorithms are designed for an underwater medium. In the following section, two different localization approaches, geometric and location-based fingerprinting and the reason for selecting the location-based fingerprinting method are explained. Then, an outline of the sections and the contributions are presented.

In addition, it should be noted that this work is done in the frame of the Smart PEAS (Smart moving Process Environment Actuators and Sensors) project. In this project, local sensors in small units (size of a tennis ball) are used to report their observations (temperature, local flow field) in an underwater medium, as well as their location, to an outside control unit. The application of this project can offer process control by enabling local measurements and three-dimensional profiling of different parameters inside process equipment. Moreover, Smart PEAS will provide local information on the actual condition within certain equipment that is of importance for the development of accurate predictive process models. The main idea here is to put sensor-equipped small microelectronic devices inside certain liquid processes. Such sensors apply ultra-wideband technology for transmitting large amounts of data with very low power over relatively short distances [13]. A specific UWB antenna is designed for this project in [14] and its specifications are presented in the set-up description section.

1.1 Localization Methods

The general objective of localization systems is to estimate the unknown position of a wireless node which is known as agent based on a set of observation from the reference nodes which are placed in known positions and are known as anchors. In general, the following two localization scenarios can be distinguished:

1. Self positioning: In this case the mobile node finds its location from the signals that it receives from the anchors.
2. Remote positioning: There is a central unit which receives the signals from the anchors sent by an agent and which performs the localization to find the position of the agent.

In our current localization framework, the second scenario is considered. Therefore, the location of the sensor is estimated by the central unit. The central unit has a power supply, in contrast to our sensor which is battery-powered. Generally, there are two techniques for localization: geometric localization and fingerprinting-based localization. These two techniques are explained in this section and the short-comings of both algorithms are explained.

1.1.1 Geometric localization

This method consists of two phases. In the initial phase, some metrics are extracted from the signals that are received by the anchors. This phase is known as ranging or bearing. The second phase is known as fusion where these metrics are used for multi-lateration or multi-angulation to find the location of the agent. The metrics that can be used for localization are the received signal strength (RSS), angle of arrival (AOA), time of arrival (TOA) and time difference of arrival (TDOA). There exists a large amount of research concerning the low-complexity estimation of these metrics. The ranging method using RSS can be implemented by energy detectors, but they can only achieve a coarse resolution. Antenna arrays are required for AOA-based methods which decrease their popularity. The high accuracy and potentially low-cost implementation make TOA and TDOA based on ultra-wideband impulse radio (UWB-IR) a promising ranging method. If the propagation channel has a strong line of sight (LOS) component, then geometric localization techniques show a remarkable good performance. However, a rich multipath propagation environment and especially a non-LOS channel condition degrades the quality of the geometric methods. Therefore, it is essential to deploy the anchors carefully to ensure LOS conditions for all possible agent positions. If this is not feasible, the number of anchors should be increased or cooperative localization algorithms should be used.

The accuracy of geometric localization techniques is determined by the SNR of the RF signal, the propagation environment and the number of anchors. There is a lower bound on the required number of anchors such that an unambiguous location estimate is obtained. For instance, we need at least three anchors for a TOA-based method to have an unambiguous 2D location estimation. In this project, geometric localization has been used for an underwater medium and the localization results are not satisfactory due to the dense multipath environment and the difficulty to estimate the TOA of the direct path.

1.1.2 Location-based fingerprinting

This technique has two phases. First of all, we have an off-line phase in which the dictionary matrix is constructed. In this phase a number of signatures is gathered from the surveillance area. These signatures are location dependent and they are recorded in the central unit. In the on-line phase, the received new signature from the agent is compared with the existing signatures in the database. The location can be estimated by finding a signature in the dictionary matrix that is similar to the new received signature in the on-line phase. In this method, there is no lower bound on the required number of anchors. Thus, theoretically it is possible to locate agents unambiguously based on the observed RF signal at a single anchor. However, any change in the surveillance area will change the signature database and a new measurement campaign should be carried out for the off-line phase.

In the on-line phase, the complexity of the localization algorithms is of minor importance, since the processing can be done at the central unit. An important parameter is the grid spacing, which determines the total number of training points and the resolution of our spatial samples.

The performance of the location-based fingerprinting algorithms is commonly measured with accuracy and precision. Accuracy is the distance error between the estimated and the true location and the precision presents the percentage of position location estimates with a certain accuracy. In general, location-based fingerprinting is a promising alternative to position localization techniques, especially in harsh indoor environments, where the propagation channel does not have a strong LOS component. Therefore, due to the discussed limitations of the previous technique, in this research, the performance of the fingerprinting-based localization algorithms for an underwater medium is evaluated.

1.2 Thesis Outline and Contributions

In this section, the major contributions of this thesis are highlighted. We focus on designing underwater localization algorithms with UWB signals for the first time. For this purpose, a thorough evaluation of the existing location-based fingerprinting algorithms in the literature is performed for an underwater medium. Besides, sparse localization algorithms are proposed for fingerprinting-based localization by utilizing channel impulse responses as the location-dependent signatures.

Next, we describe the content of the thesis chapter by chapter.

Chapter 2: Channel measurement campaign To investigate the feasibility of underwater localization with UWB signals, the characteristics of the UWB channel are provided. In addition, the influence of water on the underwater wave propagation is discussed. Afterwards, the set-up that is employed to demonstrate the algorithms is described. In the last section of this chapter, a cross-correlation method for extracting the channel impulse responses and the reason for choosing this method are described.

Chapter 3: Sparse localization In this chapter, our proposed method which is based on the concept of compressive sampling is described. The system model is defined in the beginning for two scenarios. These two scenarios are based on either having or not having time synchronization between the sensor and the receiving antennas. Then, the properties the dictionary matrix should satisfy to result in a unique result are explained. In the end, recovery algorithms to estimate the sensor location are provided.

Chapter 4: Other localization algorithms Existing fingerprinting-based localization algorithms for an indoor channel are described in this chapter and they are assessed for underwater localization. More specifically, a few classical methods are described such as the k-nearest neighbours method, localization by employing a neural network and a statistical method which employs maximum likelihood estimator. Their advantages and drawbacks are explained.

Chapter 5: Simulation results Simulation results are provided in this chapter. The estimation error and the performance of the proposed methods are depicted. Besides, as real-time localization is requested for the Smart PEAS project, our algorithms are further optimized to achieve this goal. A strategy for this optimization and the related results are shown in this chapter.

Chapter 6: Conclusions and future works This chapter summarizes the major results of this work and some suggestions are provided for continuing of this work.

Channel Measurement Campaign 2

A channel measurement campaign which is designed to demonstrate the feasibility of the localization algorithms is presented in this chapter. Accordingly, some preliminary information regarding the characteristics of UWB channels and also the propagation of waves in an underwater environment are provided. In addition, different components that are used in the designed set-up are pointed out. Finally, a number of methods for the extraction of the channel impulse responses from the received signals and also for the extraction of some required parameters from the channel impulse responses are presented. As the design of localization algorithms for an underwater environment is the main focus of this work, only the channel parameters and models required for the presented methods are explained. Therefore a detailed description of the different types of channel models and parameter extraction methods are not provided. For further details see [1, 15, 16].

2.1 Preliminary Information

In this section some preliminary information regarding the UWB channel characteristics and the effect of the water environment on electromagnetic waves is explained.

2.1.1 UWB Channel

What is an UWB system? According to the federal communication commission (FCC), an UWB system is a system which occupies a bandwidth of more than 500 MHz (absolute bandwidth) or a bandwidth of more than 20% of its carrier frequency (relative bandwidth). After the introduction of UWB systems, many conventional channel models that were designed for narrowband systems aren't matched to the new UWB systems. The large bandwidth of UWB systems results in a significant difference in channel characterization. This made researchers to propose new channel models for these systems.

One of the main differences in the behaviour of an UWB channel compared to narrowband channels is due to the interactions of the multipath components with objects in the propagation environment. As the signals leave the transmitting antenna, they interact with surrounding objects. These interactions cause reflection, diffraction and scattering of the signal. Depending on the path that a signal takes it has a certain delay, attenuation and direction of arrival. The interactions between the multipath components and objects are frequency dependent. Therefore, the properties of the objects under different frequencies can't be considered constant as in the narrowband case. UWB systems are desirable for localization due to their high time resolution.

With a higher time resolution, the received signal is more dispersed and each multipath component can be distinguished easier.

Some of the key parameters to model an UWB channel are listed below.

2.1.1.1 Path loss

The path loss is defined as the ratio of the transmitted signal power P_t to the received signal power P_r averaged over any random variation of the signal. The path loss shows the amount of power dissipation of the signal, as it goes through the direct path from the transmitter to the receiver. Therefore it is possible to have an estimate of the transmitter coverage by this value. For a conventional narrowband system, this value can be written as

$$PL(d) = PL_0 + 10n \log\left(\frac{d}{d_0}\right) \tag{2.1}$$

where PL_0 is the path loss at the reference distance d_0. It should be noted that n is the path loss exponent and it relates to the propagation environment. When we consider an UWB system, the frequency dependency of the path loss should also be considered. Therefore, the path loss is proportional to

$$PL(f) \propto f^{-2\kappa} \tag{2.2}$$

where κ is the frequency decaying factor and it is also related to the geometry of the propagation environment. As this parameter has a direct dependence to the distance between the transmitter and the receiver, it can be used as one of the location dependent signatures in the fingerprinting-based localization.

2.1.1.2 Shadowing

If the receiver is kept at the same distance from the transmitter and it moves around the transmitter, it doesn't receive a constant signal and it experiences amplitude fluctuations. There is always a variation in the received signal. The source of this variation is the presence of objects that surround the receiver and the transmitter. These fluctuations happen when the transmitter (receiver) moves over distances that are multiples of the wavelength, say, 10λ. In conventional narrowband systems, it is proven that the variation of the amplitude follows a log-normal distribution. The same result also holds for UWB systems and it appears as an extra term in (2.1) and can be written in dB as

$$PL(d) = PL_0 + 10n \log\left(\frac{d}{d_0}\right) + \psi \tag{2.3}$$

where ψ is a Gaussian random variable with zero mean and standard deviation σ^2. However, this distribution has never been verified for an underwater environment.

2.1.1.3 Small-scale fading

A movement of the transmitter (receiver) over short distances causes another kind of signal fluctuations. This mostly happens if the receiver moves over very short distances, in the order of the signal wavelength. If we consider a cluster of surrounding objects which is mostly true in a realistic situation, then after the interaction of the signal with these objects, a cluster of multipath components reaches the receiving antenna. If the delay between two multipath components in this cluster is more than the inverse of the system bandwidth, then they are resolvable and two different paths can be detected. Non-resolvable components are combined into a single multipath component and delay, where the phase and amplitude correspondingly are determined by the sum of the different components. Therefore a small movement of the transmitter (receiver) causes a dramatic change in the receiving amplitude and phase. The amplitude of this summed signal will typically undergo fast variations due to the constructive and destructive combining of the non-resolvable components. Therefore, by dividing the channel impulse response by the number of bins with a width equal to the inverse of the system bandwidth, it is possible to define resolvable and non-resolvable bins. Contributions of the multipath components in each specific bin experience small-scale fading. Localization based on the statistical framework considers the statistics of the small-scale fading for localization. Further details of this localization algorithm are presented in chapter 4.

2.1.1.4 Power delay profile

The power delay profile demonstrates the received power with respect to the reference time-line. This reference time-line is mostly considered with respect to the arriving time of the first path. Due to the wide bandwidth, received signals become widely dispersed in time with respect to the transmitted signal. The power delay profile is an indicator of the degree of this dispersion. If we denote the channel impulse response as $h(t)$, the power delay profile is defined as

$$P(t) = E\{\, |h(t)|^2 \,\}. \tag{2.4}$$

Objects in the propagation area are considered in clusters. Therefore, multipath components arrive in multiple clusters with different attenuations and delays. As a result, the model that is proposed for the power delay profile is a double exponential model based on clustering :

$$h(t) = \sum_{l=0}^{L} \sum_{k=0}^{K} a_{k,l}\, \delta(t - T_l - \tau_{k,l}) \tag{2.5}$$

where $a_{k,l}$ is the tap weight of the k-th component in the l-th cluster, T_l is the delay of the l-th cluster, $\tau_{k,l}$ is the delay of the k-th multipath component relative to the l-th cluster arrival time T_l and K is the number of multipath components within the cluster and finally L is the number of clusters. Figure 2.1 shows the form of the power delay profile. For further details regarding the taps arrival time statistics, see [1]. If the channel impulse response is taken in a dense multipath environment, the delays

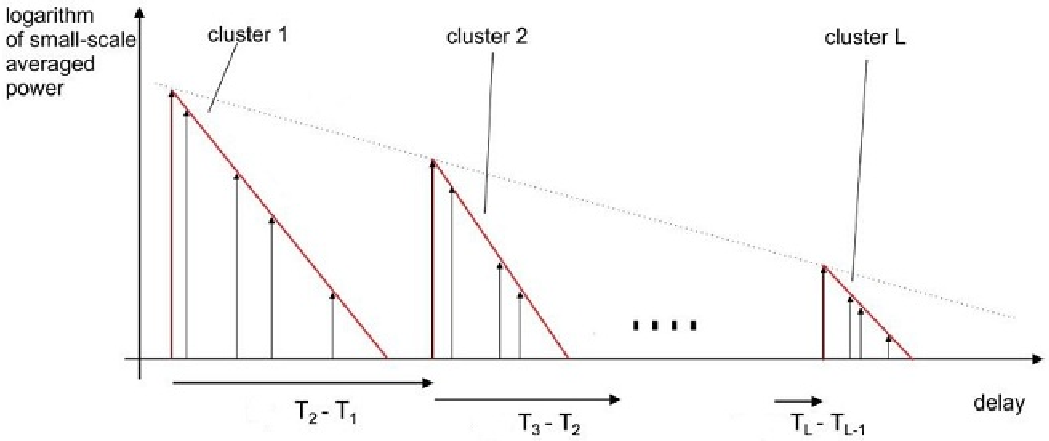


Figure 2.1: Illustration of the double exponential model for the power delay profile (the figure is taken from [1])

between the clusters are really small and the power delay profile has the form of a decaying exponential. In that case, separate clusters are not detectable when we look at the power delay profile of the dense multipath environment.

A number of parameters can be extracted from the power delay profile which more properly describe the time dispersion of the signal. As these parameters are necessary for the localization method based on the neural networks framework, each parameter is defined. In a later stage of this section, possible ways to extract these parameters from the power delay profile are also explained.

1. **Mean excess delay ($\bar{\tau}$):** The first moment of the power delay profile is defined as the mean excess delay. This value can be defined as

$$\bar{\tau} = \frac{\sum\limits_k P(\tau_k)\tau}{\sum\limits_k P(\tau_k)} \tag{2.6}$$

where $P(t)$ is the power delay profile.

2. **RMS delay spread (τ_{rms}):** The square root of the second moment of the power delay profile is known as the RMS delay spread. The RMS delay spread is a measure of the multipath spread within a given channel and it is an important parameter for characterizing time dispersion. This parameter can be defined as

$$\tau_{rms} = \sqrt{\bar{\tau^2} - \bar{\tau}^2} \tag{2.7}$$

where

$$\bar{\tau^2} = \frac{\sum\limits_k P(\tau_k)\tau_k^2}{\sum\limits_k P(\tau_k)}. \tag{2.8}$$

3. **Maximum delay spread (τ_{max}) :** The time difference between the arrival of the first path and the arrival of the last path is considered as the maximum delay spread. In other words, it is the longest path that a multipath component takes to reach the receiver. To define the first and last arrival path, a threshold which defines the noise level should be defined. The delay is calculated between the arrival of the first path and the last path above the defined threshold.

Other parameters that are extracted from the power delay profile and are used in the localization based on the neural networks framework are:

1. **Power of the first path (P_1):** This is the power of the first multipath component that reaches the receiver. Together with its corresponding delay it can be used to distinguish between a line of sight (LOS) and non-line of sight (NLOS) situation, as in a NLOS situation the magnitude of the first path is not necessarily the strongest.

2. **Total power of the received signal (P) :** This value shows the total received power above the defined threshold.

3. **Number of multipath components** (MP) : This number shows the number of received multipath signals that are above the defined threshold.

In a later stage of this section, the derivation of the power delay profile and the extraction of the required parameters from the power delay profile are explained.

2.1.2 Underwater EM wave propagation

The propagation speed of electromagnetic (EM) waves in an underwater environment in comparison with acoustic waves makes it a better option for real-time localization. The propagation speed of EM waves in an underwater medium can be defined by the Maxwell equations as

$$v = \frac{1}{\sqrt{\epsilon_r \epsilon_0 \mu}} \tag{2.9}$$

where ϵ_r is the relative dielectric permittivity, ϵ_0 is the permittivity of the vacuum ($\epsilon_0 = 8.85 \times 10^{-12}$) and μ (in water 1.256×10^{-6}) is the permeability. The high dielectric permittivity of water is the source of the difference in the propagation speed of EM waves between air and water. The permittivity depends on the temperature of the water and also on the frequency of the signals that propagate through it. In this work, a normal room temperature of 20°C is considered and the frequency goes upto 1 GHz. For these values, the relative dielectric permittivity is constant and it is equal to 81. Thus the speed of EM waves in water is 9 times slower than the speed of EM waves in air. However in comparison with the speed of acoustic waves which is equal to 1440 m/s, EM waves propagate more than 23×10^3 faster in water which makes them a good candidate for real-time localization especially for fast moving sensors. However EM waves deal with a higher signal attenuation in water, and this attenuation increases rapidly with frequency. Therefore, only for short-range localization and short-range communication EM UWB signals can be used.

The conductivity of water is also related to the dissolved salt and also to other chemicals in the water. Therefore, the conductivity is lower for pure water. For tap water, as used in our set-up, the conductivity is between 0.005 S/m to 0.5 S/m.

As the speed of the waves in an underwater environment is 9 times slower than in air, by considering the same wavelength, we can conclude that the frequency for an underwater environment is 9 times smaller.

$$f_{water} \approx \frac{f_{\mathrm{air}}}{9}$$

2.2 Description of Experimental Set-Up

In this section, the specific set-up that is used to perform the channel measurements and to demonstrate the localization is defined. Moreover different measurement equipments that are used in this set-up and their specific settings are also described.

There are many different methods for measuring and modelling a channel. One of these methods is channel sounding. The basic idea behind channel sounding is to probe a channel with a UWB signal and extract the channel impulse response from the received signal. Two techniques are introduced to perform the sounding, time-domain and frequency-domain measurements. The selection of a specific method for measuring the channel impulse response depends on the requirement of the application and the available equipment. The two above techniques are explained below and the reason for choosing the time-domain technique is explained.

In the time domain, a channel is excited by a short pulse and the received signal sampled by an oscilloscope. There are some hardware limitations in making the desired pulse very short but otherwise this technique is very simple. Thus the time-domain technique is a good candidate for real-time applications. Measurements in the frequency domain are carried out by means of a vector network analyser (VNA). A desired bandwidth, carrier frequency and spacing between two adjacent frequencies are defined by the user. According to the inputs that are entered by the user the VNA sweeps over the desired bandwidth. The sweeping is done by dividing the entered bandwidth into discrete frequencies with the spacing that is defined by the user and sending a narrowband pulse through each frequency. The output of the VNA is a complex transfer function. By applying the inverse Fourier transform (IFFT) the channel impulse response can be obtained. As the VNA sends narrowband pulses, this technique is less susceptible to noise, since there are many noise reduction algorithms available for narrowband pulses. Due to the inherent noise averaging in the frequency-domain measurements, better SNRs can be achieved. However due to the sweeping, frequency measurements take a longer time to finish. Therefore, for the considered real-time applications, this technique is not desirable. Moreover, a careful calibration and highly accurate synchronization are also needed for the frequency-domain measurements.

Selecting the best sounding technique is based on the desired requirements of our application. Real-time localization of the sensors in the underwater environment is one

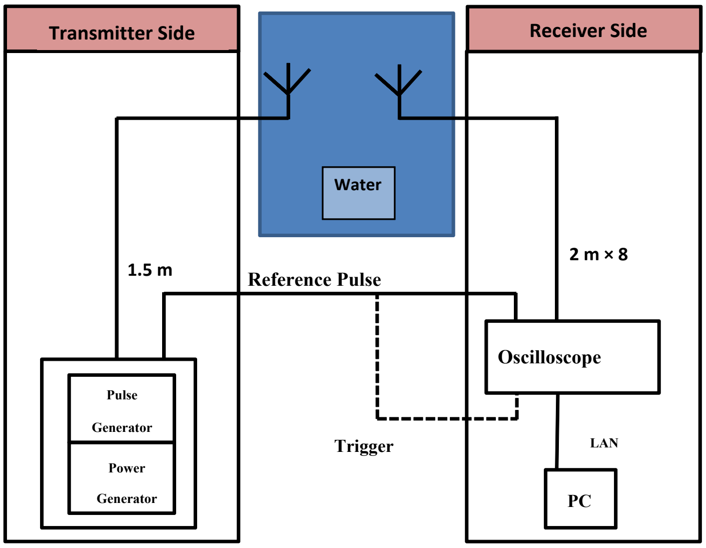


Figure 2.2: Model of the considered measurement set-up

Table 2.1: The specification of the pulse generator

Parameter Name	Value
Waveform	Approximately Delta-function
Amplitude	40 V
Polarity	Negative
Pulse Width	30 ps
Impedance	50 Ω
Additional Output : Amplitude / Rise time	1V / 30ps

of the features and requirements of the Smart PEAS project. In addition, the designed sensors are not capable of sweeping the desired bandwidth. Therefore, the time-domain technique is better matched to our needs. Thus the set-up is designed based on the time-domain measurements and its model is depicted in Fig. 2.2.

The underwater environment is simulated by a glass tank with a size of $1\text{m}\times1\text{m}\times1\text{m}$ as depicted in Fig 2.3 filled with tap water. There is also a frame inside the tank which is made out of PVC pipes and the receiving antennas are placed in the corners of this inner cubic frame. Therefore, eight receiving antennas are available. In the following, different types of equipment that are used in the set-up and their settings are explained.

2.2.1 Pulse Generator

The specifications of the pulse generator [17] that is used in this set-up are written in Table 2.1. The output of the pulse generator is shown in Fig. 2.4. The reference pulse is also taken from an additional output of the pulse generator.

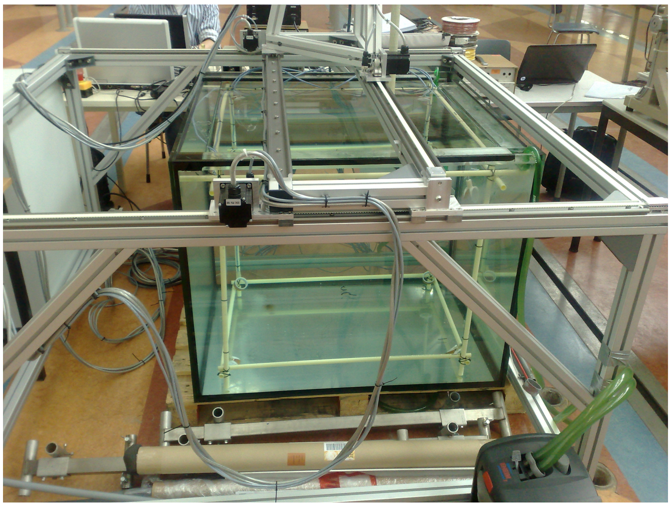


Figure 2.3: Water tank with a white inner frame containing the receiving antennas in its corners.

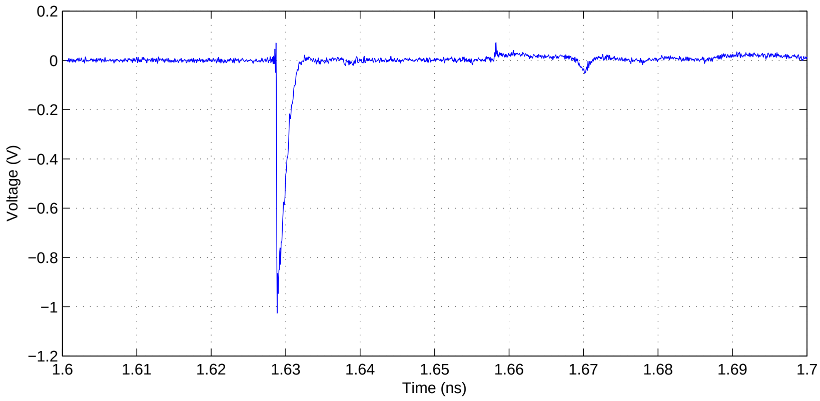
In addition, the pulse repetition frequency (PRF) should be chosen in a way to assure that all the multipath components are received before another pulse is transmitted by the pulse generator. Therefore, the pulse repetition frequency should be more than the maximum excess delay of the channel. For an indoor environment, the highest reported excess delay is equal to 900 ns [16]. However, the maximum excess delay for an underwater environment has never been reported in the literature. Therefore, a low pulse repetition frequency is chosen and it is set to 667 kHz. This results in a time window of 1.5 μs. The unambiguous range in this case is equal to

$$R = \frac{c}{9\, PRF} = 50m \tag{2.10}$$

where the 9 in the denominator appears because of the different propagation speed of EM waves in water. The unambiguous range is long enough to capture all the multipath components.

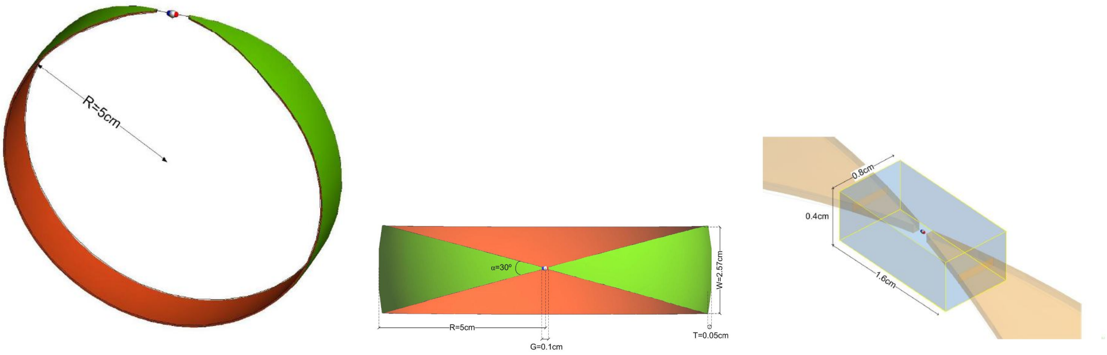
2.2.2 Antenna

An UWB antenna is designed in [14] for this project. The antenna that is used in this set-up is the first shape that is suggested for a folded bow-tie antenna. In both phases of the fingerprinting-based localization, the measurements are done with an antenna without internal isolation. The size of the antenna was also another limitation in the design. The antenna is designed with a 5 cm radius. In addition, an isolated feeding is considered by placing a free space region. The size of the feeding gap is 0.8 cm × 1.6 cm × 0.4 cm. In this design, the conductors directly touch the water because of the required small size of the antenna. The bandwidth of the antenna is 1665 MHz. The lower and the higher frequencies are 135 MHz and 1.8 GHz. It is easy to show that the



(a)

Figure 2.4: Signal from the additional output of the pulse generator which is used as a reference signal



(a) (b) (c)

Figure 2.5: The considered antenna(a)perspective view (b)top view (c)feeding gap

antenna is an UWB antenna. According to the UWB definition, it can be shown that

$$\frac{BW}{f_c} = \frac{1665}{\left(\dfrac{1800-135}{2}\right)} = 1.73 > 0.2, \tag{2.11}$$

where BW is the bandwidth and f_c represents the central frequency. For further details about the design and other proposed shapes, see [14]. Moreover, the designers wrongly assumed that the antenna rotates constantly in the water and this results in an omnidirectional antenna. However, tested results show another performance and the antenna is not omnidirectional. Therefore to simplify the set-up, the orientation of the antenna is fixed and all the measurements are done according to that orientation.

2.2.3 Oscilloscope

For the sampling unit, a real-time Agilent oscilloscope (infiniium, 6Ghz, 40 G sa/s) [18] is used. This oscilloscope has an analog bandwidth from 0 to 6 GHz. According to the Nyquist theorem the sampling frequency should be

$$f_s > 2\,BW. \tag{2.12}$$

In our set-up, the bandwidth of the system is defined by the bandwidth of the antenna and it is equal to 1665 MHz. Therefore the sampling rate should be more than 3330 MHz. In our measurements, the sampling rate is chosen approximately 6 times this value and it is set to 20 GHz which results in a high resolution. Thus in each time window with a length of 1.5 μs, the number of sampling points are

$$20\,\mathrm{GHz} \times 1.5\,\mu\mathrm{s} = 30\,\mathrm{K}. \tag{2.13}$$

2.2.4 Triggering

Two types of errors can happen in the channel measurements: short-term errors which are mostly due to thermal noise and long-term errors such as pulse drifting. If the pulse generator works for a couple of hours, the pulse might be displaced from its initial position. The variation in the pulse position is known as drifting. To avoid pulse drifting, triggering is done on the reference pulse. Therefore, as we perform the triggering on the reference pulse, the movement of the reference pulse causes a movement in the position of the received signal correspondingly. Thus pulse drifting doesn't cause any long-term errors in our set-up.

2.3 Data Extraction

After designing the set-up and gathering the received signals, it is possible to extract the channel impulse responses (CIRs). In this section a method to extract the CIRs is explained. In addition, a threshold level is defined to separate the signal from the noise.

2.3.1 Deriving the CIR

Before describing the method to extract the CIRs, it is better to mention the different components of the received signal. The impulse response of the channel is the combination of the impulse response of the true propagation channel, the impulse response of the measurement devices and the impulse response of the transmit and received antennas. Therefore, the received signal can be written as

$$r(t) = s(t) * h_{rx}(t) * h_{tx}(t) * h_{sys}(t) * h(t) \tag{2.14}$$

where $*$ is the convolution operation, $s(t)$ is the signal sent by the transmitter, $h_{rx}(t)$ and $h_{tx}(t)$ are the antenna impulse responses, $h_{sys}(t)$ is the impulse response of the measurement system and $h(t)$ is the propagation channel impulse response. In this

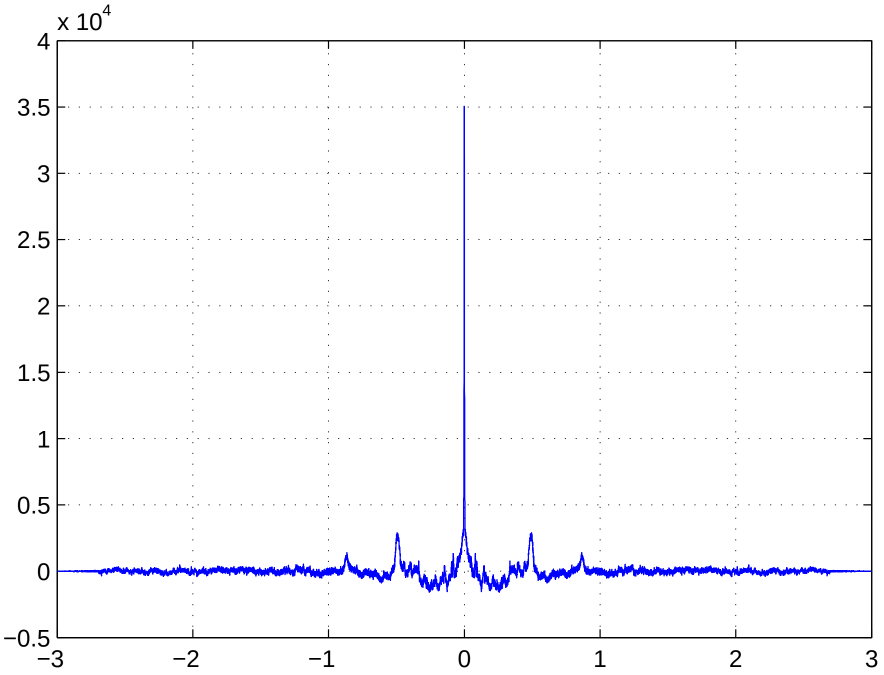


Figure 2.6: Autocorrelation of the transmitted signal

work, as the measurements are performed with the same set-up for the on-line and the off-line phases, calibration of the measurement equipment is not needed. Besides, as the antennas are not omnidirectional and removing the impulse response of the antenna is a time consuming task, the impulse responses of the channel is considered in combination with the antenna impulse response and antenna calibration is not performed.

Different methods are proposed to extract the channel impulse responses from the received signals. One of these methods is a time-domain method which is based on the high resolution CLEAN algorithm [19]. Another method is the frequency-domain approach which is known as inverse filtering. By applying the FFT function to (2.14), the convolution changes to multiplication. Therefore, the transfer function of the channel which is the division of the FFT of the received signal to the FFT of the transmitted signal is derived. By applying the inverse Fourier transform (IFFT) to the resulting transfer function, the channel impulse response is extracted. Another method extracts the CIR, by cross-correlating the received signal with the transmitted signal. In [20], it is proved that if the autocorrelation of the transmitted signal approximates a delta function, the cross-correlation of the received signal with the transmit signal results in a good approximation at the impulse response of the channel.

As the speed and simplicity of the CIR extraction method are of our interest, the cross-correlation method is chosen in this work. The autocorrelation of the transmit signal is depicted in Fig 2.6. As the autocorrelation of the transmitted signal is close to a delta function, the cross-correlation of the received and the transmitted signal approximates the channel impulse response. In Fig. 2.7, the absolute CIR value that

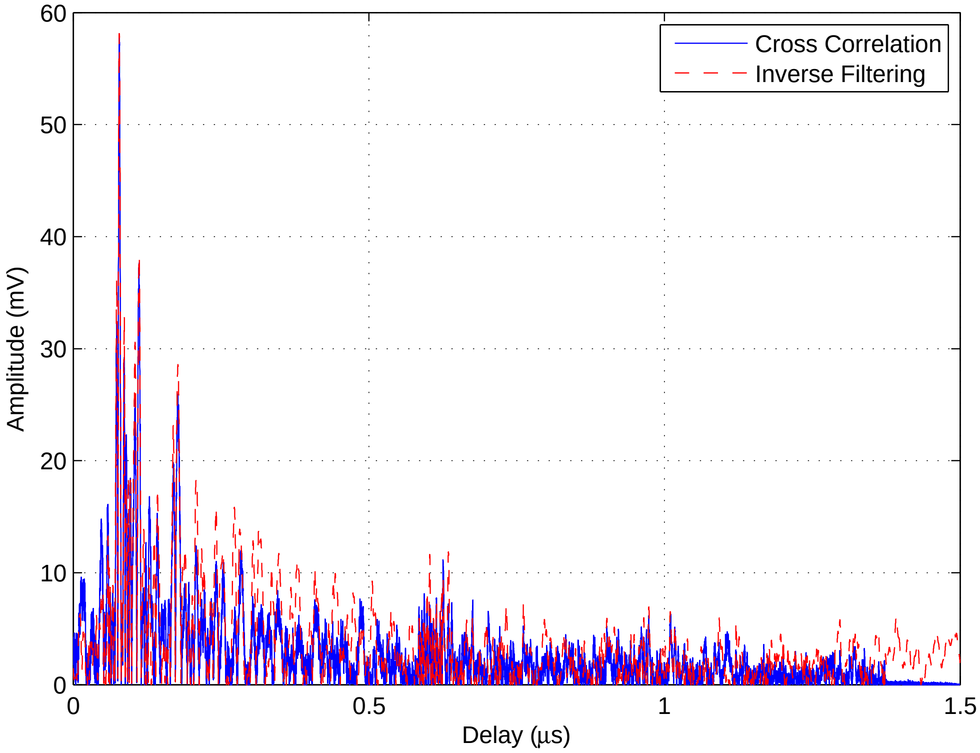


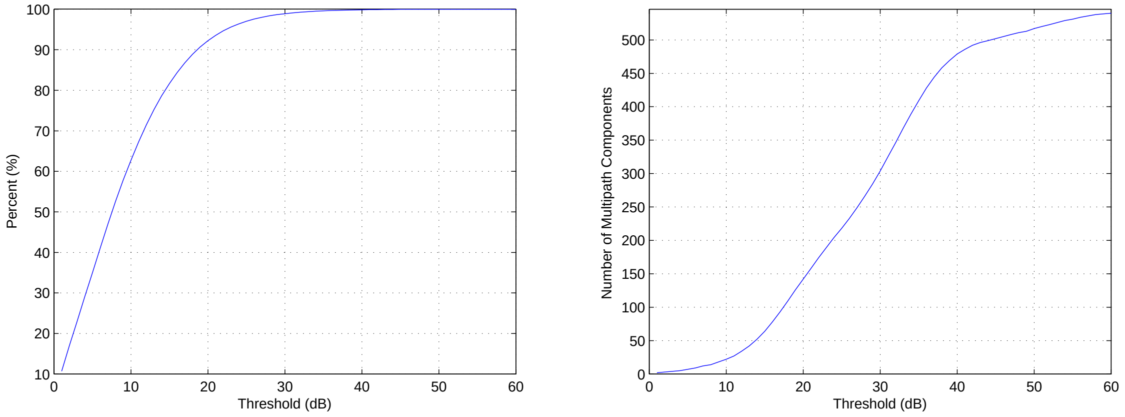
Figure 2.7: Absolute value of the CIR for comparison of the inverse filtering with the cross-correlation method for extracting the channel impulse response

results from the cross-correlation method is compared with the inverse filtering method. Almost similar results are obtained for both methods.

To find the number of multipath components, the maximum detection (MD) method is employed as it is mentioned in [20]. In this method, the power delay profile is divided into a number of bins with length equal to the inverse of the system bandwidth. For each bin, a path is present if the local maximum in that bin is above a specific threshold. In the following section, a method to select an appropriate threshold level is explained.

2.3.2 Defining the Threshold

The threshold level has a significant influence on the value of the received power, number of multipath components and the noise level. The value of the threshold is defined as a level in decibel below the strongest path in the power delay profile. To find an appropriate threshold level, the percentage of the received power above the threshold to the total received power and the number of multipath components are defined for different threshold levels. This result is averaged over 4096 CIRs from different antennas and different positions in the water tank and the results are shown in Fig 2.8. By evaluating the results, for a threshold level of 25 dB, the considered received power is 97% of the total received power. Besides the number of multipath



(a) (b)

Figure 2.8: (a) The percentage of the received power above the threshold to the total received power (b) Number of multipath components versus different threshold levels

components received for this threshold is 157. If we select a higher threshold level as 40 dB, the difference in the number of multipath components is 214 and the difference in the power value is 2% of the total received power. By evaluating the difference in the power values of the two selected thresholds, it can be concluded that the excess number of multipath components in the 40 dB case does not contain a significant amount of power and these paths can be ignored. Thus, 25 dB is selected as a threshold level in this work.

3 Sparse Localization

Sparse localization is a new approach for fingerprinting-based localization. The idea starts by introducing a new concept in sampling theory which is based on the assumption that our signal can be represented as a sparse vector. A sparse vector is a vector with a few non-zero elements. Therefore, instead of uniformly sampling the signal at the Nyquist rate and then compressing the samples based on their content, we are looking for an algorithm which randomly compresses the signal independent of the content and then extracts the non-zero coefficients from the compressed measurements. As a result, compressing and sampling are done simultaneously for a sparse signal. Under the sparsity assumption, many sparse recovery algorithms have been proposed. For example, many works by Candes, Romberg, Tao [21–25] and Donoho [26] have shown the feasibility of recovering a sparse signal from a small set of linear, non-adaptive measurements. In addition, some conditions are defined for the sampling matrix to guarantee the correct recovery of the sparse signal. Now the question that may come to mind is how we can assume sparsity in our present localization framework. This chapter starts with defining the system model to answer this question. Then the necessary properties of the dictionary matrix are explained. Next, some of the sparse recovery algorithms are introduced. At the end, the sparse total least-squares algorithm is explained for grid matching to improve the localization performance.

3.1 System Model

The system model is introduced in this section to define how sparsity can be introduced in our localization framework. For a simpler description of the system model, this section is split into two parts. The off-line phase where the dictionary matrix is constructed and an on-line phase where the sparse recovery algorithms are applied.

3.1.1 Off-line phase

In this phase, location-dependent signatures are gathered for constructing the dictionary matrix. In our approach, the gathered signatures are channel impulse responses (CIRs). The dictionary matrix $\mathbf{A}$ can be defined as

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N] \quad MJ \times N \tag{3.1}$$

where N shows the number of regions and the i-th signature vector $\mathbf{a}_i$ is derived as follows

$$\mathbf{a}_i = \frac{1}{S}\sum_{s=1}^{S}\mathbf{a}_{is} \quad MJ \times 1 \tag{3.2}$$

where S represents the total number of signature samples extracted from the i-th region. The signature vector can be defined as

$$\mathbf{a}_{is} = [\,\mathbf{CIR}_{is1}^{T}, \mathbf{CIR}_{is2}^{T}, ..., \mathbf{CIR}_{isJ}^{T}\,]^{T} \quad MJ \times 1 \tag{3.3}$$

where J shows the number of receiving antennas. The $\mathbf{CIR}_{ikj}$ vector contains the channel taps of the j-th receiving antenna and it can be written as

$$\mathbf{CIR}_{isj} = [\,h_{isj}(0),\ h_{isj}(1), ...,\ h_{isj}(M-1)\,]^{T} \quad M \times 1\ . \tag{3.4}$$

where M is the number of channel taps. The location coordinates of the grid points are put in a location matrix $\mathbf{L}$ as

$$\begin{aligned} \mathbf{L} &= [\,\mathbf{l}_1,\ \mathbf{l}_2,\ ...,\ \mathbf{l}_N] \quad 3 \times N \\ \mathbf{l}_i &= [a_i, b_i, c_i]^{T} \end{aligned} \tag{3.5}$$

where $\mathbf{l}_i$ is the location of the i-th grid point corresponding to the i-th column of the $\mathbf{A}$ matrix and a_i, b_i and c_i refers to the 3D location coordinates. One of the things that should be noted is that in contrast with compressive sampling, in this work, an overdetermined set of linear equations is considered. This means that for $\mathbf{A}$, we generally have $MJ \gg N$. The water tank that is described in the previous section is divided to 128 regions therefore N is equal to 128. In addition, we gathered 30 K channel taps for each CIR and we have 8 receiving antennas therefore the number of rows in the dictionary matrix is 240 K.

3.1.2 On-line Phase

In this phase, the sensor location can be estimated by matching a new signature vector that is extracted from the received signal of the sensor to one of the recorded signature vectors of the dictionary matrix. The received signature vector can be defined as

$$\begin{aligned} \mathbf{y} &= [\,\mathbf{CIR}'^{T}_{1}, \mathbf{CIR}'^{T}_{2}, ..., \mathbf{CIR}'^{T}_{J}\,]^{T} \quad MJ \times 1 \\ \mathbf{CIR}'_{i} &= [\,h'_i(0),\ h'_i(1), ...,\ h'_i(M-1)\,]^{T} \quad M \times 1. \end{aligned} \tag{3.6}$$

where $\mathbf{CIR}'$ is used to show that the extracted CIRs are not exactly similar to the dictionary matrix signature vectors. In this phase, the received signature can be approximated by a linear combination of the signature vectors of the dictionary matrix. Therefore, the system model can be written as

$$\mathbf{y} = \sum_{i=1}^{N} \mathbf{a}_i\, x_i + \mathbf{n} = \mathbf{A}\mathbf{x} + \mathbf{n} \tag{3.7}$$

where in this equation $\mathbf{a}_i$ defines the i-th column of the dictionary matrix, x_i is the i-th element of the $\mathbf{x}$ vector and $\mathbf{n}$ is an additive noise vector. The $\mathbf{x}$ vector which is also known as the coefficient vector can be considered as an approximately k-sparse vector.

An approximately k-sparse vector is a vector with at most k large elements and $N-k$ small or zero values where $k \ll N$. We can conclude that only k non-zero elements of the $\mathbf{x}$ vector in (3.7) are taking part in the construction of the new signature vector. The set of all k-sparse vectors can be defined as

$$\Sigma_k = \{\mathbf{x} : \| \mathbf{x} \|_0 \leq k\} \tag{3.8}$$

where $\| \mathbf{x} \|_0$ is the l_0-norm which refers to the number of non-zero elements in the $\mathbf{x}$ vector. If the sensor is exactly on the grid point, then k is equal to 1 and we have a 1-sparse vector. Otherwise, k is more than 1 due to leakage. Therefore the idea behind sparse recovery is to find the k non-zero elements of $\mathbf{x}$ based on the received signature vector. The k non-zero coefficients and their corresponding grid point locations can be employed to estimate the sensor location. If an index set Λ is defined as

$$\Lambda = \{\Lambda_1, \Lambda_2, ..., \Lambda_K\} \tag{3.9}$$

where the i-th member of this set is

$$\Lambda_i = \underset{i}{\operatorname{argmax}} \; x_i,$$

the location of the sensor can be estimated by a weighted average as

$$\hat{\mathbf{l}} = \frac{\sum\limits_{i=1}^{K} \mathbf{l}_{\Lambda_i} \, x_i}{\sum\limits_{i=1}^{K} x_i} \tag{3.10}$$

where $\mathbf{l}_{\Lambda_i}$ denotes the column of the location matrix in (3.5) which is selected by the i-th member of the index set. In (3.10), the coefficient values of the k non-zero elements in $\mathbf{x}$ are used as weights. This is due to the fact that the coefficient value depicts how close the sensor is to the corresponding grid point. As the performance of the algorithm is important for us, for simplicity, localization is performed for one sensor only. However, the algorithms proposed in this section also work for localizing multiple sensors.

In addition, as far as time synchronization between the receiving antennas and the sensor is concerned, it is possible to consider two cases :

1. **Time synchronization** : In this case, it is assumed that the receiving antennas and the transmitter are time synchronized in both phases. Therefore, the location of the sensors can be defined by applying a sparse recovery algorithm to (3.7) and then perform the above procedure to estimate the sensor location. However, sometimes there are some hardware limitations in synchronizing the receive antennas and the sensor.

2. **No time synchronization** : Assume we have no time synchronization between the receiving antennas and the transmitter. Without any temporal information, the objective is to identify a reference channel tap in (3.4), estimate it for each measured CIR and shift the time axis accordingly such that all measured CIRs

have a common time reference. Such a reference tap could be, for instance, the tap with the maximum absolute value of the CIR or a sophisticated temporal alignment strategy could be considered by identifying the TOA of the CIRs. The interested reader is referred to [27–29] for a detailed discussion of various methods with different complexities for TOA estimation. However, due to the dense multipath environment and the drawbacks of the TOA estimation this alignment strategy is not used in our work.

The alignment method that is introduced by [30,31] is employed in our localization framework. In this method, for the off-line phase before stacking the CIRs that are received by the antennas for each region in the columns of the dictionary matrix in (3.1), we find the index of a tap with the maximum absolute CIR value for all the CIRs. Therefore, we define an index set as follows

$$\Gamma = \{\Gamma_{111}, \Gamma_{112}, ..., \Gamma_{NJS}\} \tag{3.11}$$

with

$$\Gamma_{ijs} = \underset{n}{\operatorname{argmax}}\ |h_{isj}(n)|, \quad 1 \leq\ j\ \leq J,\ 1 \leq\ s\ \leq S,\ 1 \leq\ i\ \leq N \tag{3.12}$$

where Γ_{ijk} is the ijk-th member of the index set and it refers to the index of a tap where the maximum absolute CIR value of the k-th signature sample of the j-th antenna in the i-th region occurs. In addition, two other values are also recorded in the off-line phase. These values are the minimum and maximum tap indexes in the index set. These values can be obtained as

$$\Gamma_{max} = \max_{i,j,s}\ \ \Gamma_{ijs} \tag{3.13}$$

and

$$\Gamma_{min} = \min_{i,j,s}\ \ \Gamma_{ijs}.\ \ . \tag{3.14}$$

After finding the reference tap, the CIR vectors are shifted as follows

$$\mathbf{CIR}_{isj} = [\ h_{isj}(\Gamma_{ijs} - \lceil\frac{M}{\Gamma_{min}}\rceil), ...,\ h_{isj}(\Gamma_{isj}), ...,\ h_{isj}(\Gamma_{isj} + \lfloor\frac{M}{\Gamma_{max}}\rfloor)\ ]^T. \tag{3.15}$$

where $\lfloor x \rfloor$ denotes the largest integer not greater than x, $\lceil x \rceil$ denotes the smallest integer not less than x. Γ_{min} and Γ_{max} are recorded to define the number of taps that should be remained before and after the index defined by the index set. With the above method, all the CIRs are aligned to a similar tap and these aligned CIRs are used to build the dictionary matrix similar to (3.1).

In the on-line phase the received signature is similar to (3.6). To align the CIRs of the new signature vector, the index of the tap where the maximum value of the

CIR occurs is stacked into a new index set Γ_y similar to (3.12). This set can be defined as

$$\Gamma_y = \{\Gamma_{y1}, \Gamma_{y2}, ..., \Gamma_{yJ}\} \tag{3.16}$$

and

$$\Gamma_{yj} = \underset{n}{\operatorname{argmax}} \ |h_{yj}(n)|, \quad 1 \leq \ j \ \leq J \tag{3.17}$$

where Γ_{yj} denotes the index of the sample where the maximum absolute value of the CIR received by the j-th antenna occurs. The CIRs are shifted similar to (3.15) by employing Γ_{min} and Γ_{max} that are recorded in the off-line phase. Therefore, (3.6) changes to

$$\mathbf{CIR'}_j = [\, h'_j(\Gamma_{yj} - \lceil\frac{M}{\Gamma_{min}}\rceil), ..., \ h'_j(\Gamma_{yj}), ..., \ h'_j(\Gamma_{yj} + \lfloor\frac{M}{\Gamma_{max}}\rfloor)\,]^T \quad .$$

After this alignment, the system model defined in (3.7) is valid for the aligned signature vectors and by applying sparse recovery algorithms, we can estimate the sensor location similar to (3.10).

The idea of using spatial sparsity for fingerprinting-based localization is utilized for the first time in [32] and it is followed by [33–35]. In these approaches, the received signal strengths are gathered by a number of WiFi access points. These values yield the dictionary matrix for the off-line phase. In the on-line phase, sparse recovery algorithms are applied to detect the location of the users.

3.2 Dictionary Matrix Properties

Different conditions have been proposed for the dictionary matrix to ensure the uniqueness of the recovered sparse vector. These conditions are spark, null space property and restricted isometry property (RIP). As the RIP condition is a tighter condition for the dictionary matrix, in this section, this condition is explained and it is shown that the dictionary matrix in (3.1) satisfies this property.

To explain the RIP, let us define a set Σ_k as in (3.8) which contains all the possible k-sparse vectors. Further note that a linear combination of two k-sparse vectors is not necessarily a k-sparse vector. This is due to the different positions of the non-zero elements in these vectors. However, the resulting vector is definitely a member of Σ_{2k}. Assume now that the recovery algorithm recovered two different k-sparse vectors $\mathbf{x}$ and $\mathbf{x}'$ from the same received signal vector $\mathbf{y}$. As a result, we obtain

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}' \ \therefore \ \mathbf{A}(\mathbf{x} - \mathbf{x}') = 0.$$

The meaning of this equation is that there exists some set of columns of the $\mathbf{A}$ matrix that are linearly dependent. Therefore for having a unique output, no $2k$-sparse vector should exist in the null space of $\mathbf{A}$. This condition can be realized by defining the spark of a matrix. Although the spark condition assures the uniqueness of the result, this condition doesn't consider the effect of noisy received signals on the sparse recovery performance. This noise can be considered as an additive noise and can be written as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \tag{3.18}$$

where this additive noise may effect the recovery of the sparse vector. The important thing is that the distances between the sparse vectors in the null space of $\mathbf{A}$ are preserved after the transformation by the $\mathbf{A}$ matrix. This guarantees that with additive noise, sparse vectors that are far away don't end up in the same noisy received signal after transformation by $\mathbf{A}$. Therefore Candes and Tao introduced the restricted isometry property (RIP) in [24]. A matrix $\mathbf{A}$ satisfies the RIP of order k if

$$(1-\delta)\| \mathbf{x} \|_2^2 \leq \| \mathbf{A}\mathbf{x} \|_2^2 \leq (1-\delta)\| \mathbf{x} \|_2^2 \tag{3.19}$$

where $\delta \in (0, 1)$ and $\mathbf{x} \in \Sigma_k$. This property ensures (for a small enough δ) that all the sub-matrices of $\mathbf{A}$ consisting of k columns are close to an isometry and therefore norm-preserving. This means that if $\mathbf{A}$ supports the RIP of order $2k$, the $\mathbf{A}$ matrix preserves the distance between k-sparse vectors which results in robustness to noise and the uniqueness of the recovered sparse vectors.

It is computationally complex to define that the dictionary matrix satisfies the RIP. Therefore, coherence which is a simpler and a computable definition is defined by Donoho and Elad in [36].

Coherence can be defined as a largest absolute inner product between any two columns of the dictionary matrix. It can be written as

$$\mu(\mathbf{A}) = \max_{1\leq i,j\leq N \; i\neq j} |\mathbf{a}_j{}^T \mathbf{a}_i| \tag{3.20}$$

where $\mathbf{a}_i$ is the i-th column of the $\mathbf{A}$ matrix. Therefore, for a unique result, k should be :

$$k < \frac{1}{2}\left(1 + \frac{1}{\mu\left(\mathbf{A}\right)}\right). \tag{3.21}$$

The coherence of the dictionary matrix that is built in this work is 0.0381. Therefore, by considering the above definition k should be less than 13 for having a unique result. The number of non-zero elements of the coefficient vector that is considered in this work is always less than this value. Therefore, the dictionary matrix assures the uniqueness of the result.

3.3 Recovery Algorithms

As the dictionary matrix ensures the property we want for the uniqueness of the recovered sparse vector, in this section we explain three algorithms which make this recovery possible. This section starts with an explanation of the general sparse recovery problem. Based on that, three known recovery algorithms are presented. As we are having an overdetermined set of linear equations and a full column rank dictionary matrix, the trivial solution to recover the sparse vector is to use the least-squares (LS) approximation. However, LS doesn't result in a sparse solution. First, the least-squares approximation is explained. Then the orthogonal matching pursuit (OMP) is presented which is a greedy algorithm but its simplicity and speed in finding the sparse coefficient vector is more of our interest. The last algorithm is LASSO which replaces the sparse approximation problem with a linear programming problem.

3.3.1 General Sparse Recovery Problem

The idea behind sparse recovery algorithms is to find the best approximation of the received signal by finding the best set of k linear combinations from the columns of the dictionary matrix. Therefore, by looking at (3.7), we would like to find the best approximation of $\mathbf{y}$ by knowing that the coefficient vector $\mathbf{x}$ is a k-sparse vector. For an overdetermined set of linear equations, a common solution is to use the least-squares approximation which is also known as l_2-norm minimization. However, the resulting vector is almost never sparse. Now, if we consider that the l_0-norm ($\| \, . \, \|_0$) shows the number of the non-zero elements in the coefficient vector, the following optimization problem can be defined

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \| \, \mathbf{x} \, \|_0 \;\; \text{subject to} \;\; \mathbf{y} = \mathbf{A}\mathbf{x} \tag{3.22}$$

where it tries to find the sparsest coefficient vector which approximates the received signal. Since this optimization problem should check all the subsets with size k of the dictionary matrix for the optimum solution, it is computationally complex. In addition, due to the presence of the zero norm, this problem is a non-convex problem and can't be solved in polynomial time (NP hard). Therefore, the goal is to find an alternative to this optimization problem which can be solved in polynomial time.

A convex relaxation for (3.22) can be found by replacing the l_0-norm with the l_1-norm. Since the l_1-norm is convex, problem (3.22) changes to a convex problem which can be implemented as a linear program. The new optimization problem based on the l_1-norm is known as basis pursuit (BP) and it can be defined as

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \| \, \mathbf{x} \, \|_1 \;\; \text{subject to} \;\; \mathbf{y} = \mathbf{A}\mathbf{x}. \tag{3.23}$$

It is also possible to take noisy received signals into account, similar to the system model (3.18), by replacing the above problem with

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \| \, \mathbf{x} \, \|_1 \;\; \text{subject to} \;\; \| \, \mathbf{y} - \mathbf{A}\mathbf{x} \, \|_2 \leq \epsilon. \tag{3.24}$$

where the noise power is assumed bounded $\epsilon \geq \| \, \mathbf{n} \, \|_\mathbf{2}$. This problem is known as basis pursuit de-noising (BPDN).

3.3.2 Least-squares approximation

Least-squares is a standard approach to find the approximate solution of an overdetermined set of linear equations. This algorithm finds a vector in the translated null space of $\mathbf{A}$ with the smallest l_2-norm. The closed-form solution is given by

$$\hat{\mathbf{x}} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y} = \mathbf{A}^{\dagger}\mathbf{y}. \tag{3.25}$$

However, LS almost never finds a k-sparse solution and it returns instead a non-sparse solution. Besides, the results of the l_2-norm minimization are not satisfactory mainly due to two main reasons. The first one is the prediction accuracy of the LS method. The LS method often has a low bias but a large variance. The second problem is the interpretation. Sometimes, by having a large number of signatures in the $\mathbf{A}$ matrix, we only want to know the subset which has a significant effect on the output values.

These two problems are addressed in the statistical community by introducing two linear regression methods: the subset selection method and the shrinkage method. The prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero. By doing so we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence this may improve the overall prediction. The orthogonal matching pursuit can be seen as the counterpart of the subset selection method in the signal processing community. The LASSO algorithm can be realized as a hybrid solution that combines the subset selection method with the shrinkage method.

3.3.3 Orthogonal Matching Pursuit (OMP)

Orthogonal matching pursuit is one of the iterative greedy algorithms which is a modification of matching pursuit. Orthogonal matching pursuit is introduced for the first time to the signal processing community by [37,38]. The counterpart of this algorithm in the statistical community is known as forward-stagewise regression which is a regression method with a subset selection [2]. This algorithm tries to select at each step the column of the dictionary matrix which has the highest correlation with the residual of the received signal. Then a new approximation of the received signal by the selected columns of the dictionary matrix is made and it is subtracted from the residual of the previous step.

The OMP algorithm can be described as follows:

1. It starts by assigning a zero to the approximation vector $\hat{\mathbf{x}}$ and the residual is equal to the received signal vector.

   $$\hat{\mathbf{x}}^0 = \mathbf{0} \quad \mathbf{r}^0 = \mathbf{y} \quad \boldsymbol{\Lambda}^0 = \emptyset$$

   where $\boldsymbol{\Lambda}^0$ shows an initial state of an index set.

2. At the i-th step, it finds the column of the dictionary matrix with the highest correlation with the residual of the received signal. Then it adds the index of that column to the active index set $\boldsymbol{\Lambda}$. This can be implemented by finding the estimate of the coefficient vector and adding the index of the element with the highest value to the index set. This can be written as

   $$\mathbf{e}^i = \mathbf{A}^T\mathbf{r}^{i-1}$$

Then

$$\Lambda^i = \Lambda^{i-1} \bigcup \text{support}\,(H(\mathbf{e}^i))$$

where H is a thresholding operator which only keeps the largest entry of the $\mathbf{e}^i$ vector and sets all the other entries to zero.

3. In this step, the coefficient vector is estimated again by the new active set. The algorithm uses the least-squares approximation to estimate the approximation of the coefficient vector. This can be done by setting all the columns of the $\mathbf{A}$ matrix to zero except for the columns that are members of the active set. This is shown as

$$\hat{\mathbf{x}}^i = \mathbf{A}^{\dagger}_{\Lambda^i}\mathbf{y}$$

where $\mathbf{A}_{\Lambda}$ is a matrix with non-zero columns for the indexes defined by the active set. This step is the difference between the matching pursuit algorithm and the orthogonal matching pursuit. The least-squares estimation improves significantly the rate of convergence of the algorithm [39].

4. The last step is to find the new residual by subtracting the estimated received signal which resulted from the estimated coefficient vector from the initial received signal vector:

$$\hat{\mathbf{r}}^i = \mathbf{y} - \mathbf{A}\hat{\mathbf{x}}^i.$$

The stopping point for this algorithm can be set by the number of iterations or by setting a limit on the error between the estimated received signal with the initial received signal. Different conditions to ensure the convergence of the OMP algorithm and the required number of iterations are investigated by Tropp in [39]. Moreover the performance of this algorithm under the RIP condition is investigated in [40].

A similar approach but with a different implementation is proposed in [41]. In this paper, a region of 1m×1m is divided into 10000 grid points and the localization is done by using one receiving antenna. A resolution of 2 cm is achieved with a 7.5 GHz bandwidth and 18.8 cm with 500 MHz bandwidth.

3.3.4 Least Absolute Shrinkage and Selection Operator (LASSO)

LASSO sets a limit on the coefficient vector sparsity and it is an optimization principle more than an algorithm . This method is explained next.

LASSO is introduced for the first time by Tibshirani in [42] to the statistical community for solving a regression problem. As OMP (Forward-Stagewise Regression in the statistical community) is a discrete process which either selects or rejects dictionary columns, it causes a high variance. To solve this problem, a shrinkage method is introduced by imposing a penalty on the coefficient vector size. Therefore the effect of the selected column at each step is not fully taken into account and each column participates by its significance in the regression problem. Therefore, based on this idea, the ridge regression principle is introduced. In ridge regression, the penalty is on the total power of the coefficient vector (l_2-norm). LASSO chooses another penalty based on the l_1-norm and it defines the problem as follows

$$\underset{\mathbf{x}}{\operatorname{argmin}} \; \|\,\mathbf{y} - \mathbf{A}\mathbf{x}\,\|_2^2 \;\; \text{subject to} \;\; \|\,\mathbf{x}\,\|_1 \leq \text{t}. \tag{3.26}$$

where t is a tuning parameter and it defines the amount of shrinkage. There is a direct relation between the ϵ parameter in the BPDN problem as explained in (3.24) and the t parameter in the LASSO problem. Therefore, it is possible to demonstrate that solving one problem is equivalent to solving the other problem (for further details see [43]). By choosing small t values, some of the elements in the coefficient vector are set to zero. With this penalty, sparse solutions can be achieved. The Lagrangian form of this problem changes (3.26) to an unconstrained form as follows

$$\hat{\mathbf{x}} = \underset{x}{\operatorname{argmin}} \; \frac{1}{2}\|\,\mathbf{y} - \mathbf{A}\mathbf{x}\,\|_2^2 + \lambda\|\,\mathbf{x}\,\|_1 \tag{3.27}$$

where λ has a direct connection with the t parameter in (3.26) and it defines the amount of shrinkage.

By considering the geometry of the LASSO problem, the objective function in (3.26) is a quadratic function and it can be interpreted as an elliptical contour centred around the full least-squares estimate. This can be written as

$$(\mathbf{y} - \mathbf{A}\mathbf{x})^T(\mathbf{y} - \mathbf{A}\mathbf{x}) = (\mathbf{x} - \mathbf{x^0})^T\mathbf{A}^T\mathbf{A}(\mathbf{x} - \mathbf{x^0}) \tag{3.28}$$

where $\mathbf{x^0}$ is the full least-squares estimate. If we consider that the coefficient vector has only two elements, the constrained part of the problem defines a diamond as in Fig. 3.1. The solution to this problem is the point where the smallest contour touches the diamond. Due to the cornered shape of a diamond, the first touch most likely occurs at the corner of the diamond and the result is a sparse result.

For finding the LASSO solution the Least Angle Regression (LARS) is described in [2]. On the other hand, many convex optimization algorithms are available to solve this optimization problem.

3.4 Grid Matching

In this section, the performance of the localization algorithm is further improved by considering the grid matching method proposed in [44]. If our sensor is positioned exactly at one of the grid points, the estimated coefficient vector has one non-zero element exactly at the index related to the grid point and the rest of the coefficients would be significantly small. Now consider the situation where our sensor is not positioned at one of the defined grid points. This grid mismatch causes the $\mathbf{x}$ vector to have also significant values for some of the neighbouring grid points around the sensor location. We can use the information of these indexes to improve our localization performance. In this section, sparse total least squares algorithm [44] is utilized for grid matching.

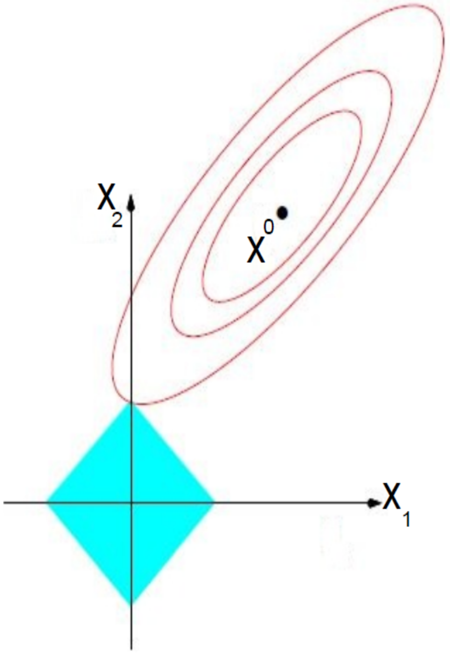


Figure 3.1: This figure illustrates the geometry of the LASSO problem. The blue diamond is the area that is defined by the constraint and the contours are centred around the full least-squares estimate (this figure is taken from [2])

The total least-squares (TLS) framework presents a solution for a non-sparse overdetermined set of linear equations. In addition, it has many applications in system identification based on the errors-in-variables (EIV) model and many other areas. For the grid matching, we are trying to add the sparsity assumption to the TLS framework. The EIV model models the perturbation in the dictionary matrix as an additive error matrix as follows

$$\mathbf{y} = \hat{\mathbf{A}}\mathbf{x} + \mathbf{n}, \quad \hat{\mathbf{A}} = \mathbf{A} + \mathbf{E} \tag{3.29}$$

where $\mathbf{E}$ is the additive error matrix. The TLS problem can then be stated as

$$\underset{\mathbf{E},\mathbf{n}}{\text{argmin}} \; \| \, [\mathbf{E} \;\; \mathbf{n}] \, \|_F^2 \;\; \text{subject to} \;\; \mathbf{y} = (\mathbf{A} + \mathbf{E})\mathbf{x} + \mathbf{n} \, . \tag{3.30}$$

The above problem can be solved by TLS. However, we put a constraint on the sparsity of the coefficient vector. Similar to LASSO, this constraint appears as an extra l_1-norm term with λ. Here again, λ represents the amount of shrinkage. The new problem can be written as

$$\underset{\mathbf{E},\mathbf{n}}{\text{argmin}} \; \| \, [\mathbf{E} \;\; \mathbf{n}] \, \|_F^2 + \lambda \| \, \mathbf{x} \, \|_1 \;\; \text{subject to} \;\; \mathbf{y} = (\mathbf{A} + \mathbf{E})\mathbf{x} + \mathbf{n} \tag{3.31}$$

which it is known as sparse total least-squares (S-TLS). As the structure of the $\mathbf{E}$ matrix and the variance of the noise is unknown, in this work, the weighted sparse structured total least squares (WSS-TLS) is not considered. Our problem has two unknowns. These parameters are the additive error $\mathbf{E}$ and the coefficient vector $\mathbf{x}$. To solve the problem, we initialize the $\mathbf{E}$ matrix with zero and by solving the LASSO problem, we find the $\mathbf{x}$ vector. Based on the estimated $\mathbf{x}$, we update our initial assumption on $\mathbf{E}$. This algorithm stops when the estimated $\mathbf{x}$ in the i-th iteration exceeds the $(i+1)$-th estimation by a predefined boundary value. The detailed description of the algorithm goes as follows :

1. The algorithm starts with $\mathbf{E}^0 = \mathbf{0}$. At the i-th iteration $\mathbf{E}^i$ is assumed known, the algorithm tries to solve the optimization problem

$$\mathbf{x}^i = \underset{\mathbf{x}}{\text{argmin}} \; \| \mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{E}^i\mathbf{x} \|_2^2 + \lambda \| \mathbf{x} \|_1 \tag{3.32}$$

   which is a total least-squares problem equipped with a term $\lambda \| \mathbf{x} \|_1$ to force a penalty on the sparsity of the coefficient vector. The above problem can be solved with LASSO.

2. Then we have an estimation of our coefficient vector. Now it is possible to update our initial assumption on $\mathbf{E}$ by solving the following problem

$$\mathbf{E}^{i+1} = \underset{\mathbf{E}}{\text{argmin}} \; \| \mathbf{y} - \mathbf{A}\mathbf{x}^i - \mathbf{E}\mathbf{x}^i \|_2^2 + \| \mathbf{E} \|_F^2. \tag{3.33}$$

   However the above problem has a quadratic form and can be solved by taking the derivative regarding the $\mathbf{E}$ matrix to find the optimum solution. After setting the derivative to zero the solution is

$$\mathbf{E}^{i+1} = \left(1 + \| \mathbf{x}^i \|_2^2\right)^{-1} (\mathbf{y} - \mathbf{A}\mathbf{x}^i)\mathbf{x}^{i^T} \tag{3.34}$$

3. This algorithm stops when the estimated $\mathbf{x}$ in the i-th iteration exceeds the $(i+1)$-th estimation by a predefined error value

The S-TLS algorithm can be applied to our fingerprinting-based localization framework in two different ways. These two methods are presented below.

1. The first method is to use the dictionary matrix defined in (3.1) and solve (3.7) with the S-TLS algorithm. After deriving the coefficient vector and the $\mathbf{E}$ matrix, the location of the sensor can be estimated by finding the index of the coefficient vector where the maximum value occurs. This step is similar to (**??**) and it can be written as

$$\Lambda' = \underset{i}{\text{argmax}} \; x_i \ .$$

The $\mathbf{E}$ matrix that is derived from this algorithm has one non-zero column and the other columns of the matrix are zero or close to zero. Therefore, the new signature is

$$\mathbf{y}' = \mathbf{e}_{\Lambda'} + \mathbf{a}_{\Lambda'}. \tag{3.35}$$

By replacing $\mathbf{y}$ in (3.7) with $\mathbf{y}'$, we can apply one of the sparse recovery algorithms. The sensor location can be estimated by applying (3.10) to the resulting coefficient vector and a better estimate of the sensor location can be achieved.

2. For the second method, in the off-line phase, a same dictionary matrix as (3.1) is constructed. However, for the on-line phase, instead of looking through all the grid points. First, we find a rough estimate of the sensor location and then we search more thoroughly through the region where the sensor is found with the previous estimation. Therefore, instead of applying the dictionary matrix in (3.1), we reduce the number of grid points and then the S-TLS algorithm is applied. By selecting some of the columns of the dictionary matrix, we can define the following dictionary matrix

$$\mathbf{A}' = [\mathbf{a}_1, ..., \mathbf{a}_{N'}] \;\; (MJ) \times N' \text{ where } N' < N. \tag{3.36}$$

Now, by applying the S-TLS algorithm to

$$\mathbf{y} = \mathbf{A}'\mathbf{x}' + \mathbf{n}, \tag{3.37}$$

the coefficient vector and the $\mathbf{E}$ matrix are derived. After deriving the new signature vector similar to (3.35), instead of using the $\mathbf{A}$ matrix similar to the previous method in (3.7), we only select the grid points that are in the vicinity of the estimated location. In this method, the computational complexity is reduced by decreasing the number of grid points.

Other Localization Algorithms 4

In this chapter, existing algorithms for indoor fingerprinting-based localization are evaluated for an underwater medium. This section starts with a description of the classical k-nearest neighbours localization algorithm. Then a statistical localization method is explained which employs the maximum likelihood estimator for localization. Finally, a localization algorithm which utilizes neural networks is explained.

4.1 K-nearest neighbours

The k-nearest neighbors (KNN) algorithm is a classical method for fingerprinting-based localization. Some preliminary information regarding the KNN algorithm and how it can be adapted to our current localization framework are provided in this section.

In the statistical and machine learning communities, the k-nearest neighbors algorithm is known as a method for solving a regression problem or is a method for classification. To employ this algorithm for our fingerprinting-based localization framework, the initial step is to define the dictionary matrix. The dictionary matrix that is used in this work is similar to (3.1). For the on-line phase, the same vector as (3.6) is constructed. Therefore, the dictionary matrix is restated below as

$$\begin{aligned}\mathbf{A} &= [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N] \quad MJ \times N \\ \mathbf{a}_i &= \frac{1}{S}\sum_{s=1}^{S} \mathbf{a}_{is} \quad MJ \times 1 \\ \mathbf{a}_{is} &= [\,\mathbf{CIR}_{is1}^T, \mathbf{CIR}_{is2}^T, ..., \mathbf{CIR}_{isJ}^T\,]^T \quad MJ \times 1\end{aligned} \tag{4.1}$$

and in the on-line phase, the received new signature is

$$\mathbf{y} = [\,\mathbf{CIR}'_1, \mathbf{CIR}'_2, ..., \mathbf{CIR}'_J\,]^T \quad MJ \times 1.$$

The description of the KNN algorithm is provided below.

1. The new signature is compared with the signatures of the dictionary matrix and k similar signatures are selected. Different metrics can be used to find a similarity between the signatures. Some of the known metrics that are used in the KNN algorithm are

$$\begin{aligned}\text{Euclidean Distance} &= \|\,\mathbf{y} - \mathbf{a}_i\,\|_2, \\ \text{Cityblock Distance} &= \|\,\mathbf{y} - \mathbf{a}_i\,\|_1, \\ \text{Chebyshev Distance} &= \|\,\mathbf{y} - \mathbf{a}_i\,\|_\infty.\end{aligned}$$

In this work, the Euclidean distance is used as a similarity metric. The distance of the new signature from each column of the the dictionary matrix is put in a vector $\mathbf{d}$ and it can be written as

$$\mathbf{d} = [\, d_1, d_2, ...d_N \,]^T \quad N \times 1 \tag{4.2}$$

where

$$d_i = \| \, \mathbf{y} - \mathbf{a_i} \, \|_2. \tag{4.3}$$

The indexes of the k minimum distance values in the $\mathbf{d}$ vector are selected and these values are put in an index set

$$\Lambda = \{\Lambda_1, \Lambda_2, ..., \Lambda_k\}$$

where Λ_i shows the index of the i-th minimum in the $\mathbf{d}$ vector.

2. The next step is to find the location of the sensor. The location can be estimated by averaging over the relative locations of the k nearest neighbours. If we consider the location matrix $\mathbf{L}$ in (3.5), the estimated location of the sensor can be defined as

$$\hat{\mathbf{l}} = \frac{1}{k} \sum_{i=1}^{k} \mathbf{l}_{\Lambda_i} \tag{4.4}$$

where $\mathbf{l}_{\Lambda_i}$ denotes the column of the location matrix which is selected by the i-th member of the Λ set. There is another version of this algorithm which is known as the weighted k-nearest neighbours (WKNN) algorithm. In this approach, a discrimination is made between the grid point that has a closer distance with our new signature and the other $k-1$ grid points. This can be formulated as a weighted average where the influence of the grid point with a closer distance to our sensor location is more significant to the final estimated location. The WKNN can be written as

$$\hat{\mathbf{l}} = \frac{\sum\limits_{i=1}^{k} \left(\frac{\mathbf{l}_{\lambda_i}}{d_{\lambda_i} + \epsilon} \right)}{\sum\limits_{i=1}^{k} \left(\frac{1}{d_{\lambda_i} + \epsilon} \right)} \tag{4.5}$$

where ϵ is put to avoid the division by zero.

The adaptation of this algorithm to the fingerprinting-based localization is done by Bahl and Padmanabhan for the first time in [45]. In this work, location dependent signatures are the RSS values that are received by a number of access points and the location of the mobile node is calculated with the algorithm explained above. Moreover, Brunto and Battiti have employed WKNN in [46] for localization and similar to the previous work, RSS values are gathered as the location-dependent signatures. The result of this work shows an accuracy of 5.79 cm with 95% precision.

4.2 Maximum Likelihood

Another method that is proposed for fingerprinting-based localization is a statistical approach which estimates the sensor location by employing the maximum likelihood (ML) estimator. In this section, this method is explained and some existing works in the literature are presented.

In the ML method, localization is performed by using a geo-regioning technique. Generally, this technique divides the surveillance area into a number of regions instead of grid points and the center of each region depicts the location of the region. Then, a large number of samples is gathered in each region. In [30], It is assumed that the movement of the sensor in the region causes small scale effects. By this assumption, the statistics of the CIRs in each region are evaluated based on the known amplitude and phase distribution for an indoor channel.

In the off-line phase, the dictionary matrix is similar to (3.1). However, it is assumed that there is no time synchronization between the transmitter and the receiver in both phases. Therefore, the alignment strategy that is defined in 3.1.2 is applied to align the CIRs to a reference sample. In the on-line phase, the received signature is a vector similar to (3.6) and the CIRs in this vector are aligned similar to (3.18).

For localization, the new signature vector that is derived in the on-line phase should be related to one of the signature vectors in our dictionary matrix. This can be done by defining the amplitude distribution of the S signature vectors that we extracted from each region. For an indoor environment, many amplitude distributions are proposed. However, there is no existing work that shows the amplitude distribution of the CIRs for an underwater medium. Therefore, for simplicity, it is assumed that the distribution of the samples in each region is Gaussian. This assumption is further assessed for indoor channel in [31] by employing Akaike weights (AIC). After defining the amplitude distribution, three cases can be defined.

1. The first case occurs when the average of S signature vectors in each region is not zero. Then the average value is equal to (3.2) and it is restated below

$$\mathbf{a}_i = \frac{1}{S}\sum_{s=1}^{S}\mathbf{a}_{is} \quad MJ \times 1.$$

The covariance matrix can be written as

$$\mathbf{C}_i = \frac{1}{S-1}\sum_{s=1}^{S}(\mathbf{a}_{is} - \mathbf{a}_i)(\mathbf{a}_{is} - \mathbf{a}_i)^H \tag{4.6}$$

where S is the number of samples in each region, i is the index of the region, $\mathbf{a}_{is}$ is the s-th sample in the i-th region and $\mathbf{a}_i$ is the average vector of the i-th region.

To localize the sensor we can use the maximum likelihood (ML) estimator. The ML estimator chooses the region with the highest probability as estimated region. The probability density function for region i can be written as

$$p(\mathbf{y}|i) = \frac{1}{(2\pi)^{\frac{MJ}{2}} \det^{\frac{1}{2}}(\mathbf{C}_i)} \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{a}_i)^H {\mathbf{C}_i}^{-1}(\mathbf{y}-\mathbf{a}_i)\right) \tag{4.7}$$

where i is the index of the region and $p(\mathbf{y}|i)$ shows the probability that the received signature in the on-line phase is originating from region i. Therefore, by considering the log-likelihood, we can define the maximum likelihood estimator as

$$\begin{aligned} i_{ML} &= \underset{i}{\operatorname{argmax}} \;\; \ln(p(\mathbf{y}|i)) \\ &= \underset{i}{\operatorname{argmax}} \;\; \left(\frac{-MJ}{2}\ln(2\pi) - \frac{1}{2}\ \ln(\det(\mathbf{C}_i)) - \frac{1}{2}(\mathbf{y}-\mathbf{a}_i)^H {\mathbf{C}_i}^{-1}(\mathbf{y}-\mathbf{a}_i)\right) \end{aligned} \tag{4.8}$$

where i_{ML} denotes the index of the estimated region, $\det(\mathbf{C})$ is the determinant of the covariance matrix of region i.

2. If the measured CIRs from a region are so different that the constructive and destructive interference of the multipath reflections cancel each other on average, the average vector in (3.2) can be considered zero and therefore the covariance matrix is equal to

$$\mathbf{C}_i = \frac{1}{S-1}\sum_{s=1}^{S} \mathbf{a}_{is}\mathbf{a}_{is}^H. \tag{4.9}$$

   Similar to the previous approach the ML estimator can be used to find the estimated location of the sensor. This can be written as

$$\begin{aligned} i_{ML} &= \underset{i}{\operatorname{argmax}} \;\; \ln(p(\mathbf{y}|i)) \\ &= \underset{i}{\operatorname{argmax}} \;\; \left(\frac{-MJ}{2}\ln(2\pi) - \frac{1}{2}\ \ln(\det(\mathbf{C}_i)) - \frac{1}{2}\mathbf{y}^H {\mathbf{C}_i}^{-1}\mathbf{y}\right) \end{aligned} \tag{4.10}$$

3. The last case occurs when we assume that the different signature vectors from a region are statistically independent and the mean also is considered zero. This assumption results in a diagonal covariance matrix where the values on the diagonal are the average PDP values. This assumption results in a major reduction in the computational complexity. The ML estimator can now be defined as

$$\begin{aligned} i_{ML} &= \underset{i}{\operatorname{argmax}} \;\; \ln(p(\mathbf{y}|i)) \\ &= \underset{i}{\operatorname{argmax}} \;\; \left(\frac{-MJ}{2} \ln(2\pi) - \frac{1}{2} \sum_{n=1}^{MJ} \ln([\mathbf{C}_i]_{(n,n)}) - \frac{1}{2} \sum_{n=1}^{MJ} |[\mathbf{y}]_n|^2 \left(\frac{1}{[\mathbf{C}_i]_{(n,n)}} \right) \right) \end{aligned} \tag{4.11}$$

where $[\mathbf{C}_i](n,n)$ is the n-th element on the diagonal of the covariance matrix of the region i, $[\mathbf{y}]_n$ is the n-th element of the $\mathbf{y}$ vector. This algorithm is also employed in [47–49] for localization. Due to the computational simplicity and the fact that the amplitude distribution of the CIRs in each region is unknown, this case is applied to our localization framework.

4.3 Neural Networks

In this section, localization is performed by utilizing the artificial neural networks (NN) framework. In the beginning of this section, some preliminary information about the multi-layer perceptron, its components and error back-propagation are provided. Afterwards, some previous localization algorithms which use this framework for localization and their results are presented. It should be noted that a detailed description of different types of neural networks and machine learning algorithms is not provided as the localization is the main focus of this work. Thus, for more details see [50–52].

4.3.1 Preliminaries

The interest in artificial neural networks emerged after the introduction of simplified neurons by McCulloch and Pitts in 1943. Artificial neural networks consist of a pool of processing units that are communicate with each other over weighted connections. This idea can be considered as a variation of the parallel distributed processing idea. The processing units are known as neurons. Different topologies are proposed for connecting the neurons. One of these topologies is a feed-forward network, which is a network where the data flow from input to output neurons is strictly feed-forward and no feedback connections are present. The data processing can extend over multiple layers of neurons.

The multi-layer perceptron is a type of feed-forward network. Each layer consists of neurons which receive their inputs from neurons in a layer directly below them and send their outputs to neurons in the next layer. There are no connections within a layer. The topology of this network with one hidden layer is shown in Fig. 4.1. Generally, in the input units, no processing takes place. Thus a network with one input layer, one hidden layer and one output layer is referred to as a network with two layers. Below, we list the components of the first output neuron of the MLP network in Fig. 4.1. This can be simply extended to all the neurons in the network.

1. Three types of neurons can be defined: input neurons (indicated by an index *in*) that receive data from outside the neural network, output neurons (indicated

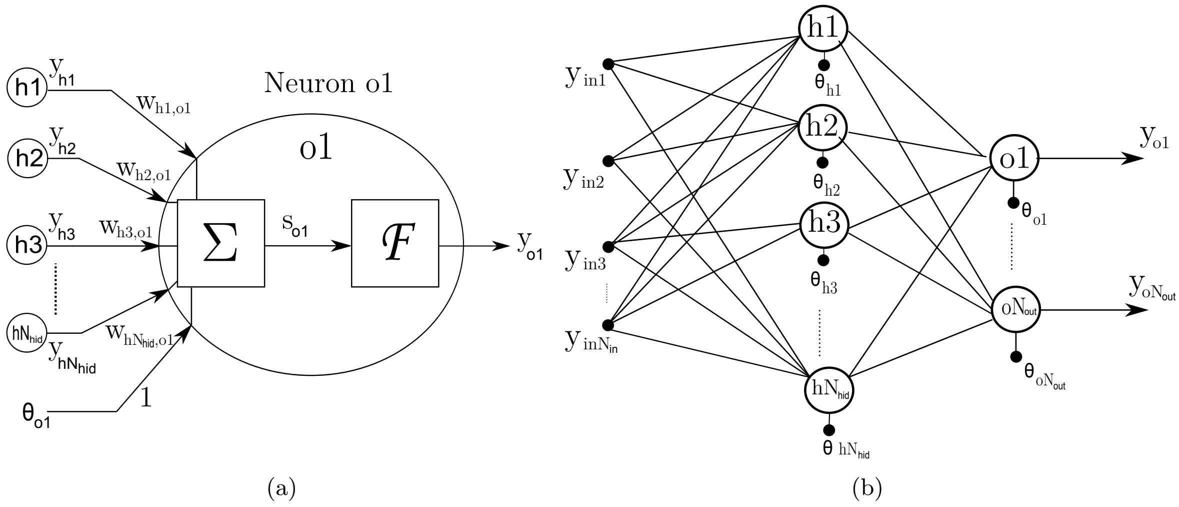


Figure 4.1: Multi-layer perceptron (a) one neuron (processing unit) (b) multi-layer perceptron topology with two layers

by an index o) that send data out of the neural network and hidden neurons (indicated by an index h) whose input and output signal remain within the neural network.

2. A state of activation y_{o1} for neuron $o1$: the output of the neuron.

3. Connection between neurons: each connection is defined by a weight $w_{hj,o1}$ which determines the effect that the signal of j-th hidden neuron has on neuron $o1$.

4. An external bias or offset θ_{o1} for neuron $o1$: this value is a constant that is independent of the network inputs and this constant is adapted by the learning rule.

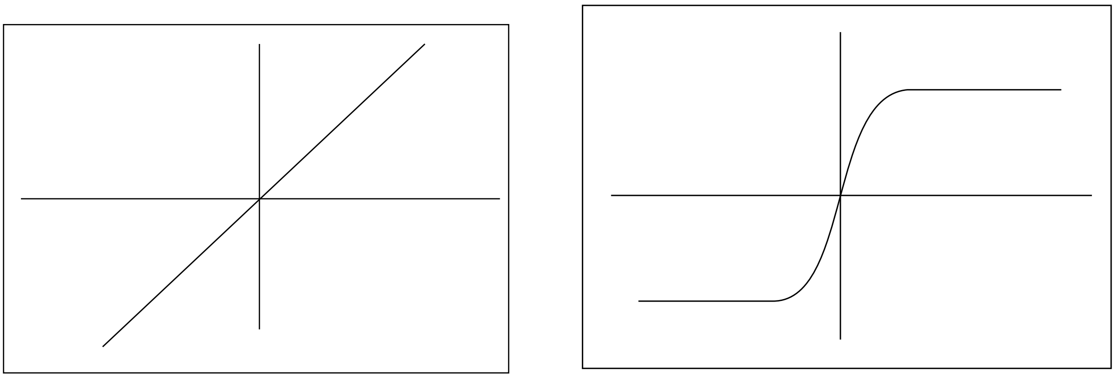
5. A propagation rule: this determines the effective input s_{o1} of a neuron $o1$ from its external inputs. In our case we assume that each neuron provides an additive contribution to the input of the neuron with which it is connected. The effective input can be defined as

$$s_{o1} = \sum_{j=1}^{N_{hid}} w_{hj,o1}\, y_{hj} + \theta_{o1}. \tag{4.12}$$

   Neurons with this propagation rule are known as sigma neurons.

6. An activation function f_{o1} is a function which determines the new level of activation based on the effective input s_{o1}. Generally this function is defined as a non-decreasing function of the total input of the neuron

$$y_{o1} = f_{o1}(s_{o1}) = f_{o1}\left(\sum_{j=1}^{N_{hid}} w_{hj,o1} y_{h1} + \theta_{o1}\right) \tag{4.13}$$



(a) (b)

Figure 4.2: Activation function (a) pure linear function (b) tan-sigmoid function

Generally, the activation function is some sort of a threshold function: a hard limiting threshold function (a sign function), a linear or semi-linear function or a smoothly limiting threshold. Two activation functions that are used in this work are the tan-sigmoid function and the linear function and they are shown in Fig. 4.2. The tan-sigmoid function is a smoothly limiting function which can be defined as

$$y_{o1} = f_{o1}\left(s_{o1}\right) = \frac{\exp(s_{o1}) - \exp(-s_{o1})}{\exp(s_{o1}) + \exp(-s_{o1})} \ . \tag{4.14}$$

After defining the components of a neuron, a learning rule to update the weights between the neurons should be defined. The weights in a neural network have to be configured in a way to produce the desired set of outputs by entering a set of inputs. Updating the weights can be done in two different ways: supervised learning where an external teacher provides a set of desired input and output pairs and the weights change according to these pairs or unsupervised learning where the network finds the pattern of its inputs by itself and it updates the weights accordingly. In both methods, the rule for changing the weights follows the Hebbian learning rule. The basic idea is that if two input neurons $o1$ and hj are active simultaneously, their interconnection must be strengthened. If $o1$ receives an input from hj, the simplest version of the Hebbian learning rule is to modify the weight $w_{hj,o1}$ with

$$\Delta w_{hj,o1} = \gamma y_{hj} y_{o1} \tag{4.15}$$

where γ is the learning rate. For supervised learning that is used in this work, the neural network starts with a set of random weights between the neurons. A delta rule which is derived from the Hebbian learning rule tries to minimize the error between the training samples and the actual output of the network iteratively. This can be done by defining total mean squared error cost function as

$$\text{error} = \frac{1}{t}\sum_{i=1}^{t} e_{o1}^{(i)} = \frac{1}{t}\sum_{i=1}^{t}(d_{o1}^{(i)} - y_{o1}^{(i)})^2 \tag{4.16}$$

where $d^{(i)}$ is the i-th training output sample, $y^{(i)}$ is the actual network output in the i-th iteration, t is the total number of training pairs that is used to train the network. The idea is to reduce the cost function iteratively. This can be done by employing the gradient decent algorithm. Since total error is the mean of all samples one cannot technically calculate the gradient of the error with regard to weights until the entire set of training samples is available. However, this is very computationally intensive so this is usually performed individually with each training pattern as an approximation. Therefore, the idea is to minimize the output error of each training sample by moving towards the negative of the derivative of the error with respect to the weight between the two neurons.

$$w_{hj,o1}^{(i+1)} = w_{hj,o1}^{(i)} + \Delta w_{hj,o1} = w_{hj,o1}^{(i)} + \gamma\frac{\partial e^{(i)}}{\partial w_{hj,o1}}|_{w_{hj,o1}^{(i)}} \tag{4.17}$$

where γ is the learning rate and $w_{hj,o1}$ is the weight of the connection between the j-th hidden neuron and the output neuron $o1$. This learning rule can be applied to a single neuron for updating the weights. A generalization of the delta rule for nonlinear activation functions and multilayer networks is known as error back propagation. The central idea behind this solution is that the errors for the neurons of the hidden layer are determined by back propagation of the error of the output layer. Thus the algorithm can be explained as follows. The training pairs are used as an input and the output values are propagated to the output neurons. Then the actual network output is compared with the desired output values and we usually end up with an error in each of the output neurons. Let's call this error $e_{oj}^{(i)}$ for the j-th output neuron for the i-th training sample. We have to bring the total output error to zero. The simplest method is a greedy method: we try to change the connections in the neural network in such a way that in the next iteration the error e_{oj} gets closer to zero. In order to reduce an error we have to adapt its incoming weights according to (4.17). This is the first step. However, this is not enough due to the fact that by applying the delta rule, the weights from the input to the hidden neurons are never changed. In order to adapt the weights from the input to the hidden neurons, we again have to apply the delta rule. We distribute the error of an output neuron oj to all the hidden neurons that it is connected to. In other words, a hidden neuron hj receives an error value from each output neuron o equal to the error of that output neuron weighted with the weight of the connection between those neurons. In this way, the weights between the input and the hidden layer can also be changed with the delta rule. Propagating the errors back in order to update the weights of the MLP network is known as the back-propagation algorithm.

Basically, feed-forward networks have two training phases. In the first phase, the training set is fed to the network. After training and adapting the weights, the same data that was used for training is fed again to the network to compute the error between the desired outputs and the actual outputs. This step is known as the memorization phase. In the second phase, untrained data sets are used as an input of the network and the error is calculated between the desired outputs and the actual outputs. This step is called the generalization phase. For localization, the performance of the network in the generalization phase is much more important than the memorization phase.

4.3.2 Localization Algorithms

In this section, some previous works that utilized neural networks for localization are reviewed. In addition, our approach for underwater localization by using the neural network framework is explained.

The idea of using neural networks for fingerprinting-based localization is presented for the first time by Nerguizian et al. in [53]. This work also uses the geo-regioning technique for localization. In [53], a different training matrix is built for the off-line phase and only one receiving antenna is considered for localization. Each column of this training matrix is composed of seven parameters that are extracted from the CIRs in each region. These seven parameters are mean excess delay ($\bar{\tau}$), RMS delay spread (τ_{rms}), maximum delay spread τ_{max}, power of the first path (P_1), delay of the first path (τ_1), total power of the received signal (P) and the number of multipath components (MP). These parameters are explained in section 2.1.1.4 and the training vector which is used as an input of the MLP network can be written as

$$\mathbf{t}_{is} = [\tau_{max,is}, \tau_{m,is}, \tau_{rms,is}, \tau_{1,is}, P_{1,is}, MP_{is}, P_{is}]^T \quad 7 \times 1. \tag{4.18}$$

where $\mathbf{t}_{is}$ refers to the s-th sample of the i-th region. For the on-line phase, the new signature vector is defined as

$$\mathbf{y}' = [\tau'_{max}, \tau'_m, \tau'_{rms}, \tau'_1, P'_1, MP', P']^T \quad 7 \times 1. \tag{4.19}$$

A neural network is applied to perform the function approximation between the input data and the location coordinates. The parameters extracted for each grid point are used as an input for the NN. In this case, a simple feed-forward neural network is used to perform the non-linear mapping between these parameters and the location coordinates. An MLP is used for this case due to the smaller location error in the generalization phase of the network. In addition, to get a good generalization property, the Bayesian regularization with the Levenberg-Marquardt algorithm is used as a learning algorithm and a hidden layer with 10 neurons is chosen. A wideband signal with a central frequency of 2.4 GHz with 200 MHz bandwidth is used for underground mine localization.

In [54], the parameters are reduced to three parameters and an UWB signal is used. The parameters are excess delay, total received power and rms delay spread. An MLP network with one hidden layer is used with 7 neurons in its hidden layer. The learning algorithm is also similar to the previous method. The simulation is performed in 1D. For the 3 GHz to 6 GHz and the 3 GHz to 10 GHz bandwidths signal errors of 91 cm and 3 cm for the trained data set and 160 cm and 81 cm for the untrained data set are respectively reported. This work is extended to 2D localization in [55]. In this work, the first two parameters are similar to the previous work and a new location flag is used which is related to the application. An MLP network with 2 hidden layers of 7 and 12 neurons is used and the training algorithm is a scaled conjugate gradient (SCG) method. For the training data and the test data, accuracies of 0.2m and 0.5m with 74% and 67% precision in the x direction and 0.195m and 0.5 with 80% and 60% precision respectively in the y direction are reported. In this work, the localization performance of a general regression neural network (GRNN) which is a radial basis function type of network is also evaluated. This GRNN network has 1 hidden layer with a radial activation function. For the training and the test dataset, an accuracy of 0.6m with respectively 80% and 60% precision in the x direction are observed and for the y direction accuracies of 0.4m and 0.65m with 73% and 70% precision are respectively reported.

In [56], CIR and PDP values are directly fed to the neural network. Path loss information is neglected by normalizing the energy to one and one receiving antenna is considered. Two types of signature vector are used as the input of the MLP network. One of them is derived by utilizing the CIR directly as an input of the network. The other signature vector is derived by employing PDP values instead of the CIR itself.

For the training set, N realizations of the CIR are put in a matrix. The matrix is gone through a linear pre-processing stage before it is used as an input of the MLP. This pre-processing changes each CIR vector to a zero mean and unit variance vector. Then these vectors go through a linear transformation by employing a feature selection algorithm based on a principal component analysis (PCA).

After this preprocessing, this matrix is used as a training set for an MLP feed-forward network with one hidden layer. Besides, the scaled conjugate gradient algorithm is used as a learning algorithm. For the vector with 90 elements, 115 hidden neurons and for the vector with 45 elements, 65 hidden neurons are used. An accuracy of 1m with 93% precision is reported for this method.

In [57], information is extracted from the CIR via a wavelet feature extraction. The CIR is compressed by using the discrete wavelet transform (DWT). A radial basis network is used for function approximation. An accuracy of 2 meters with 90% and 80% precision for respectively the trained an untrained data set is achieved.

In [58], two receiving antennas are used for localization. The dictionary matrix that is defined in this work is used also for our localization framework. The dictionary matrix is an extension of the dictionary matrix in (4.18) to multiple receiving antennas and it can be defined as

$$\mathbf{t}_{is} = [\mathbf{t}_{is1}^T, \mathbf{t}_{is2}^T, ..., \mathbf{t}_{isJ}^T]^T \;\; 7J \times 1 \tag{4.20}$$

$$\mathbf{t}_{isj} = [\tau_{max,isj}, \tau_{m,isj}, \tau_{rms,isj}, \tau_{1,isj}, P_{1,isj}, MP_{isj}, P_{isj}]^T \;\; 7 \times 1 \tag{4.21}$$

where J shows the total number of antennas. For the on-line phase, the new signature matrix is defined as

$$\mathbf{y}'' = [\mathbf{y}'^T_1, \mathbf{y}'^T_2, ..., \mathbf{y}'^T_J]^T \quad 7J \times 1 \tag{4.22}$$

with

$$\mathbf{y}'_i = [\tau'_{maxi}, \tau'_{mi}, \tau'_{rmsi}, \tau'_{1i}, P'_{1i}, N'_i, P'_i]^T \quad 7 \times 1. \tag{4.23}$$

A wideband signal with a central frequency of 2.4 GHz and a 200 MHz bandwidth is used for detecting an underground mine in 1D. An accuracy of 60cm with 90% precision for the trained data and 1 meter with 90% precision for the untrained data are reported. Our approach for underwater localization is similar to this work but with more antennas.

In [59], four parameters are extracted from the CIR. These parameters are total energy integrated over the CIR (E_{tot}), the energy integrated over the strongest received path (E_{max}), rms delay spread τ_{rms} and the delay associated with the percentile of the received energy ($\tau_{percent}$). Four receiving antennas are used. Therefore, for the parameter E_{tot} the input of the neural network is a vector with four elements extracted from the receiving antennas. The test data results show an accuracy of 2 m with 90% precision. In addition, the time of arrival (TOA) information of four antennas is added to the vector, a better performance of 1 m with 95% precision is achieved.

Our approach is similar to [58] with 8 antennas. A MLP network with one hidden layer and 65 neurons is used. In addition, the back propagation learning algorithm is used to adjust the weights. The number of training vectors that are used in this work to train the MLP network is NS where N shows the total number of regions and S shows the total number of samples in each region. The output of the network is the 3D location coordinates. Therefore, the network has 56 input neurons and 3 output neurons.

Simulation Results 5

In this chapter, numerical results of the localization algorithms described in the previous chapters are provided. This section starts by defining the grid points for an off-line phase. Afterwards, the localization results of different algorithms are provided.

5.1 Grid Points

For the off-line phase, the water tank is divided into 128 regions where the size of each region is 8.06 cm × 8.06 cm × 6.75 cm. In each region, four CIR samples are extracted around the center of the region. Therefore, in (3.1), N is 128 and S is 4. These points have 2 cm distance in x and y direction and 6.75 cm in z direction. The center of the region from the region next to it has 8.06 cm distance in the x and y direction and 6.75 cm distance in the z direction. The defined grid points are depicted in Fig. 5.1. The green points show the center of the regions and the blue points are the four samples that are measured in each region. For each blue point, CIRs are averaged over 5 samples. For localization, 8 receiving antennas are used. These antenna are put in the corners of the plastic frame that is put inside the tank. In addition, for each CIR 30K channel taps are gathered and by creating a signature vector similar to (3.3), a signature vector with 240 K ($8 \times 30K$) samples can be derived. Therefore, in (3.1), M is also 240 K and J is equal to 8. The final vector that is used as a location dependent signature in our database is shown in Fig. 5.2. The location matrix $\mathbf{L}$ is also constructed as it is defined in (3.5).

For the on-line phase, the transmitting antenna is put in 57 random places in the tank and the CIRs are extracted with the same vector size as the database signature vectors. These random signatures are not averaged over 5 samples and only 1 sample is obtained for each point. These random points are shown in Fig. 5.1 with red points.

5.2 Localization Algorithms

5.2.1 k-nearest neighbors

To assess the performance of the classical KNN and WKNN algorithms described in Section 4.1, the Euclidean distance of the signature vector that is received in the on-line phase from the dictionary matrix signature vectors is calculated and the 3 nearest neighbors are selected. The mean squared error between the estimated location of the sensor and the actual location is calculated and the cumulative distribution function (CDF) of this error is plotted in Fig. 5.3. The accuracy of the KNN and WKNN algorithms is 5.77 cm and 6.72 cm respectively with 96% precision. WKNN performs

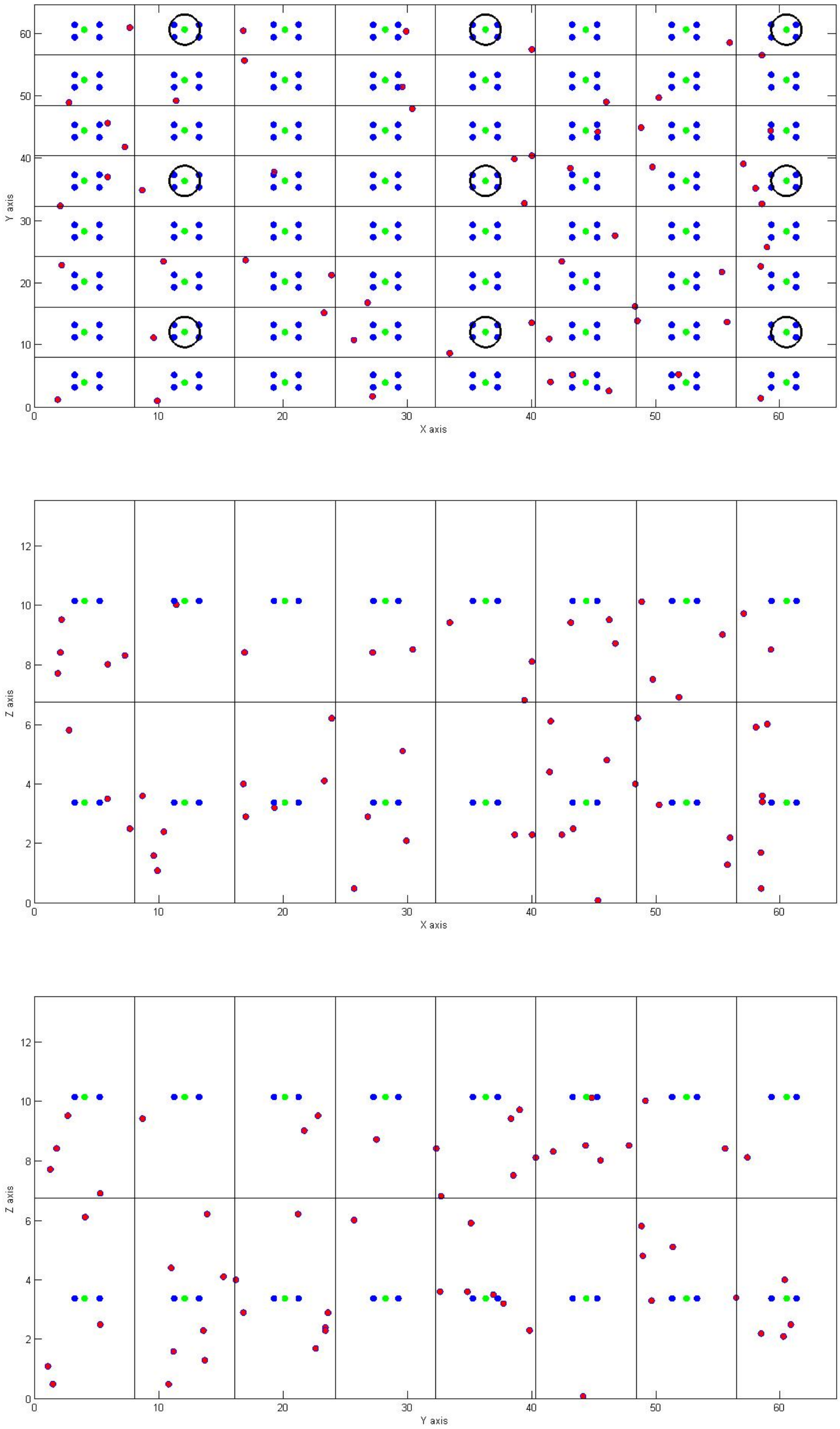


Figure 5.1: XY plane, XZ plane and YZ plane. Green points represent the centre of the regions, blue points are the samples in each region, red points are the random locations that are used for the on-line phase and the black circles show the selected grid points of the dictionary matrix that are used in the S-TLS algorithm

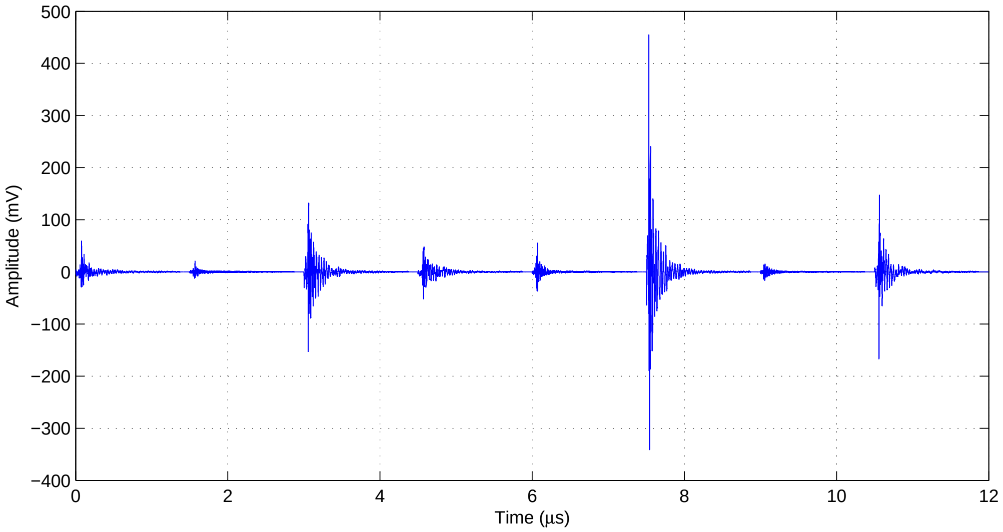


Figure 5.2: An example of a signature vector that is used as a column of the dictionary matrix

better as it takes the distance between the neighbors into consideration and calculates the weighted average to estimate a location.

5.2.2 Neural Networks

To train the neural networks, a dictionary matrix is built by extracting 7 parameters from the CIRs for each region. These parameters are extracted by employing the threshold level that is defined in Section 2.3.2. These parameters are stacked together as defined in (4.21). For this simulation, we use the Matlab neural networks toolbox (nntool). A MLP network with one hidden layer is used and the hidden layer has 65 neurons. Neurons in the hidden layer use a non-linear tan-sigmoid activation function and neurons in the output layer use a linear function. The learning algorithm is the Bayesian regularization with the Levenberg-Marquardt (learnbr.m) to have a good generalization property. The network has three outputs which is relative to the 3D location coordinates of the training points. The block diagram of this network is shown in Fig. 5.4. The training is repeated as many times as possible to reach the minimum mean square error (MSE) between the desired and the actual output vectors.

For the on-line phase, 7 parameters are extracted from the CIRs similar to the training phase and a vector similar to (4.22) is built. Therefore, the **y** vector is used as an input of the neural network and the outputs for the training data and the test data are shown in Fig 5.5. These results show an accuracies of 60 cm and 90 cm with precisions of 96% and 90% for the training and testing dataset respectively. Due to the short distances between the grid points and the small variation in the extracted vectors, the signature vectors have a significant correlation and the results are not satisfactory.

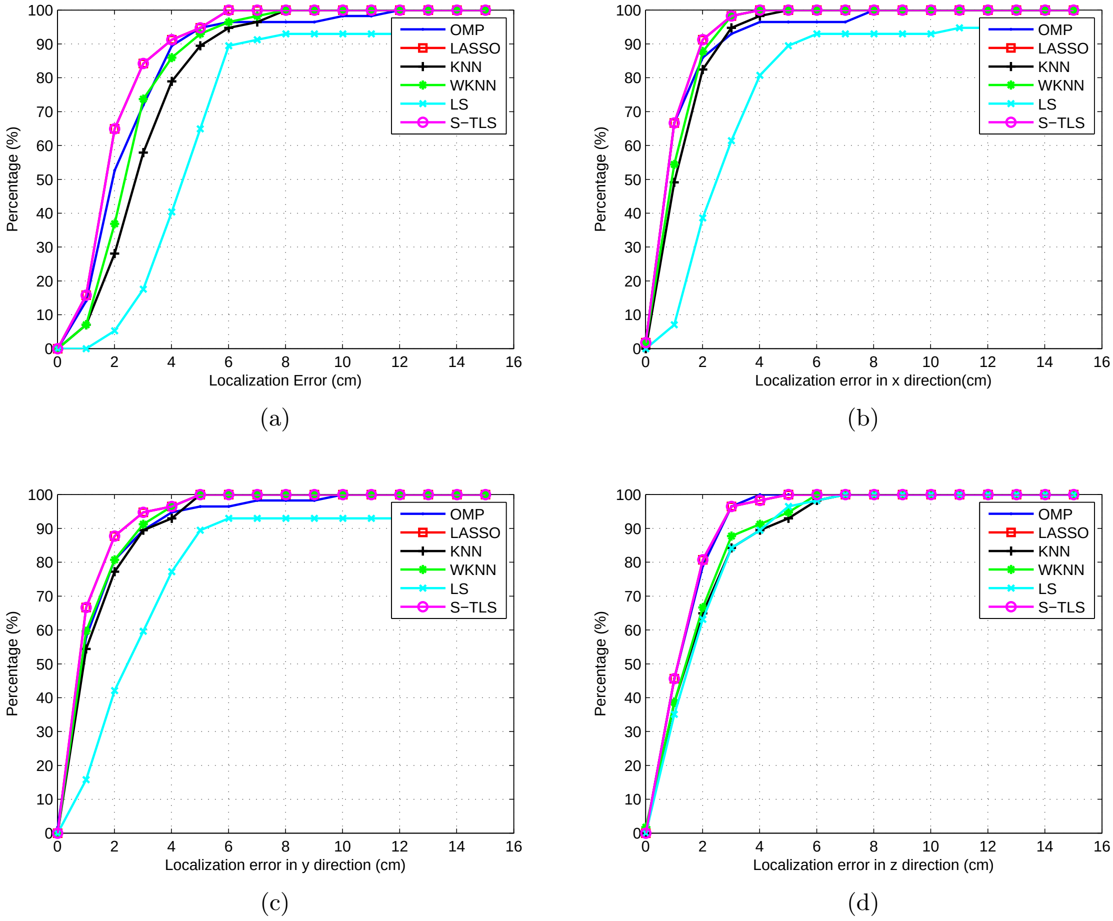


Figure 5.3: Numerical results for the OMP, LASSO, KNN, WKNN, S-TLS algorithms

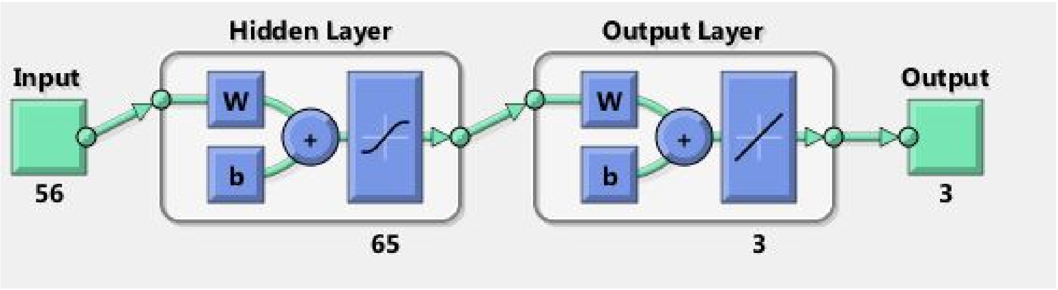


Figure 5.4: Block diagram of the neural network that is used for the fingerprinting-based localization

5.2.3 Sparse Localization

To evaluate the performance of the sparse localization algorithms, the dictionary matrix that is build in the off-line phase is similar to (3.1). For the 57 random locations, the location is estimated by the OMP, LASSO and LS algorithms. Besides, it should be

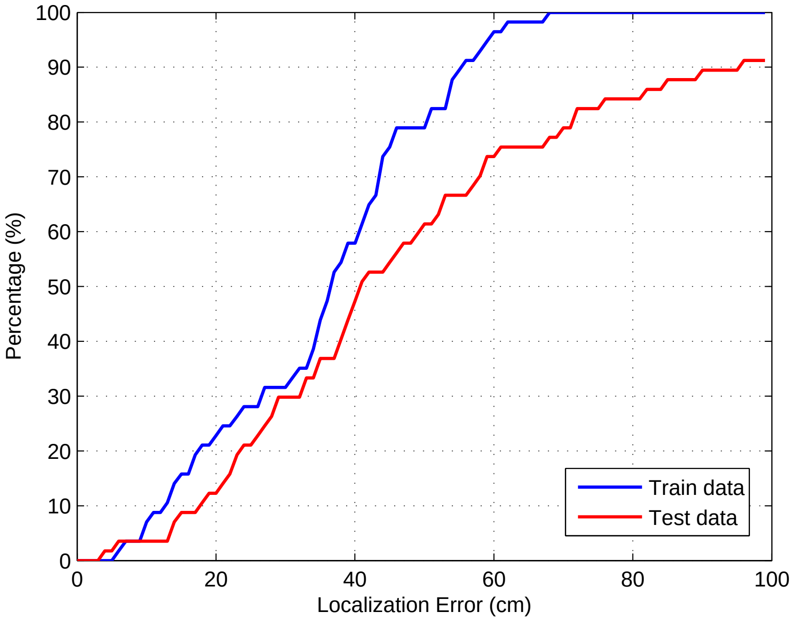
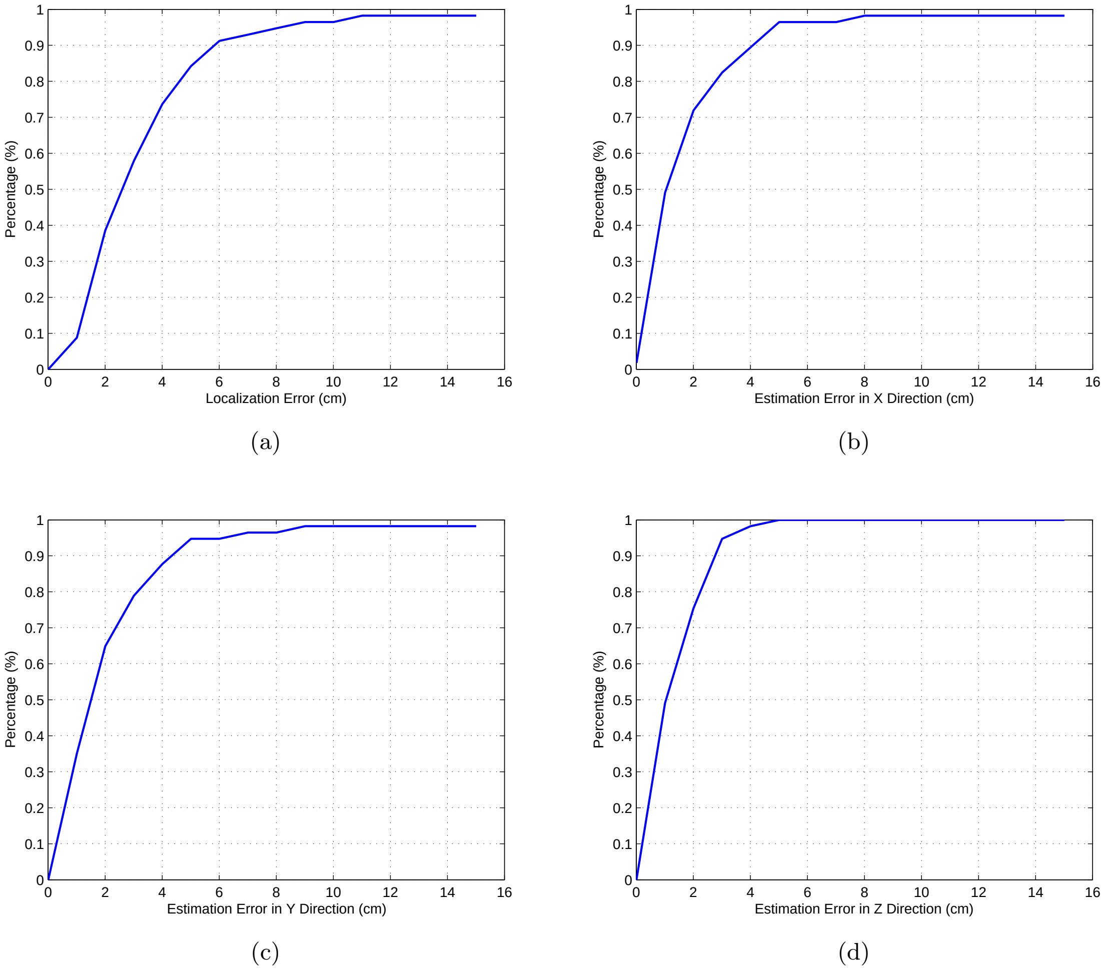


Figure 5.5: Localization performance of the neural network method

noted that for estimating the sensor location, it is assumed that the coefficient vector is a 3-sparse vector. Therefore, in (3.10), k is equal to 3. Results show an accuracy of 6 cm, 5.70 cm and 8 cm with 96%, 98%, 91 % precisions respectively for the OMP, the LASSO and LS methods. LASSO function in Matlab is used for simulating the LASSO algorithm. The S-TLS is also applied as it is explained in Section 3.4. In Fig. 5.3, the first S-TLS method is used for localization. An accuracy of 5.70 cm with 98% precision is achieved. Therefore, among all the localization algorithms that are explained in this work, the best localization performance is achieved with the LASSO and S-TLS algorithms. Both methods have almost a similar localization performance, although, S-TLS is slightly better than LASSO. The better performance of the LASSO and the S-TLS algorithms is due to the selection of different weights in the weighted average for these methods in comparison with the WKNN algorithm. In addition, The WKNN method is better than the OMP algorithm.

For the second S-TLS method, in the on-line phase, a new dictionary matrix with a fewer number of signature vectors is employed to find an estimate of the sensor location. The grid points that are selected from the dictionary matrix $\mathbf{A}$ are marked with a black circle around them in Fig. 5.1. These 9 grid points are selected for all the XY planes in the z direction. Therefore, the columns of the dictionary matrix is reduced from 128 to 72. After applying the S-TLS algorithm and achieving the coefficient vector and the $\mathbf{E}$ matrix, the maximum element in the coefficient vector shows the estimated location of the sensor. To find an exact location, the grid points in the vicinity of the estimated location are selected from the dictionary matrix $\mathbf{A}$. The simulation results are shown in Fig. 5.6. These results show an accuracy of 8 cm with 94% precision. The computational complexity is reduced with the price of having a lower accuracy.



(a) (b)

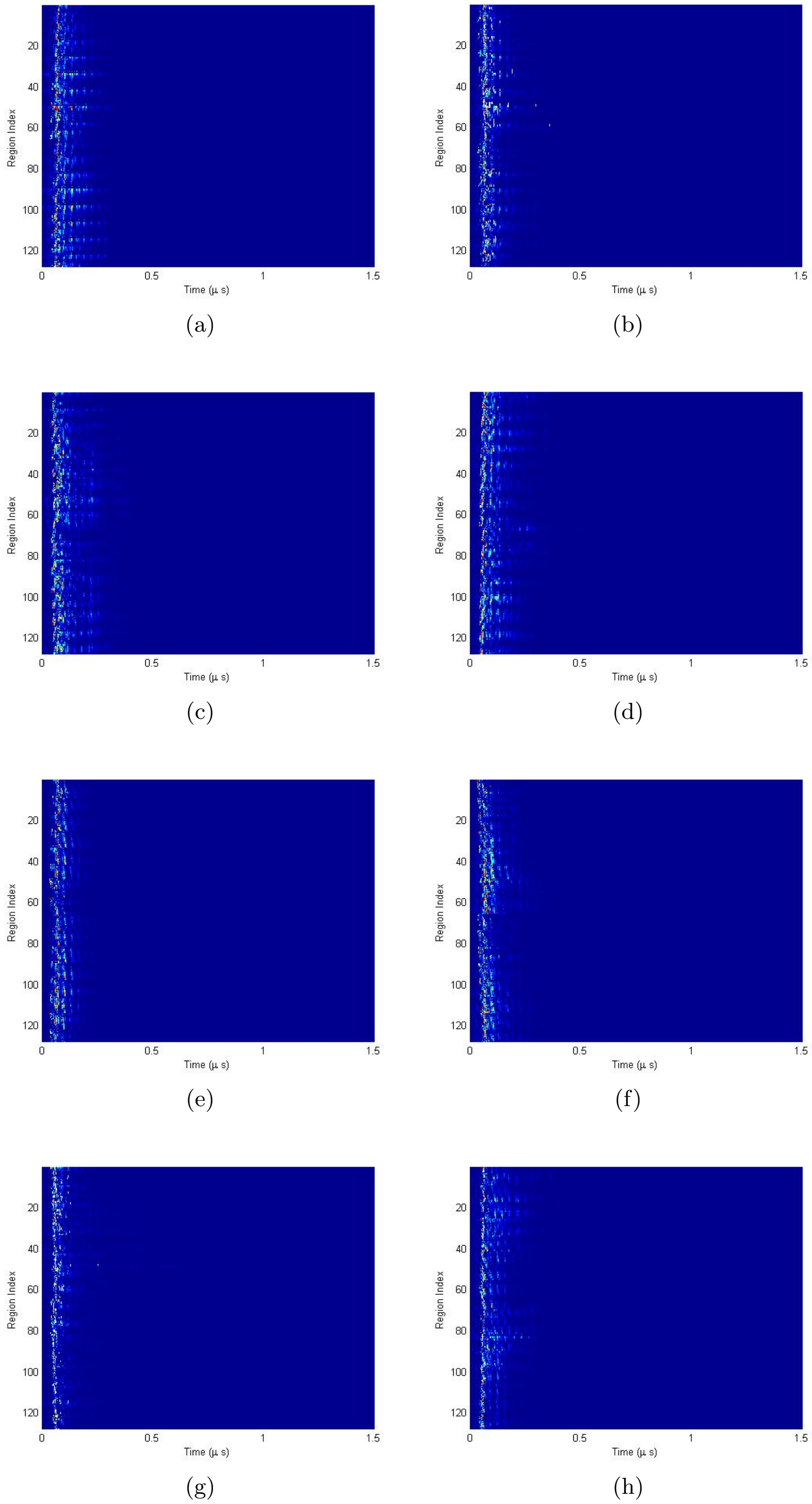
(c) (d)

Figure 5.6: Localization performance of the grid matching algorithm with fewer number of grid points

5.3 Length of the signature vectors

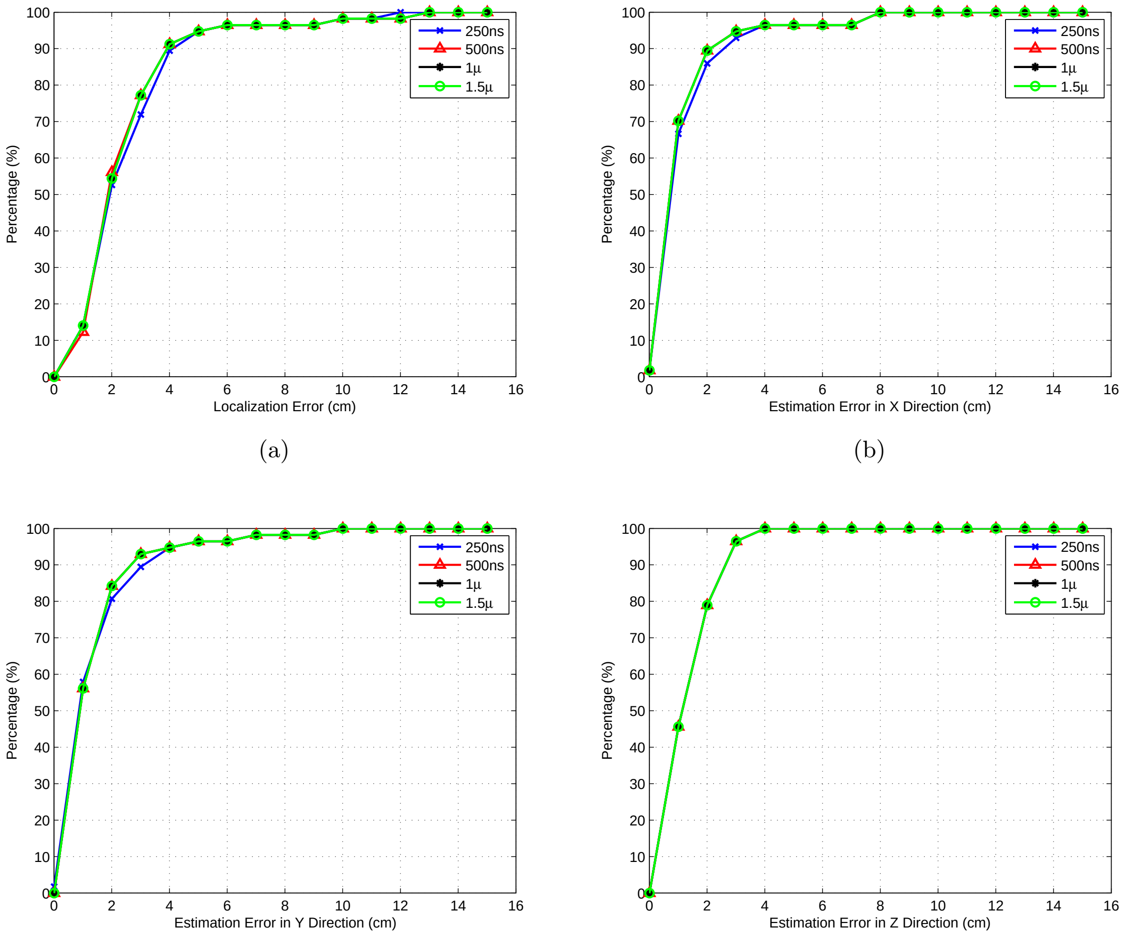
Till now, the results show that the localization is feasible for an underwater medium with a high accuracy. However, one of the requirements of our application is the real-time localization. Therefore, the execution time of our algorithms is an important parameter. To have high resolution CIR vectors, in the off-line phase a long time window and also a high sampling frequency are selected. However, due to this selection, the dictionary matrix is a huge tall matrix. In this section, we investigate the effects of reducing the channel samples and also the sampling frequency on our localization performance. This section starts by evaluating the different time window lengths and it follows by the reduction in sampling frequency.

First, different time window lengths are selected and the localization results are shown by using the OMP algorithm. In Fig. 5.7, CIRs that are extracted for each antenna in all the regions are depicted where the index shows the region index. This



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5.7: Normalized CIRS as received by the 8 antennas from different regions (a-h) antenna 1 to 8



(a) (b) (c) (d)

Figure 5.8: Simulation results for different time window lengths

figure shows that a significant amount of the channel taps are only noise and they are not containing any information. Therefore, by reducing the size of the time window, we should not experience any significant change in our localization performance. Different time windows with lengths 1.5 μs, 1 μs, 500 ns and 250 ns are selected. The OMP algorithm is applied to these signature vectors for location estimation. With a 20 GHz sampling frequency, the 1.5 μs, 1 μ s, 500 ns and 250 ns have 30 K , 20 K, 10 K and 5 K sampling points for each CIR. Accordingly the dictionary matrix has vectors with 240 K, 160 K, 80 K and 40 K elements. In Fig. 5.8, the simulation results show that a shorter time window can be used. The execution time of this algorithm for different time windows is shown in Table 5.1.

Another parameter that can reduce the length of the signature vectors is the sampling frequency. In this approach, the signatures are down-sampled and the results are shown again by applying the OMP algorithm. Downsampling removes the unnecessary samples that result from the oversampling. Different sampling frequencies that are used

Table 5.1: Algorithm execution time for different time window lengths

Time window size	Execution time
1.5 μs	3.16 s
1 μs	2.36 s
500 ns	1.19 s
250 ns	900 ms

in this work are 20 GHz, 10 GHz and 5 GHz. The simulation results are shown in Table 5.2.

Table 5.2: Algorithm execution time for different sampling frequency

Sampling frequency	Execution time
5 GHz	632ms
10 GHz	1.29 s
20 GHz	3.16 s

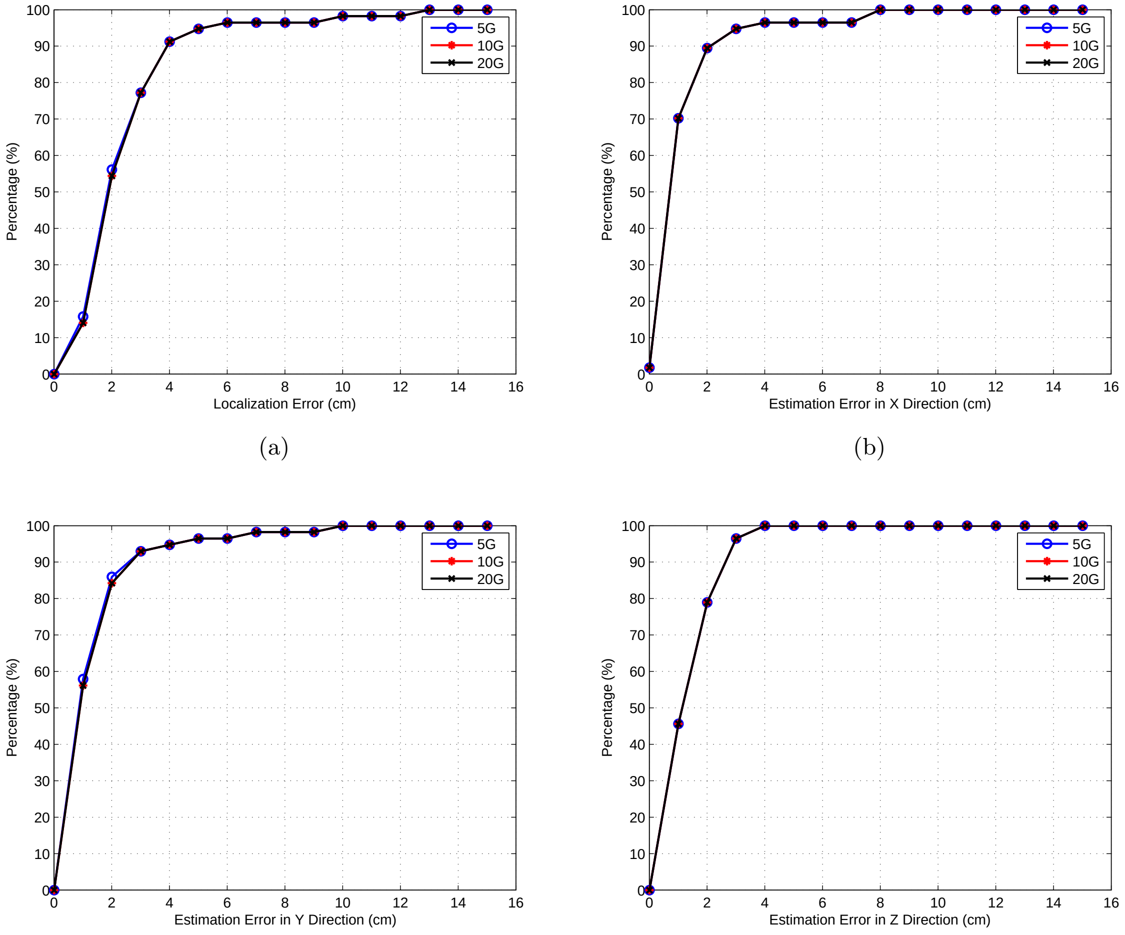
The achieved results shows that the signature can be reduced significantly without any major error in its localization performance. Therefore, the time window cab be reduced to 250 ns and the sampling frequency to 5 GHz. The localization performance of the resulting CIR is compared with the high resolution CIR in Fig. 5.10. The execution time is reduced from 3.16 s to 500 ms . The execution time can further be reduced by optimizing the codes.

5.4 Unsynchronized CIRs

Till now the simulations are done by synchronizing the transmitter with the receiver in both phases. In this section, the localization performance is evaluated for an unsynchronized scenario. For this scenario, the maximum likelihood method and the sparse localization method are assessed.

5.4.1 Maximum likelihood

The CIRs are aligned with the alignment strategy explained in Section 5.6. The third approach is used. In our localization scenario, in contrast to the works that are done in [30,47–49], the number of samples that we obtain in each region is not more than the length of the signature vector ($S \ll MJ$) due to two main reasons. First, the region that we define is so small and the number of samples that we can gather in each region is small. The second reason is related to the number of channel taps that is used in the columns of the dictionary matrix which makes the calculation of the covariance matrix difficult due to the large size of the signature vectors. Therefore, the average PDP values are calculated. Localization is performed with the assumption that the channel taps follow the Gaussian distribution and they are statistically independent. The estimated location can be found by identifying the region with the maximum probability. The



(a)

(b)

(c)

(d)

Figure 5.9: Localization performance with different sampling frequencies

localization results are depicted in Fig. 5.11. The results show an accuracy of 30 cm with a 96% precision. The bad performance of the maximum likelihood algorithm can be related to the two simplifying assumptions that we made. This shows that the channel taps are not exactly following the complex Gaussian distribution and also the independent channel taps assumption is not correct. This method can further be improved by finding the taps statistics by employing Akaike weights.

5.4.2 Sparse Localization

The alignment is done in this algorithm as explained in Section 3.15. The results for the LASSO, OMP and S-TLS algorithms are shown in Fig. 5.12. An accuracy of 8 cm with 90%, 92% and 93% precisions are achieved for the OMP, LASSO and S-TLS algorithms.

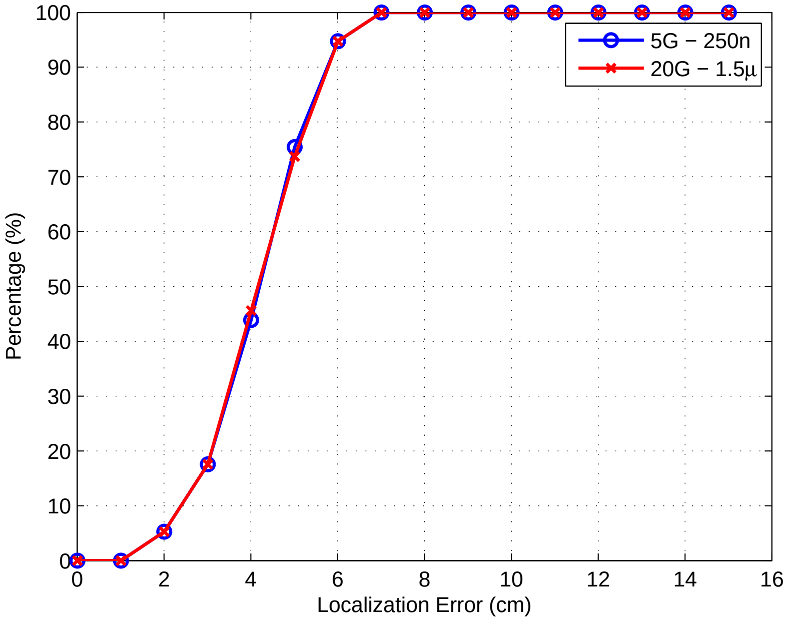


Figure 5.10: Comparison between a high resolution and shorter CIR vectors

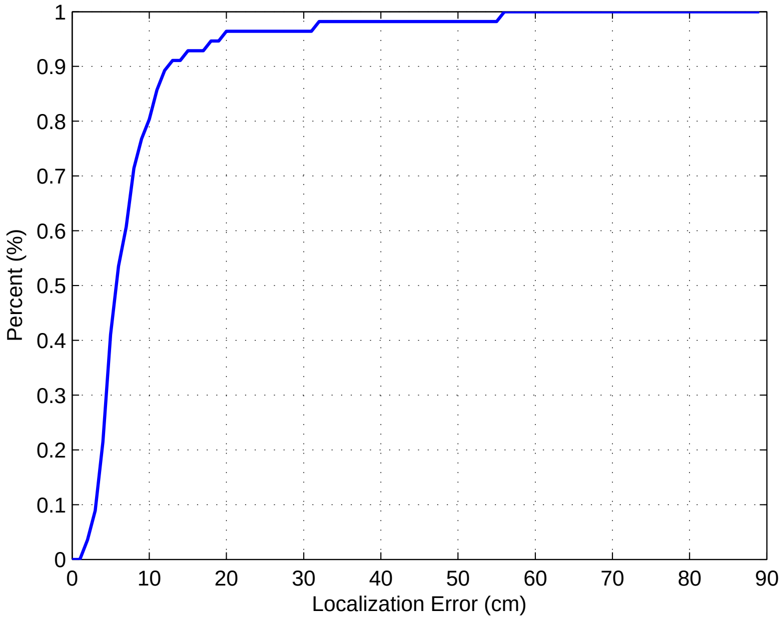
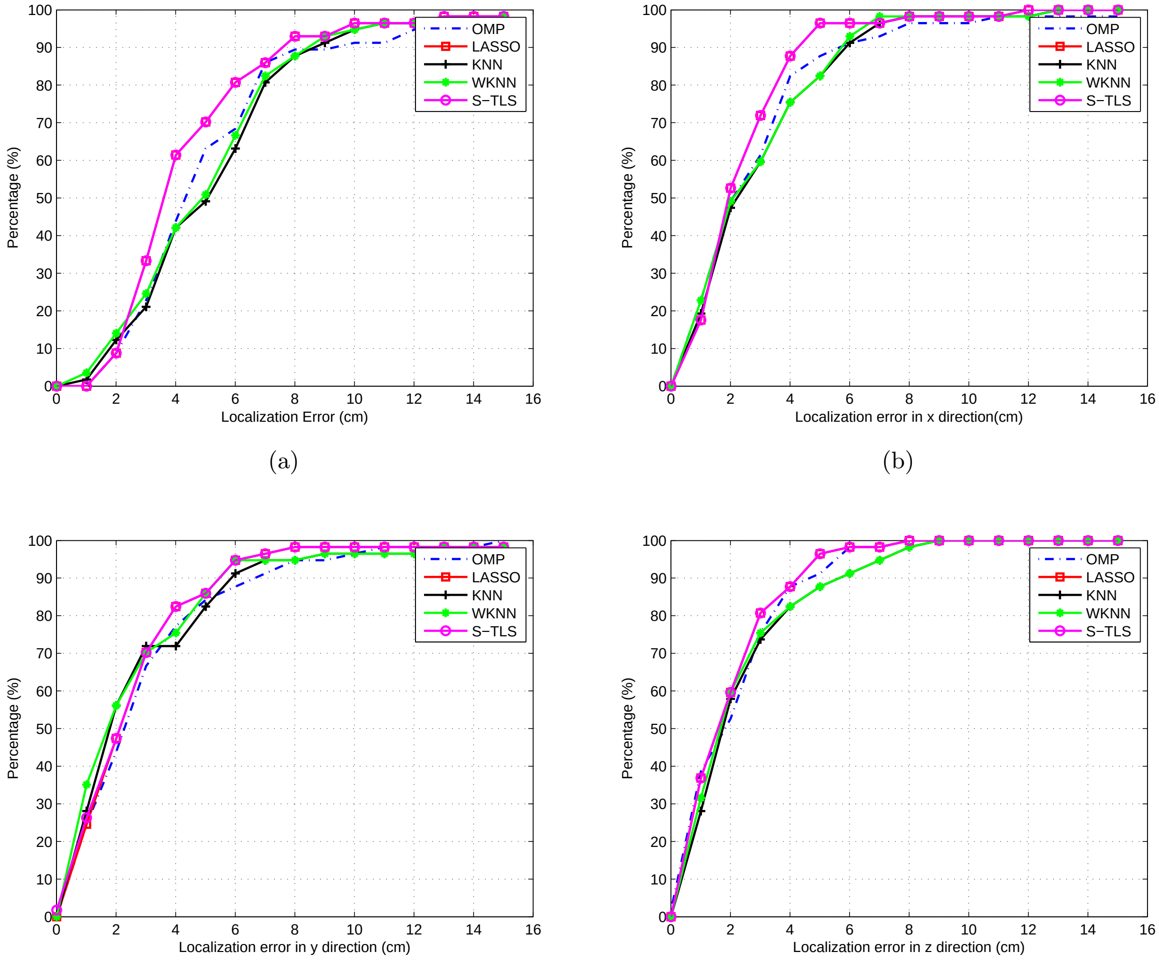


Figure 5.11: Localization performance of the maximum likelihood method

5.4.3 k-Nearest Neighbors

The KNN and WKNN algorithms are also applied with the same alignment strategy as explained in Section 3.15. The localization performance is depicted in Fig. 5.12. An



(a)

(b)

(c)

(d)

Figure 5.12: Localization performance for unsynchronized transmitter and receiver

accuracy of 8 cm with 87% precision for both methods is achieved.

In this case, similar to the synchronized case, the best localization performance is achieved for the LASSO and S-TLS algorithms. Besides, the simulation results of the OMP, WKNN and KNN algorithms show similar localization performance.

6 Conclusions and Future Works

With the introduction of UWB technology, many localization algorithms are proposed for an indoor channel. However, the utilization of UWB signals for underwater localization has never been proposed in literature. Therefore, through this work the feasibility of performing localization in water with UWB signals is investigated. A fingerprinting-based localization technique is deployed due to the unacceptable results that were achieved before by geometric techniques. In this technique, the physical space is first divided into a number of grid points. A set of location-dependent signatures is then gathered in the off-line phase for each grid point and is recorded in a matrix which is known as the dictionary matrix. The location-dependent signatures that are gathered for each region are the channel impulse responses (CIRs). In the on-line phase, the new received signature is compared with the signatures in our dictionary matrix and the location of the sensor can be estimated accordingly.

The influence of water on the propagation of electromagnetic waves and also some preliminary information about the UWB signals have been provided in Chapter 2. The set-up that is used for the demonstration of the localization algorithms have been described. Different instruments that are used in this set-up were explained and their specifications were provided. Afterwards, different methods for extracting the CIRs were evaluated and the cross-correlation method has been chosen due to its simplicity and the need for real-time localization. The CIRs have been further assessed to define the threshold to distinguish the noise level.

In Chapter 3, sparse localization which is based on the compressive sampling concept has been described. First, the system model was defined and the dictionary matrix which contains the CIRs was constructed for a number of grid points. Two scenarios have been considered for estimating the sensor location: having or not having time synchronization between the transmitter and the receiver in both phases. For an unsynchronized scenario, an alignment strategy was defined to align the CIRs to the same reference sample. Afterwards, the RIP condition for the dictionary matrix that assures the robustness to noise and uniqueness of the result was evaluated for the dictionary matrix that is used in our localization framework. As the dictionary matrix satisfies the RIP condition, four recovery algorithms have been proposed. Due to having an overdetermined set of linear equations and a dictionary matrix with full column rank, the least-squares method has first been assessed. For this method, the high variance causes unreliable estimates. As we can use the spatial sparsity for a more accurate location estimation, three sparse recovery methods were introduced. OMP was introduced as a greedy method and the LASSO was presented as a method which replaces the sparse approximation problem with a linear programming problem. In addition, the S-TLS algorithm which forces a sparsity condition on the total least-squares problem was assessed for two different approaches. One approach is to use the same dictionary matrix as we used in the LASSO and OMP methods to estimate the sensor location and the second method is to reduce the number of grid points in the on-line phase and then estimate the sensor location by the perturbation that is derived from the S-TLS algorithm from the estimated grid point to find the exact location of the sensor. Among the algorithms that are explained in this work, LASSO and the first approach of the S-TLS algorithm achieved the best estimation results. This better performance in comparison with the WKNN algorithm is due to the selection of different weights for the weighted average. The localization performance of the WKNN and the OMP algorithms is almost similar. The worst performance is achieved for the LS method due to the fact that it has a high variance and it never results in a sparse coefficient vector. By employing the S-TLS algorithm with the second approach, we achieve an acceptable localization performance. Although the computational complexity is reduced, due to the reduction in the number of grid points, the localization performance is degraded in comparison with other methods. Thus, the trade off is between the computational complexity and localization accuracy.

The existing algorithms for an indoor fingerprinting-based localization were also evaluated for underwater localization in Chapter 3. The first method is a traditional localization algorithm which is known as k-nearest neighbours. This algorithm is split into two methods: KNN and weighted KNN (WKNN). In both methods, the algorithm starts by looking through the recorded signatures in the off-line phase to find k signatures that have the minimum distance with our received signature in the on-line phase. Correspondingly, the location of these k signatures are found and in the KNN algorithm a simple location average is used to find the sensor location. In the WKNN algorithm, a weighted average is used and therefore a discrimination is made between a grid point that has a closer distance to our sensor with regard to the other grid points. The estimation results were shown in Chapter 6 and acceptable estimation results have been achieved for both methods. However, WKNN achieves a slightly better result KNN due to the employment of the weighted average. A statistical method was also evaluated in this work. This method employs the maximum likelihood estimator to find the sensor location. In this method, geo-regioning is performed and the physical space where the localization is performed is divided into a number of regions. For each region, the CIR amplitude distribution is defined. The sensor location is estimated by utilizing the maximum likelihood estimator which chooses the region with the highest probability as the estimated region. As there is no available channel model for the underwater medium, some simplifying assumptions were made. However, these assumptions resulted in unacceptable location estimation results. The last method that was evaluated in this work is a method which employs the neural network for localization. Instead of using the CIRs as the signature vectors, 7 parameters are extracted from the CIRs. However, simulations with this algorithm show unacceptable results due to the short distance between the grid points and therefore only small variations of these parameters with the location are observed.

At last, to perform real-time localization, the length of the signature vectors have been shortened by removing the unnecessary samples which resulted from a long time widow or an over-sampling of the CIRs. Therefore, a significant improvement is achieved in the algorithm's execution time.

6.1 Future Works

Some suggestions for the continuing this work are provided below.

Design of an omni-directional antenna: In this work, a fixed orientation is used for the antenna. However, in practice, the antenna is rotating and the antenna orientation is not fixed. The design of an omni-directional antenna for underwater communications is necessary to remove the dependency of our algorithms to the orientation of the antenna. With the current antennas, it is seen that a small change in the antenna orientation causes a significant error on our location estimates.

Addition of tracking algorithms: Adding a tracking algorithm can further simplify our localization methods. With a tracking algorithm, we can use the previous location information to estimate the sensor location. The previous location helps us to look only through the grid points in the vicinity of the previous grid point to find the location of our sensor instead of checking all the grid points for each new location.

Extension to multiple sensors: It is also possible to apply the proposed algorithms to localize multiple sensors. A new received signature vector in the on-line phase for this case is different from the signature vectors of our dictionary matrix as the signal that is received by the antenna is the result of a constructive and destructive addition of the signals of multiple sensors. Thus, this work can be further improved by extending the algorithms to the detection of multiple sensors in the on-line phase.

Bibliography

[1] A.F. Molisch. Ultra-wide-band propagation channels. *Proceedings of the IEEE*, 97(2):353 –371, feb. 2009.

[2] *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer, 2nd ed. 2009. corr. 3rd printing 5th printing. edition, February 2009.

[3] A.I. Al-Shamma'a, A. Shaw, and S. Saman. Propagation of electromagnetic waves at mhz frequencies through seawater. *Antennas and Propagation, IEEE Transactions on*, 52(11):2843 – 2849, nov. 2004.

[4] A. Shaw, A.I. Al-Shamma'a, S.R. Wylie, and D. Toal. Experimental investigations of electromagnetic wave propagation in seawater. In *Microwave Conference, 2006. 36th European*, pages 572 –575, sept. 2006.

[5] B. Kelley, K. Manoj, and M. Jamshidi. Broadband rf communications in underwater environments using multi-carrier modulation. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 2303 –2308, oct. 2009.

[6] M. Rhodes. Electromagnetic propagation in seawater and its value in military systems. In *SEAS DTC Technical Conference, Edinburg*, 2007.

[7] Umberto M. Cella, Ron Johnstone, and Nicholas Shuley. Electromagnetic wave wireless communication in shallow water coastal environment: theoretical analysis and experimental results. In *Proceedings of the Fourth ACM International Workshop on UnderWater Networks*, WUWNet '09, pages 9:1–9:8, New York, NY, USA, 2009. ACM.

[8] H.F. Guarnizo Mendez, F. Le Pennec, C. Gac, and C. Person. Deep underwater compatible wi-fi antenna development. In *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*, pages 1 –5, oct. 2011.

[9] Jaime Lloret, Sandra Sendra, Miguel Ardid, and Joel J. P. C. Rodrigues. Underwater wireless sensor communications in the 2.4 ghz ism frequency band. *Sensors*, 12(4):4237–4264, 2012.

[10] A.A. Abdou, A. Shaw, A. Mason, A. Al-Shamma'a, J. Cullen, and S. Wylie. Electromagnetic (em) wave propagation for the development of an underwater wireless sensor network (wsn). In *Sensors, 2011 IEEE*, pages 1571 –1574, oct. 2011.

[11] Liu Lanbo, Zhou Shengli, and Cui Jun-Hong. Prospects and problems of wireless communication for underwater sensor networks. *Wirel. Commun. Mob. Comput.*, 8(8):977–994, October 2008.

[12] J H Goh, A Shaw, and A I Al-Shamma'a. Underwater wireless communication system. *Journal of Physics: Conference Series*, 178(1):012029, 2009.

[13] `http://ens.ewi.tudelft.nl/Research/array/smartpeas/intro.php`.

[14] Aleix Garcia Miquel. Uwb antenna design for underwater communications, 2009. Universitat Politcnica de Catalunya.

[15] Z. Irahhauten. *Ultra-wideband wireless channel: measurements, analysis and modeling.* PhD thesis, TU Delft, 2009.

[16] Andreas Molisch. Wireless communications. 2005.

[17] `http://www.geozondas.com/Pulse_Gen_New/Units/Pulse_Generator_Unit_GZ1117AN.htm`.

[18] `http://www.home.agilent.com/agilent/product.jspx?cc=SG&lc=eng&nid=-34286.536908562&&cc=NL&lc=dut`.

[19] S.M. Yano. Investigating the ultra-wideband indoor wireless channel. In *Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th*, volume 3, pages 1200 – 1204 vol.3, 2002.

[20] G. Matz, A.F. Molisch, F. Hlawatsch, M. Steinbauer, and I. Gaspard. On the systematic measurement errors of correlative mobile radio channel sounders. *Communications, IEEE Transactions on*, 50(5):808 –821, may 2002.

[21] E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489 – 509, feb. 2006.

[22] Emmanuel J. Candes and Justin Romberg. Quantitative robust uncertainty principles and optimally sparse decompositions. *Found. Comput. Math.*, 6(2):227–254, April 2006.

[23] Emmanuel C, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements, 2005.

[24] E.J. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203 – 4215, dec. 2005.

[25] E.J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406 –5425, dec. 2006.

[26] David L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, 2006.

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

[28] D. Dardari, Chia-Chin Chong, and M.Z. Win. Threshold-based time-of-arrival estimators in uwb dense multipath channels. *Communications, IEEE Transactions on*, 56(8):1366 –1378, august 2008.

[29] C. Steiner and A. Wittneben. Robust time-of-arrival estimation with an energy detection receiver. In *Ultra-Wideband, 2009. ICUWB 2009. IEEE International Conference on*, pages 834 –838, sept. 2009.

[30] Christoph Steiner, Frank Althaus, Florian Trösch, and Armin Wittneben. Ultra-wideband geo-regioning: A novel clustering and localization technique. *EURASIP Journal on Advances in Signal Processing, Special Issue on Signal Processing for Location Estimation and Tracking in Wireless Environments*, November 2007.

[31] Christoph Steiner. *Location Fingerprinting for Ultra-Wideband Systems - The Key to Efficient and Robust Localization.* PhD thesis, PhD Thesis, 2010.

[32] Volkan Cevher, Marco F. Duarte, and Richard G. Baraniuk. Distributed target localization via spatial sparsity. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Laussane/Switzerland, Aug. 2008.

[33] S. Nikitaki and P. Tsakalides. Localization in wireless networks via spatial sparsity. In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, pages 236 –239, nov. 2010.

[34] Chen Feng, S. Valaee, and Zhenhui Tan. Multiple target localization using compressive sensing. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1 –6, 30 2009-dec. 4 2009.

[35] Bowu Zhang, Xiuzhen Cheng, Nan Zhang, Yong Cui, Yingshu Li, and Qilian Liang. Sparse target counting and localization in sensor networks based on compressive sensing. In *INFOCOM, 2011 Proceedings IEEE*, pages 2255 –2263, april 2011.

[36] David L. Donoho and Michael Elad. Elad m 2003 optimally sparse representation in general (non-orthogonal) dictionaries via 1 minimization. In *Proc. Natl Acad. Sci. USA 100 2197202.*

[37] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40 –44 vol.1, nov 1993.

[38] S. Mallat and Z. Zhang. Adaptive time-frequency decomposition with matching pursuits. In *Time-Frequency and Time-Scale Analysis, 1992., Proceedings of the IEEE-SP International Symposium*, pages 7 –10, oct 1992.

[39] J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231 – 2242, oct. 2004.

[40] M.A. Davenport and M.B. Wakin. Analysis of orthogonal matching pursuit using the restricted isometry property. *Information Theory, IEEE Transactions on*, 56(9):4395 –4401, sept. 2010.

[41] Wasim Q. Malik and Ben Allen. Wireless sensor positioning with ultrawideband fingerprinting. In *Antennas and Propagation, 2007. EuCAP 2007. The Second European Conference on*, pages 1 –5, nov. 2007.

[42] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[43] Ewout van den Berg and Michael P. Friedlander. Probing the pareto frontier for basis pursuit solutions, 2008.

[44] Hao Zhu, G. Leus, and G.B. Giannakis. Sparsity-cognizant total least-squares for perturbed compressive sampling. *Signal Processing, IEEE Transactions on*, 59(5):2002 –2016, may 2011.

[45] P. Bahl and V.N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775 –784 vol.2, 2000.

[46] Mauro Brunato and Roberto Battiti. Statistical learning theory for location fingerprinting in wireless lans. *Comput. Netw.*, 47(6):825–845, April 2005.

[47] Frank Althaus, Florian Trösch, and Armin Wittneben. Geo-regioning in UWB networks. In *14th IST Mobile and Wireless Communications Summit 2005*, June 2005.

[48] Frank Althaus, Florian Trösch, and Armin Wittneben. UWB geo-regioning in rich multipath environment. In *IEEE Vehicular Technology Conference, VTC Fall 2005*, September 2005.

[49] A. Chehri, P. Fortier, and P.-M. Tardif. Geolocation for uwb networks in underground mines. In *Wireless and Microwave Technology Conference, 2006. WAMICON '06. IEEE Annual*, pages 1 –4, dec. 2006.

[50] Simon Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, 2 edition, July 1998.

[51] Yu Hen Hu. *Handbook of Neural Network Signal Processing*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2000.

[52] Ben Krse, Ben Krose, Patrick van der Smagt, and Patrick Smagt. An introduction to neural networks, 1993.

[53] C. Nerguizian, C. Despins, and S. Affes. Geolocation in mines with an impulse response fingerprinting technique and neural networks. In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 5, pages 3589 – 3594 Vol. 5, sept. 2004.

[54] A. Taok, N. Kandil, S. Affes, and S. Georges. Fingerprinting localization using ultra-wideband and neural networks. In *Signals, Systems and Electronics, 2007. ISSSE '07. International Symposium on*, pages 529 –532, 30 2007-aug. 2 2007.

[55] Anthony Taok, Nahi Kandil, and Sofiene Affes. Neural networks for fingerprinting-based indoor localization using ultra-wideband. *Journal of Communications*, 4(4), 2009.

[56] Luciano Leins and Christoph Steiner. Neural network based geo-regioning. In *IEEE International Conference on Ultra-Wideband, ICUWB 2008*, volume 2, pages 229–232, September 2008.

[57] C. Nerguizian and V. Nerguizian. Indoor fingerprinting geolocation using wavelet-based features extracted from the channel impulse response in conjunction with an artificial neural network. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, pages 2028 –2032, june 2007.

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

[59] Lei Yu, M. Laaraiedh, S. Avrillon, and B. Uguen. Fingerprinting localization based on neural networks and ultra-wideband signals. In *Signal Processing and Information Technology (ISSPIT), 2011 IEEE International Symposium on*, pages 184 –189, dec. 2011.