Graz University of Technology

CHANNEL IMPULSE RESPONSE MEASUREMENT SOFTWARE FOR THE M-SEQUENCE CHANNEL SOUNDER

Author: Lafer Manuel, 0830301
Date: Graz, December 9, 2013
Rev.: 1.4

Contents

1 Introduction

The UWB channel is well suited for indoor localization applications due to its high temporal resolution. A theoretical description of the channel and the channel modeling can be found in [1, 2]. UWB channel measurements (or channel sounding) have been done in many publications. In [3], measurements in four different building types using a vector network analyzer (VNA) were done. A UWB channel model was established in [4] using a carrier modulated by a PN-sequence. In [5], UWB channel impulse responses (CIRs) in an office building were measured using a VNA. The focus there was to make use of the multipath components (MPCs) in the CIR for localization.

This project had the same goal, to measure UWB CIRs for localization, but uses a different method for acquiring the CIR. The method used is the same as in [4], a carrier modulated by a PN- sequence, and the device used for that is the M-Sequence device by Ilmsens. The M-Sequence device is used, because measuring a campaign using the VNA takes a lot of time (approximately one minute per measurement point), therefore the M-Sequence device should provide an alternative which allows faster measurement.

The following chapters provide an overview over channel sounding techniques (Chap. 2), information on the M-Sequence device itself and the measurement method (Chap. 3), on Pre-processing (Chap. 4), a chapter describing the MatLab™software in detail (Chap. 5) and a comparison of the results to the VNA measurement campaigns (Chap. 6).

2 Channel Sounding

This overview over channel sounding is based on Chapter 8: Channel Sounding in Wireless Communications [6]. Basically, channel sounding is sending out a signal from a transmitter (TX) which penetrates the channel and is received at the receiver (RX). By comparing transmitted and received signal, the desired channel system function (e.g. the impulse response) can be obtained. The transmit signal $s(t)$ consists of pulses $p(t)$ which are periodically repeated with the repetition interval T_{rep}

$$s(t) = \sum_{i=0}^{N} p(t - iT_{rep})$$

Fig. 2.1 shows the block diagram of the channel sounder principle. The transmit pulse $p(t)$ is the convolution of the basis pulse $\tilde{s}(t)$ and the impulse response of the transmit filter $g(t)$

$$p(t) = \tilde{s}(t) * g(t)$$

AGC denotes automatic gain control and f_c is the clock frequency. GPS is one of the possible methods for synchronization between TX and RX.
The repetition interval T_{rep} is important in time-variant systems, i.e. environments, in which the channel properties change over time, due to movement of TX, RX, interacting objects or any combination of those. As the channel impulse response for any pulse $p(t)$ can be considered a "snapshot" of the channel, T_{rep} must be smaller than the time over which the channel properties change in order to track those changes.

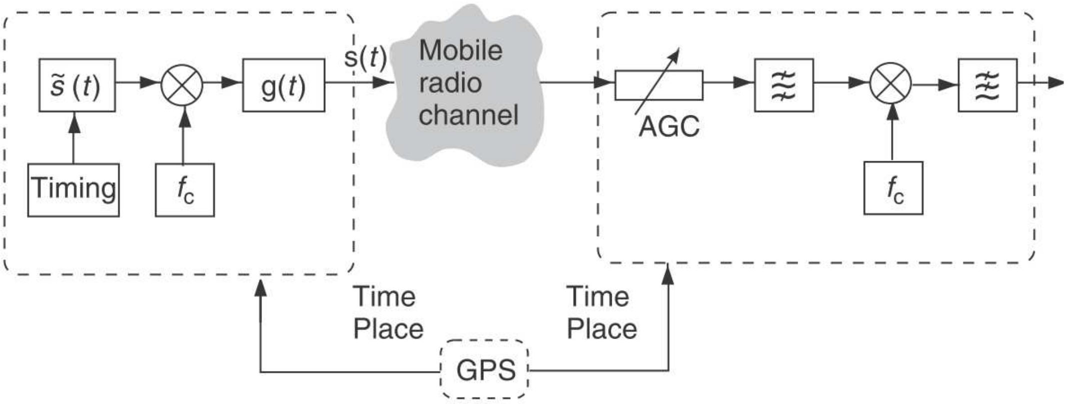


Figure 2.1: Principle of the channel sounder [6]

The sounding signal should fulfill some requirements to perform efficient measurements:

- *Large bandwidth*: the bandwidth determines the achievable delay resolution as it is inversely proportional to the shortest temporal change of the sounding signal.
- *Large time bandwidth product*: a time bandwidth product TW larger than unity allows the transmission of higher energy which results in a higher SNR at the receiver. To achieve a

TW greater than unity, the sounding signal has to have a duration longer than the inverse of the bandwidth.

- *Signal duration*: to fulfill the above mentioned TW requirement, the signal duration should be long, but on the other hand it is not allowed to be longer than the time in which the channel is considered approximately constant (coherence time). Therefore the pulse repetition time T_{rep} should be larger than the duration of the pulse $p(t)$ and the maximum excess delay of the channel but smaller than the coherence time.
- *Power-spectral density*: $|P_{TX}(j\omega)|^2$ should be uniform over the bandwidth of interest to ensure the same quality of channel estimation over this bandwidth. Outside the bandwidth of interest, there should be little signal energy for high efficiency.
- *Low crest factor*:

  $$C_{crest} = \frac{\text{Peak amplitude}}{\text{rms amplitude}} = \frac{max\{s(t)\}}{\sqrt{\overline{s^2(t)}}}$$

  The transmit power amplifier is used efficiently if the signal has a low crest factor.
- *Good correlation properties*: for correlation based measurements, the Autocorrelation Function (ACF) should have a high Peak to Off Peak (POP) ratio and a zero mean (to allow unbiased estimates).

There are two different approaches for channel sounding, the time-domain and the frequency-domain measurement, which are described in the following sections.

2.1 Time-domain Measurements

In a time-domain measurement, the impulse response is measured directly by sending out a sequence of pulses. The channel is assumed to be slowly time variant, then the measured impulse response $h_{meas}(\boldsymbol{p}_l, \tau)$ at the position $\boldsymbol{p}_l$ is the convolution of the true channel impulse response $h(\boldsymbol{p}_l, \tau)$ with the impulse response of the sounder $\tilde{p}(\tau)$:

$$h_{meas}(\boldsymbol{p}_l, \tau) = \tilde{p}(\tau) * h(\boldsymbol{p}_l, \tau)$$

$\tilde{p}(\tau)$ is the convolution of the transmitted pulse shape $p_{TX}(\tau)$ and the RX filter impulse response $p_{RX}(\tau)$:

$$\tilde{p}(\tau) = p_{TX}(\tau) * p_{RX}(\tau)$$

if the channel and the transceiver are linear. To minimize the impact of the measurement system on the result, the sounder impulse response should be as close to a Dirac delta function as possible.

2.1.1 Impulse Sounder

This type of channel sounder sends out a sequence of short pulses $p_{TX}(\tau)$. With shorter pulses a better spatial resolution can be achieved. To get a good SNR the pulses should contain much energy. The receive filter is a bandpass filter and ideally $p_{RX}(\tau)$ should not have any influence, so that the sounder impulse response becomes:

$$\tilde{p}(\tau) = p_{TX}(\tau)$$

As the pulses should contain much energy they have high peak powers. RF components with such requirements are either expensive or have other disadvantages, e.g. non-linearities. Furthermore, impulse sounders have a low resistance to interference, i.e. interfering signals might be interpreted as part of the CIR.

2.1.2 Correlative Sounder

The convolution of $p_{TX}(\tau)$ with $p_{RX}(\tau)$ determines the impact of the measurement system on the observed impulse response. A general relationship between $p_{TX}(\tau)$ and $p_{RX}(\tau)$, which is well known from digital communication theory, is that the SNR at the receiver filter output is maximized, if the receiver filter is matched to the transmit waveform. By concatenation of transmit and receive filter, the sounder impulse response becomes the autocorrelation function (ACF) of the transmit filter if $p_{TX}(\tau) = p_{RX}(\tau)$:

$$\tilde{p}(\tau) = p_{TX}(\tau) * p_{RX}(\tau) = R_{p_{TX}}(\tau)$$

Therefore, the sounding pulse should have a high autocorrelation peak $R_{p_{TX}}(0)$ and low ACF sidelobes, which means, it should be a good approximation of a delta function.
The most used sounding sequences in practice are Pseudo Noise (PN) sequences, especially popular are Maximum-length PN sequences (M-Sequences) which can be created by a shift register with feedback. The M-Sequence has an ACF with a periodicity of M_c, has only a single peak of height M_c and a POP ratio of M_c.
In case of time-varying channels, some extra care has to be taken. The basic principle requires, that the channel is the same at the beginning of the PN sequence and at the end of it. If that does not hold, correction procedures are needed.

2.1.3 Swept Time Delay Cross Correlator (STDCC)

The aim of this method is to reduce the sampling rate as a typical correlative sounder has to sample at Nyquist frequency. To reduce the sampling rate, the STDCC uses just one sample value for each m-sequence, taken at the maximum of the ACF. This allows sampling at rate T_{rep}. The time basis of the RX is shifted with respect to the TX for each repetition, so K_{scal} transmissions of the m-sequence result for a single impulse response $h(\tau_i), i = 1, ..., K_{scal}$ are needed. The advantage of this method is a higher delay resolution and reduced sampling rate, but the time for each measurement is increased by the factor K_{scal}.

2.2 Frequency-domain Measurements: Vector Network Analyzer (VNA)

In frequency-domain measurements, the channel transfer function is directly estimated. It is important, that the transmit waveform $p(\tau)$ has a power spectrum $|P(jw)|^2$ that is approximately constant over the bandwidth of interest. One way to transmit the pulses is by sending one frequency at a time, increasing the frequency linearly over a range of frequencies, this is called a sweep. This type of measurement is very slow, therefore it is only usable in static environments as the channel is not allowed to change during one sweep.

A different approach is to send different frequencies at the same time by generating different sinusoidal sounding signals with different weights, phases and frequencies and transmit them simultaneously.
A VNA does a slow sweep over the frequency range of interest. It measures the S-parameters of the device under test (DUT), which can be the wireless channel. The parameter S_{21} is in that case the channel transfer function of the transmitted frequency. Doing a sweep over the whole bandwidth of interest, a sampled version of the transfer function $H(\tau, f)$ is obtained.
As for all kind of channel sounding, a calibration is needed. For VNAs the so called SOLT calibration (Short Open Loss Termination) is used. The frequency response of the VNA itself is measured and in subsequent measurements, the VNA compensates for this frequency response so that only the frequency response of the DUT is measured. Antennas are not taken into account in this type of calibration, but this is not a problem if they are considered to be part of the channel. If some antenna effects have to be taken into account, a separate calibration is needed.
Results of VNAs are usually accurate and straightforward, but there are some things to consider:

- The measurement is slow due to the sweep over the bandwidth of interest, repetition rates cannot exceed a few Hz, therefore the channel is not allowed to change drastically during measurement, which allows VNA measurements mainly in static environments.
- As TX and RX are often in one device, this does limit the distance between TX and RX antenna.

2.3 Implementation Issues

2.3.1 Inverse Filtering

It is possible to use a receive filter which is not optimized on the SNR but on the POP ratio. This results in a worse SNR, which is practically unproblematic, but in smaller sidelobes, which is advantageous as they can lead to additional errors.
The receiver filter transfer function for inverse filtering is chosen as $1/P_{TX}(f)$ in the bandwidth of interest, the total transfer function $P_{IF}(f)$ is:

$$P_{IF}(f) = P_{TX}(f) \cdot \frac{1}{P_{TX}(f)} \approx 1$$

The inverse filter compensates for distortions by the transmit filter.

2.3.2 Averaging

Averaging increases the SNR by $10 \cdot log_{10} M$ dB, where M is the number of averaged impulse responses. It is assumed that the channel does not change during the M measurements and that noise is statistically independent.

2.3.3 Synchronization

Synchronization between TX and RX is important in order to establish a common frequency and time basis. This can be quite complicated in wireless channels due to multipath propagation and time variations in the channel. Different approaches are in use:

1. *Synchronization by cables* is possible in indoor environments. For distances of up to 10m, coaxial cables are used, for longer distances fiber-optic cables are needed. For both, the synchronization signal is sent over a known and well-defined medium.
2. The *Global Positioning System (GPS)* can be used in outdoor systems and helps to establish common time and frequency references. Additionally, also the measurement location can be recorded. A disadvantage is that both TX and RX need a line-of-sight connection to GPS satellites.
3. *Rubidium clocks* at TX and RX can be synchronized at the beginning of a measurement, they are extremely stable (typical relative drifts of 10^{-11}) allow synchronization for several hours.
4. *Measurement without synchronization* means doing the synchronization over the measured channel itself. The receiver triggers the recording if a certain threshold is exceeded. This approach is simple but noise and interferences can erroneously trigger the recording and it is not possible to determine absolute delays.

For further information on the measurement methods, synchronization and detailed descriptions see Chap. 8 Channel Sounding in [6].

3 Correlative Channel Sounding using Pseudo Random Sequences

3.1 Maximum- Length Sequence

A M-Sequence (or Maximum- Length Sequence, MLS) is a periodic binary pseudo random sequence. It has a length of $2^N - 1$, where N is a positive integer. Such a periodic sequence can be generated using a digital shift register containing N bits. The binary sequence acts as stimulus and interacts with the scenario. The M-Sequence is often chosen ahead of other alternatives as it has a very short auto-correlation function, i.e. a flat spectrum over a large bandwidth.

The signal is affected by the environment and will be deformed according to the objects in this environment. The receiver captures the signal using a Track&Hold (T&H) circuit and transforms the analog signal into the digital domain using an ADC. The block diagram of the specific M-Sequence based measurement system used in this project is shown in Fig. 3.1.

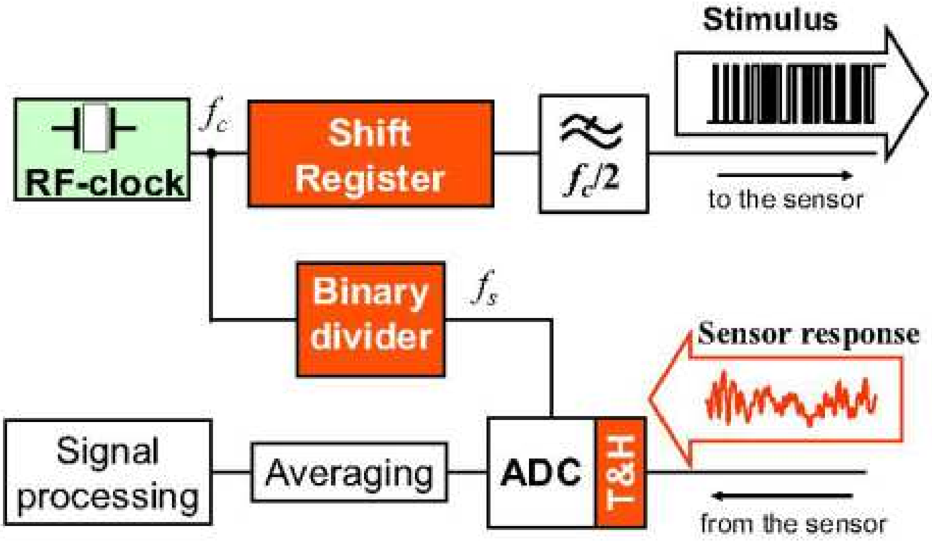


Figure 3.1: Block diagram UWB M-Sequence device [7]

The result depends on the application, it can be either a complex impulse response function (CIR), a step response function or a frequency response function. In the case of this project, a CIR is the result.

The received signal is cross-correlated with the original sequence which results in the CIR. Fig. 3.2 shows in the upper plot an ideal M-Sequence with $2^N - 1 = 511$ chips, where N is the number of bits. The plot in the middle shows the delayed M-Sequence, affected by the environment and sampled at the receiver. The delay equals 346 chips, which can be seen in the bottom plot, which shows the correlation of the noisy signal with the originally sent M-Sequence. It is easy to find the delay as there is a very high correlation gain of $2^N - 1$.

A very precise and mathematical description of M-Sequences can be found in [8].

The stimulus will occupy a frequency spectrum from DC to $f_c/2$, where f_c is the clock frequency, if a suitable pseudorandom noise-code is chosen. In this frequency band the power is nearly constant and the energy will drop drastically beyond $f_c/2$.

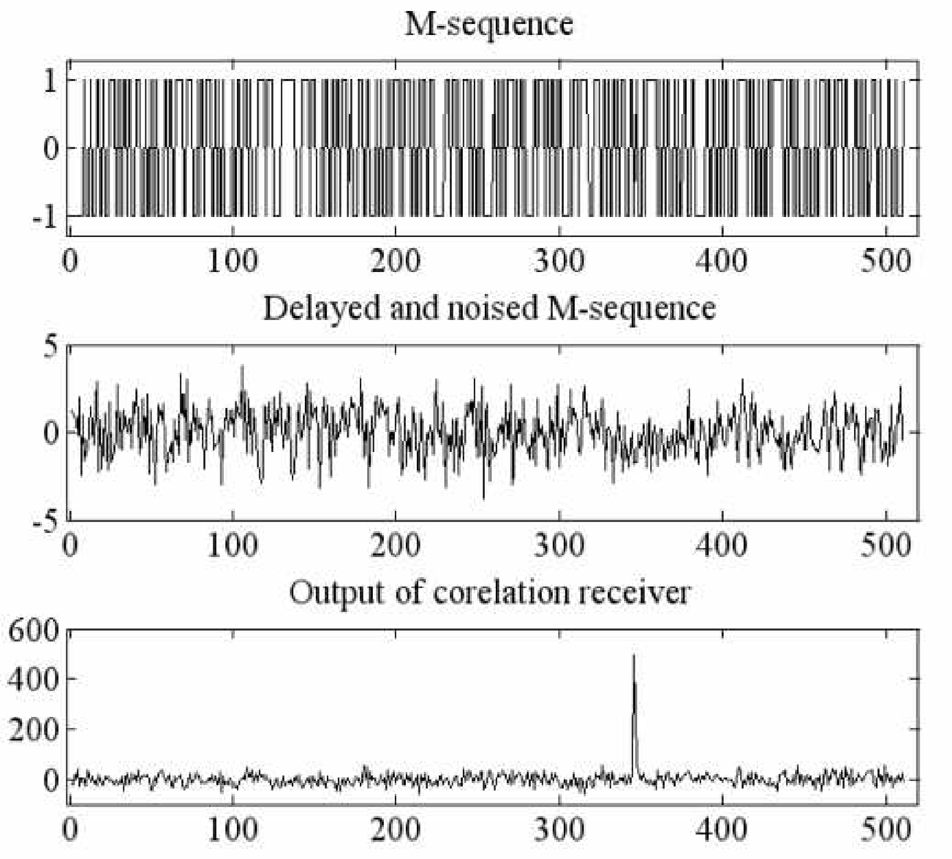


Figure 3.2: Principle of M-Sequence, the binary pseudo random sequence (upper plot), the received noisy signal (middle plot) and the correlated signal (lower plot) [9]

As the signal takes a large bandwidth in the frequency domain, the pulse in the time domain is very short. It would therefore be necessary that the receiver electronic works at a very high sampling rate, but as the pseudorandom-noise is a periodic signal and therefore deterministic, sub-sampling for data capturing can be applied. This drastically reduces the sampling requirements for the receiver electronic. The timing of data capturing must be very exact as ultra-wideband signals change their amplitude rapidly. An example for sub-sampling can be seen in Fig. 3.3 where capturing is distributed over two periods.

As the M-Sequence is a cyclic code, the starting time (or delay) of the received signal is not determinable without any further information. One possibility to determine it, is to get the start sample of the M-Sequence generator from the transmitter. If this is not possible or the transmitter does not allow that, some other mechanism is needed. In case of the used M-Sequence device (described in the next section), getting the start phase is not possible.

One way to determine the start of the received signal is to do a reference measurement at a known distance to determine the start of the impulse response. In this project the problem is solved by the device calibration as described in Section 4.1.

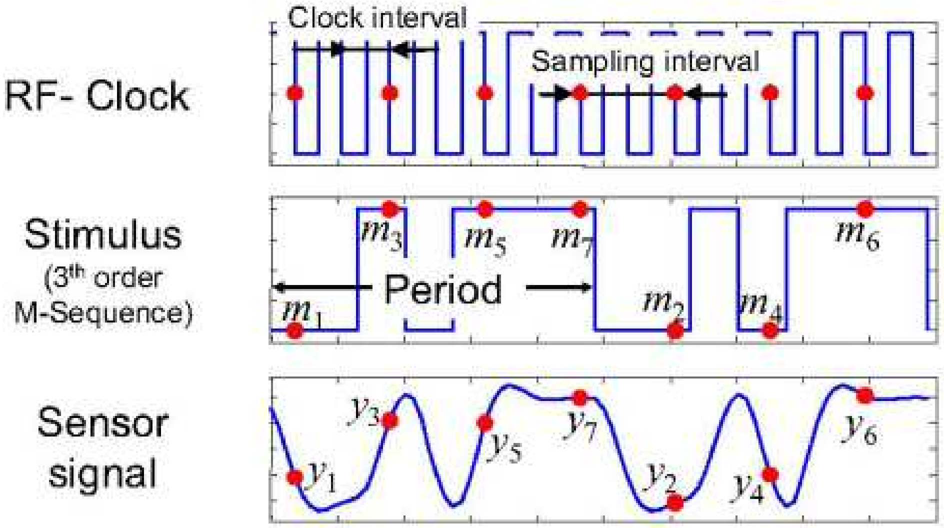


Figure 3.3: Example of sub-sampling over two periods [7]

3.2 ILMsens M-Sequence Device

The M-Sequence device is a sensor device manufactured by ILMsens (TU Ilmenau Service GmbH) which allows measuring the impulse response of UWB signals using a M-Sequence.
The M-Sequence device used in this project consists of two units, the RF electronics unit and the power supply unit (see Fig. 3.4). The device includes a UWB signal generator for wideband pseudorandom codes and two receivers which operate synchronously and support sub-sampling.

Figure 3.4: RF electronics unit (left) and power supply unit (right) [10]

The device can be operated in baseband from 0.1 .. 3.2 GHz or in FCC passband from 3.8 .. 10.2 GHz using an I/Q up-down converter.
In order to use the device in baseband, connections are made directly to the connectors named Rx1, Rx2 and Tx, to use the FCC passband, SMA bridges must be installed according to Fig. 3.5. The antennas are then connected via the RF1-In, RF2-In and RF-Out connectors.
The device is connected to the computer via USB. For more details on the M-Sequence device

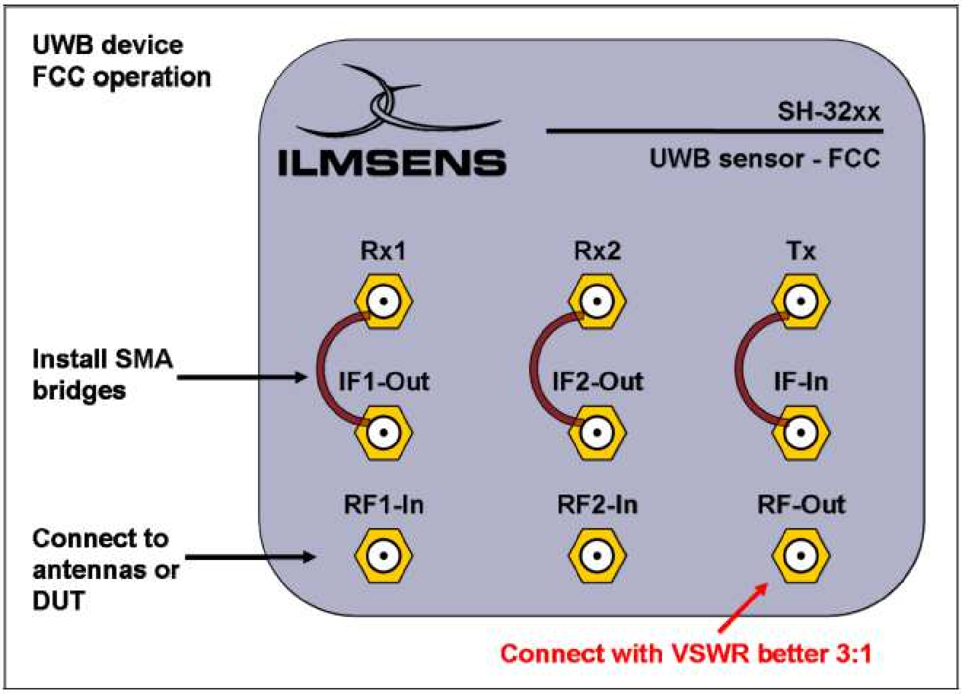


Figure 3.5: Connectors of the M-Sequence device FCC configuration [10]

see the hardware manual [10], information on the software can be obtained from the software manual [11].

4 Pre-Processing of Measurement Data

Before any interpretations of the measured signal or distance measurements can be done, some pre-processing is necessary. From the measured signal $r_h(\tau)$ influences of the system impulse response $r_{\text{sys}}(\tau)$ and the internal crosstalk $r_{\text{cross}}(\tau)$ of the device have to be eliminated. Those influences are measured in a calibration step which is done before the actual measurements. The calibrated impulse response $h(\tau)$ is convolved with a suitable pulse shape $r(\tau)$ resulting in the shaped impulse response $h_{sh}(\tau)$. This is done to allow the selection of the desired frequency band.

Fig. 4.1 shows a block diagram of those pre-processing steps, cal represents the calibration and $r(\tau)$ the pulse shaping.

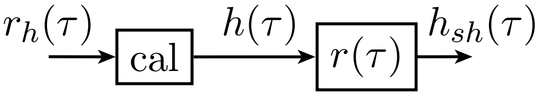


Figure 4.1: Block diagram of pre-processing steps

4.1 Calibration

The block diagram in Fig. 4.2 shows the influences of the channel sounder and the measurement setup. The internal crosstalk between the transmitter and the receivers is denoted as $H_{\text{cross},1}$ and $H_{\text{cross},2}$. The block $H_{\text{sys,TX}}$ models the part of the system response on the transmit side, $H_{\text{sys,RX1}}$ and $H_{\text{sys,RX2}}$ model the system response on the receiver sides one and two, respectively. These responses have to be measured before the channel measurements. They are used to compute the actual channel impulse response.

The calibration is not only used to remove the system influences but it is also necessary to obtain a CIR with correct delay information. As described in Section 3.1, the M-Sequence is a cyclic code and the code delay (delay of the received signal due to the cyclic code) is not determinable without any further information. One possibility to determine it is to get the start sample of the M-Sequence generator from the transmitter. If this is not possible or the transmitter does not allow that, some other mechanism is needed. In case of the used M-Sequence device, getting the start sample is not possible.

Due to the calibration the code delay is canceled out and the correct delay information is in the received CIR. How this is done is shown in Section 4.1.3.

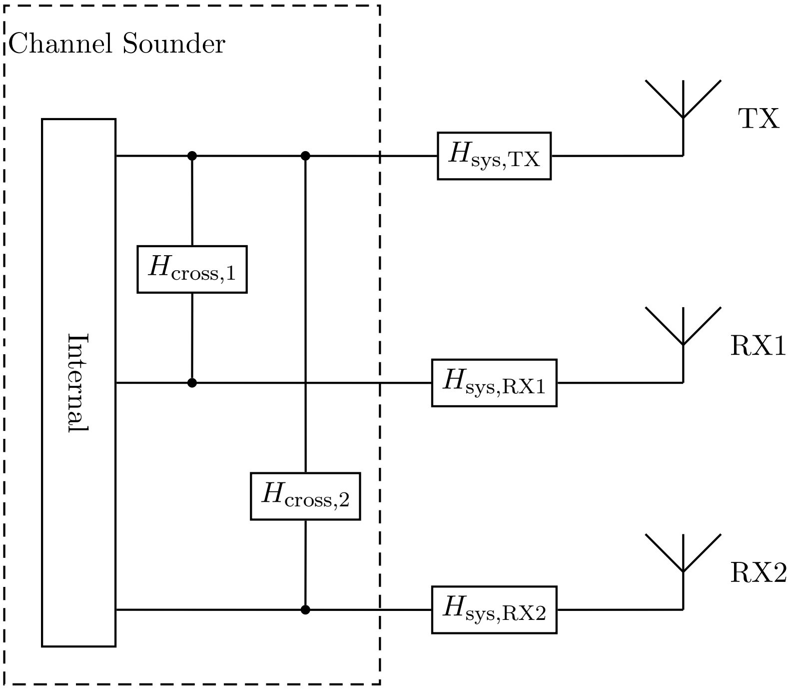


Figure 4.2: Block diagram measurement influences

4.1.1 Measuring the Crosstalk

To measure the crosstalk, the transmitter cable is disconnected and a 50 Ω match is mounted on the transmit port. The receive antennas used for the actual measurements are connected to the respective receiver ports. This allows to measure only the influence on the receive ports inside the device as most of the transmit signal is attenuated and not received by the antennas.
Fig. 4.3 shows the setup of the crosstalk measurement.

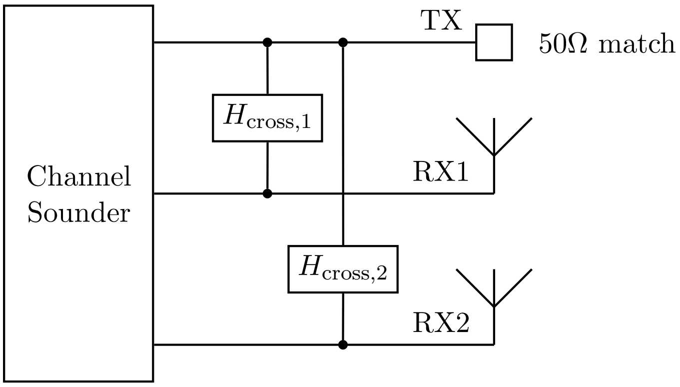


Figure 4.3: Block diagram of the crosstalk measurement

4.1.2 Measuring the System Impulse Response

The system response has to be measured for each channel individually. To measure it, the cables of the transmitter and the receiver are disconnected from the antennas and the cables are connected directly using an appropriate connector.
As the measurement does not include the antennas, their additional delay has to be accounted for in the computations. Also, the used connector is considered part of the system response but is not used in the actual channel measurement. The block diagram of the system response measurement setup is shown in Fig. 4.4.

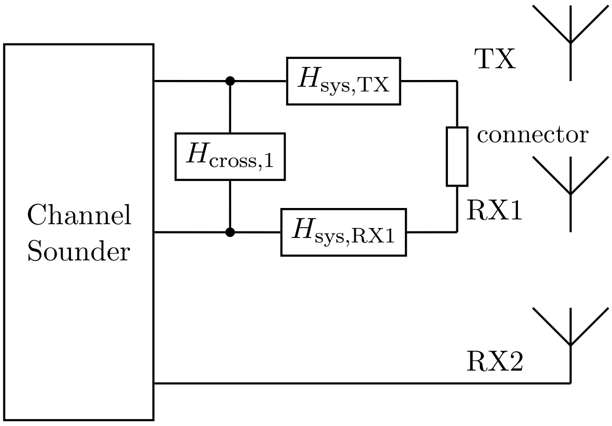


Figure 4.4: Block diagram of the system response measurement of channel 1

4.1.3 Computing the Correct Impulse Response

Fig. 4.5 shows the calibration process and how to obtain the actual channel impulse response (depicted as CIR). The channel sounder used in this project does not have the AGC gain & attenuation compensation functionality.
The CIR can be computed as follows (receive channel indices are dropped):
The system response measurement gives

$$H_{\text{sys,meas}}(f) = H_{\text{sys}}(f) \cdot 1 + H_{\text{cross}}(f)$$

where the measured system response $H_{\text{sys,meas}}$ consists of the actual system and the device internal crosstalk. The 1 depicts the channel for connected cables.
The actual system response $H_{\text{sys}} = H_{\text{sys,TX}} \cdot H_{\text{connector}} \cdot H_{\text{sys,RX}}$ consists of the transmit part, the connector and receive part.
Measuring the UWB channel using the measurement configuration results in:

$$H_{\text{ch,meas}}(f) = H_{\text{sys}}(f) \cdot H_{\text{ch}}(f) + H_{\text{cross}}(f)$$

where $H_{\text{ch}}(f)$ is the actual channel impulse response.
Inserting the system response $H_{\text{sys}}(f)$ gives:

$$H_{\text{ch,meas}}(f) = [H_{\text{sys,meas}}(f) - H_{\text{cross}}(f)] \cdot H_{\text{ch}}(f) + H_{\text{cross}}(f)$$

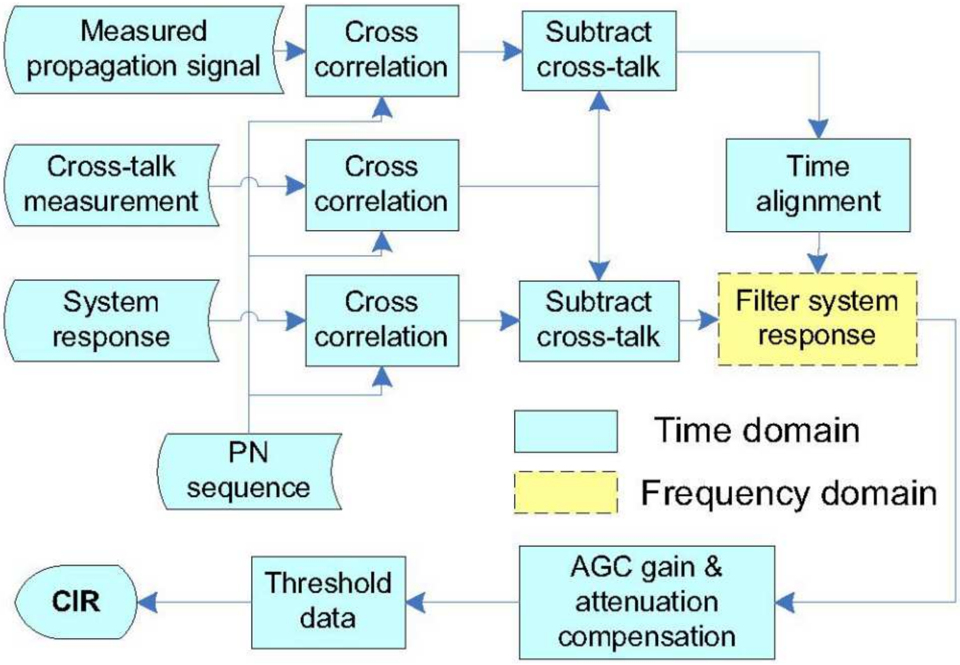


Figure 4.5: Calibration block diagram [12]

To compute the actual channel, use:

$$H_{\text{ch}}(f) = \frac{H_{\text{ch,meas}} - H_{\text{cross}}(f)}{H_{\text{sys,meas}} - H_{\text{cross}}(f)}$$

To compute the CIR $h(\tau)$ in time domain, use

$$h(\tau) = \text{IDFT}\left[H_{\text{ch}}(f)\right]$$

where IDFT[·] is the inverse Fourier Transform.
The CIR still contains the antenna influences and the connector used during the system response measurements. To account for them, the CIR has to be shifted by an additional delay τ_{add} which is described in Section 5.5.1.
In Fig. 4.6 a comparison between the raw measured CIR $r_h(\tau)$ and the calibrated CIR $h(\tau)$ is shown. The raw measured CIR was shifted manually to allow a comparison as it does not have the correct delay. In the raw CIR it can be seen that there is a smaller peak shortly after each signal component, e.g. after the line-of-sight peak at approximately 21 ns. This is because of the influence of the system response. It is not present anymore in the calibrated CIR.

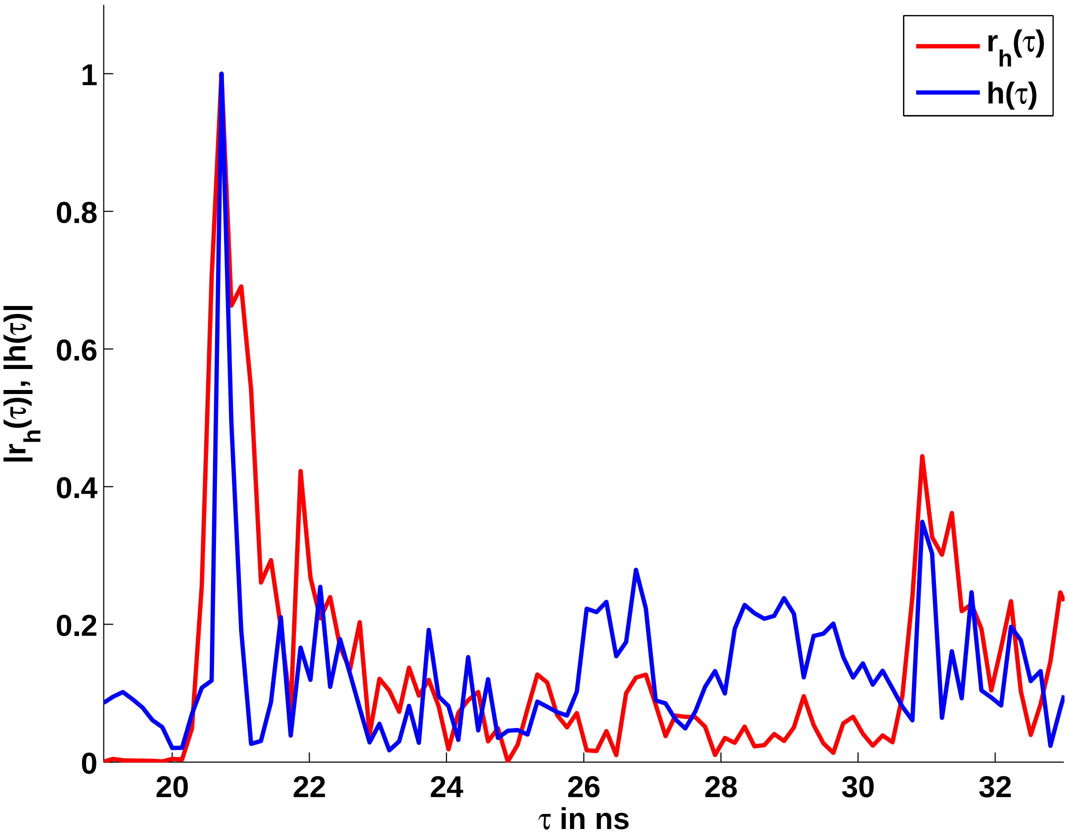


Figure 4.6: Comparison of the raw measured CIR $r_h(\tau)$ and the calibrated CIR $h(\tau)$

4.2 Pulse Shaping

Pulse shaping is done to select a desired frequency range from the signal.
The received signal is shaped using a raised cosine pulse. It works in a manner similar to that described in [5] where a VNA was used for measuring the CIR. A difference is that the signal obtained using a VNA is in frequency domain whereas the M-Sequence device in this project delivers a time domain signal. The pulse shape and bandwidth are determined by TX and RX filters and the selected frequency band. A commonly used model for the impulse response $h(\tau)$ is

$$h(\tau) = \sum_i a_i \delta(\tau - \tau_i) + n(\tau)$$

where a_i is the complex gain, τ_i the delay of the ith multipath component and $n(\tau)$ the noise term which also includes diffuse multipath.
For the pulse shaping, a suitable pulse form has to be defined first, here it is a raised cosine pulse. Typically, this is done in the frequency domain. The raised cosine pulse is defined by its center frequency f_c, the roll-off factor β and its bandwidth, which is typically defined as the inverse of the pulse duration T_p. This results in a 3-dB bandwidth $B_N = (1+\beta)/T_p$. The calculated bandwidth B of the cosine pulse is then between f_{min} and f_{max} (Fig. 4.7(a)):

$$f_{min} = f_c - \frac{\beta + 1}{2T_p}$$

$$f_{max} = f_c + \frac{\beta + 1}{2T_p}$$

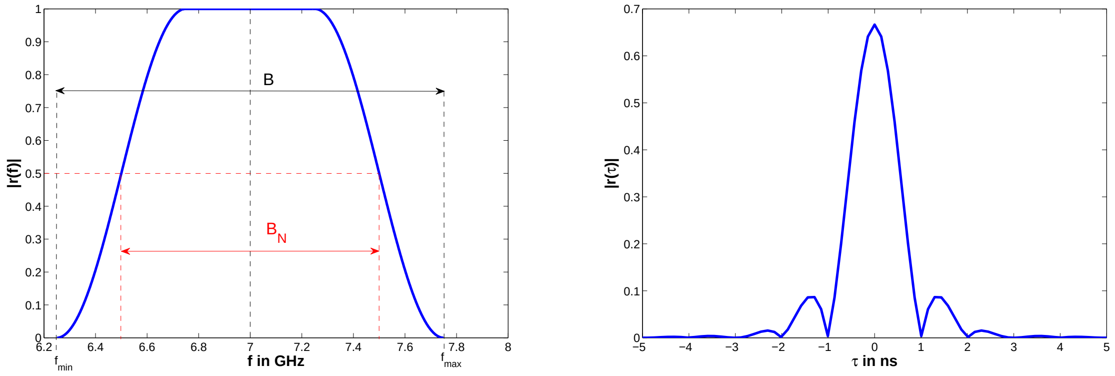
The frequency domain signal is then transformed to time domain as described in [5]. To get the time domain raised cosine pulse $r(\tau)$, a transformation matrix $\boldsymbol{P}$ is multiplied with the frequency domain pulse in a vector, called $\mathbf{r}$, which consists of N_f frequency points.

$$r(\tau) = \boldsymbol{P}^T \boldsymbol{r}$$

The transformation matrix $\boldsymbol{P}$ holds the IDFT (inverse discrete Fourier transform) coefficients and is defined as

$$\boldsymbol{P} = \left[\mathrm{e}^{j2\pi f_0 \tau} \dots \mathrm{e}^{j2\pi (f_0 + (N_f - 1)\Delta f)\tau} \right]^T$$

where Δf is the spacing of the frequency vector and f_0 is the lowest extracted frequency, i.e. $f_0 = f_c - (\beta + 1)/(2T_p)$.



(a) Raised cosine in frequency domain

(b) Raised cosine in time domain

Figure 4.7: Raised cosine in frequency and time domain with f_c =7 GHz, $\beta = 0.5$ and T_p =1 ns

Alternatively, the raised cosine pulse could be defined directly in the time domain by

$$r(\tau) = \mathrm{sinc}\left(\frac{\tau}{T}\right) \frac{\cos \frac{\pi \beta \tau}{T}}{1 - \frac{4\beta^2 \tau^2}{T^2}}$$

The measured impulse response $h(\tau)$ and the raised cosine pulse $r(\tau)$ in time domain are convolved to result in the shaped impulse response $h_{sh}(\tau)$.

$$h_{sh}(\tau) = r(\tau) e^{+2\pi f_c \tau} * h(\tau)$$

Fig. 4.8 shows the originally measured, calibrated impulse response $h(\tau)$ and the shaped impulse response $h_{\mathrm{sh}}(\tau)$.

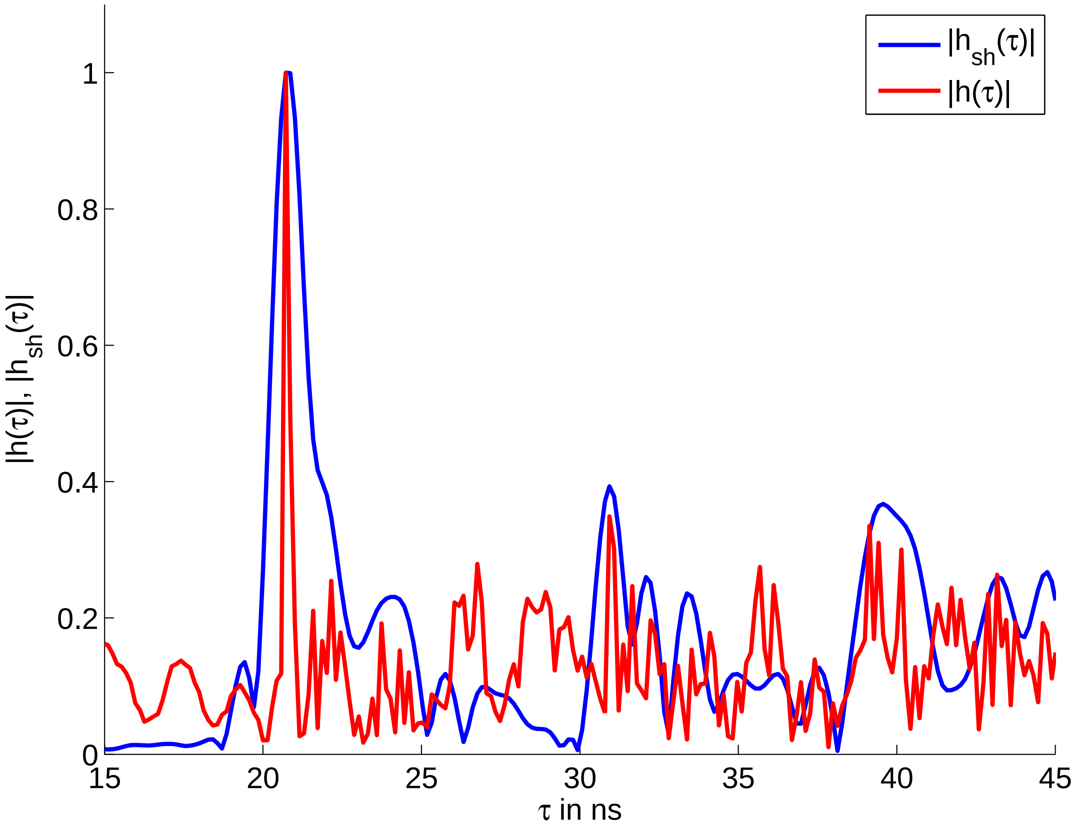


Figure 4.8: Calibrated impulse response $h(\tau)$ and shaped impulse response $h_{sh}(\tau)$ in time domain. Pulse shaping parameters: $f_c = 7$ GHz, $\beta = 0.5$ and $T_p = 1\,ns$

5 The Measurement Software

This software allows to measure and store impulse responses of UWB signals using the M-Sequence device. The main purpose is to store those impulse responses for further processing. Additionally, ranging with two receivers (base stations) is done in order to check whether the results are plausible or not.
The script is based on the demo framework delivered with the M-Sequence device. Commands are sent to device via an USB API. All necessary API calls are used in the demo framework so further documentation is usually not necessary.
The main intention of this software is to measure the impulse responses along a certain way in a room, called trajectory, which is a series of points in the two-dimensional space. Fig. 5.1 shows a part of a floor plan. The black line in the lower part of the floor plan is the trajectory, the dots on it are the measurement points. Building structure like walls and windows are plotted as thicker solid lines.

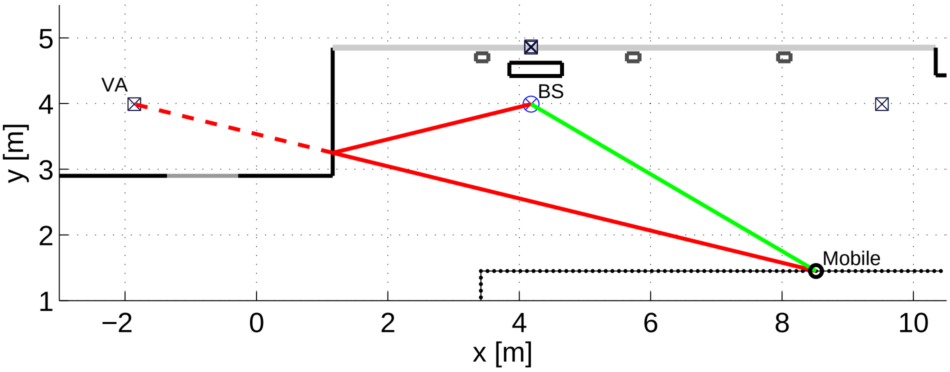


Figure 5.1: Floor plan showing VA concept

The mobile at a certain position on the trajectory is marked in the plot. From this position, the UWB signal is transmitted and received at the base station (BS). The direct signal path from the mobile to the BS and some of the reflections due to the building structure can be associated with pulses in the measured CIR. Those pulses are called the multipath components (MPCs). To each MPC a virtual base stations or virtual anchor (VA) can be associated. Such a VA is illustrated in the floor plan and denoted as VA.
The green solid line indicates the direct signal path from the transmitter to the base station (BS). The red solid line denotes the signal which is reflected at a wall and arrives at the base station later as it has a longer signal path (a higher delay) than the direct path. The red dashed line is the signal if it would not have been reflected, called the virtual path. The path length of the reflected part and the virtual path are the same. The virtual path ends in the VA. Due to the same path length, the VA coordinates are the BS coordinates mirrored at the wall at which the reflection happens. The result of this is that the VA coordinates can be computed using the BS coordinates and the floor plan.
There exist different types of VAs, depending on how many reflections are involved. In Fig. 5.1 a first order VA is shown because one signal reflection happens. A VA of second order would involve two reflections. The number of order which can be used depends on how large the room is (the larger the room the less reflections can be used due to path loss) and how strong the building structure attenuates the signal (strong attenuation means that higher order reflections

are unlikely).

5.1 Functionality of the Script

The script is started with the `main.m` file. Before starting, the parameters have to be defined as described in the next sections.
Before the measurement starts, the script tells the user where to place the receiver(s). When they are placed, the calibration measurements can be started by pressing *Enter*. They are necessary as the M-Sequence device does not allow reading out the start sample of the M-Sequence and as this start sample (which indicates the start of the measured impulse response) varies with each initialization of the device, it cannot be defined hardcoded in the script. The calibration measurements include measuring the crosstalk (described in Sec. 4.1.1) and the system response of each channel (described in Sec. 4.1.2).
The calibration measurement has to be done each time the channel sounder is started or an initialisation using the according Matlab command has been done. In case different measurements have to be done, calibration is only needed once. Therefore, it is important to keep the measured system response and the crosstalk variables in the Matlab workspace, or store them to be able to reload them. If they are deleted (e.g. by using the Matlab command *clear all*) the calibration has to be done again.
The next step is measuring the trajectory points. The script tells the user where to place the transmitter and then the measurement has to be started by pressing *Enter*. The script waits some seconds (defined by the parameter `pause_time`) to allow the user to leave the room in order to not influence the measurement. The script outputs a beep signal before the actual measurement starts (to inform the user in case he has not left the room early enough) and after the measurement is finished, to inform the user that the room can be entered and the next point can be measured.
Sometimes a measurement can take a little longer. During testing it has been observed that sometimes the impulse response does not change because an old impulse response is read out of the memory. In that case the measurement is repeated until the distance changed, but at most five times. After five repetitions the measured distance and the impulse response are accepted and considered correct. The description of the algorithm for distance measurement can be found in Sec. 5.5.2.
In case both receivers are used, the distance information can be used to compute the two possible positions of the transmitter. The transmitter position is known, so the computation of the position is just for checking whether the measured data and computed distances are plausible or not. The algorithm can be found in Sec. 5.5.3 and the helper function description in Sec. 5.7.3.
After the measurement is finished the user can plot desired impulse responses. Plotting is not done automatically as there are usually many impulse responses which are measured. The code part for plotting can be found at the very end of the script. Before plotting, the parameters have to be defined (see Sec. 5.3.5) and the code has to be executed by marking it and pressing F9. To show a plot, mark the code of the desired plot type in the script and execute the marked part by pressing F9.

5.2 Functionality of the 'Free Mode'

The 'Free Mode' allows to measure an arbitrary number of trajectory points which do not have to be pre-defined. The only difference to the normal mode is that no position estimation is done, also the stored data does not have any information on where the transmitter was placed. This mode is intended to be used when some specific positions want to be examined several times without knowing how many measurements are needed. To quit the 'Free Mode', type *quit* and press *Enter*.

5.3 Script Parameters

5.3.1 Device Parameters

The parameters of the M-Sequence channel sounder are defined in the `MDevCreateParamsFile` function. Most of the device parameters can be changed in the script, but usually it is not necessary, therefore only some parameters, which might make sense to be changed, are described here.

Parameter	**Description**
mClk	The master clock in Hz defines the sampling rate of the system.
mMLBS_Order	The order of the M-Sequence, must be 12 as the device does not support other orders.
mFPGA_BIN	The name of the FPGA design file used by the device with file ending .bin. This file must be contained in the root folder of the measurement software.
mNum_AVG	The number of total averages (hardware and software averages), a value between 1 and 1024, the manual advises to use at least 64.

Table 5.1: Device parameters

5.3.2 Measurement Parameters

This Section describes all parameters that are needed in order to use the software properly.
In the Matlab script file the parameters can be found at the top in the cell called 'Parameters for the measurement'. Tab. 5.2 describes those measurement parameters.

Parameter	**Description**
do_calibration	Set to 1 to do the calibration measurements. Set the variable to 0 if the script is started again and the calibration has already been done.
free_mode	Set to 0 to measure the defined trajectory points, set to 1 to use the free mode and measure an arbitrary number of points.
use_ch	Defines which receiver channel(s) are used, can be `1`, `2` or `[1,2]`.
geometry_path	The path to the scene file from which the receiver (base station) positions and the trajectory are read (path without the trailing file separator).
scen_file	Name of the file containing the receiver positions, trajectory and further information on the measurement campaign. For the format of the data in the scen_file see [5].
VA_path	Path to the file holding the virtual anchor data (without the trailing file separator).
VA_file	The file containing the virtual anchor information.

Parameter	**Description**
BSidxs	This vector is only used when the receiver positions are read from the scen_file. It holds the indices of the receivers and defines which one the first and which the second one is, therefore the order of the indices is important. The base stations are stored in the struct `fp.BS` of the scen_file.
tx_len	Length of the cable to the transmitter antenna in meters.
rx1_len, rx2_len	Length of the cables to the receivers in meters.
c_cable	Transmission speed in the cables in meters/second.
tx_antenna_len rx1_antenna_len rx2_antenna_len	Length of the antennas of transmitter and the two receivers in meters.
c_antenna	Transmission speed in the antennas in meters/second.
pause_time	Time in seconds the program waits when the user started a new measurement point. The pause_time allows the user to leave the room before the actual measurement starts in order to not influence the results.
folder	Path to the folder in which the measurement results are stored (without the trailing file separator).

Table 5.2: Measurement parameters

5.3.3 Additional Information on the Measurement Setup

All additional information is stored as string and has information on the measurement scenario.

Parameter	**Description**
tx_antenna_type rx1_antenna_type rx2_antenna_type	Description of the used antenna types.
location	Describes where the measurement was done.
measurement_description	Describes the purpose of the measurement and could for instance also include some remarks.
tx_cable_type rx1_cable_type rx2_cable_type	Description of the used cable types.

Table 5.3: Additional information parameters

5.3.4 Pulse Shaping Parameters

All parameteres of the cosine pulse are stored in the `md` struct. The individual parameters are:

Parameter	**Description**
md.Tp	The pulse duration in seconds. It is the inverse of the bandwidth of the pulse in frequency domain and is in most cases defined via the bandwidth.

Parameter	Description
md.beta	The roll-off factor of the raised-cosine.
md.fc	The center frequency of the cosine in Hz.
md.type	The type of the pulse, can be 'RC' (raised cosine), 'RRC' (root raised cosine) or 'RC2' (raised cosine squared).
Frequency vector	The second parameter of the `rcos_win` function is a frequency vector. The calculated bandwidth of the cosine pulse from $f_{l_{rcos}} = f_c - (1+\beta)/(2T_p)$ to $f_{u_{rcos}} = f_c + (1+\beta)/(2T_p)$ must be inside the given frequency vector. As the M-Sequence device measures the signal in time domain a frequency vector has to be defined. The smallest frequency step is $\Delta f = \frac{1}{\tau_{max}}$ where $\tau_{max} = \frac{IRF\_len}{f_s}$. IRF_len is the length of the M-Sequence and f_s is the sampling clock of the M-Sequence device. With the frequency step, a lower frequency f_l and upper frequency f_u (both defined by the user) the frequency vector can be defined in MatLab as $$f = f_l : \Delta f : f_u$$

Table 5.4: Pulse shaping Parameters

5.3.5 Plotting Parameters

Those parameters are to control the plotting of impulse responses.

Parameter	Description
meas_point	Defines which measurement point is plotted. It holds the according index of a point in the m_pos vector.
ch	Defines which channel to plot, it can be 1 or 2.
va_order	The highest order of virtual anchor data to plot.
x_type	Defines which x- axis to use for plotting. Possible values are: `'x_n'` samples `'x_tau'` tau - time delay in ns `'x_dist'` distance in meter

Table 5.5: Plotting Parameters

5.4 Format of the Stored Data

The script stores general information on the M-Sequence and measurement and the pulse shaping parameters. The stored impulse responses are the raw, the calibrated and the shaped ones. The file name is the current date time string and the file is stored in the folder defined in the parameters.

5.4.1 Impulse Responses

Different types of impulse responses are stored during measurement. All impulse responses have the same format: A three-dimensional matrix containing the measured complex impulse response. The first dimension m is the measurement number, i.e. the m-th measurement point of the position (trajectory) vector. The second dimension n is the complex impulse response, typically with 4095 points (using 12 bit M-Sequence). The third dimension c defines is the channel number, can be 1 or 2.
The stored impulse responses are:

Variable	**Description**
CIR_raw	The CIR measured by the device without calibration
CIR_calibrated	The calibrated CIR (also includes distance correction)
CIR_shaped	The pulse shaped CIR (also includes distance correction)

Table 5.6: Names of the stored impulse responses

5.4.2 Measurement Device Parameters (mdev)

The struct `mdev` holds all the device parameters and created variables. It also holds the measured crosstalk and system impulse responses. They are needed if the calibrated impulse response has to be computed from the measured raw impulse response.

Variable	**Description**
mdev.tau	The tau vector used for plotting and computation of delays.
mdev.mMLBS_Order	The order of the M-Sequence.
mdev.numSamples	The number of samples per impulse response, can also be computed by $2^{mMLBS\_Order} - 1$.
mdev.mClk	The master clock or sampling rate of the M-Sequence device in Hz. Default value is 6.95 GHz.
mdev.mNum_Ch	Number of receivers, usually there are two receivers.
mdev.mNum_Avg	The total number of averages, can have integer values between 1 and 1024, but usually it should be at least 16.
mdev.r_sys	The measured system impulse response for each channel.
mdev.r_cross	The measured crosstalk impulse response for each channel.
mdev.p	A struct holding the information for the calibration process, e.g. the antenna lengths. Important fields of this struct are: • th: A threshold for the zero-forcing equalization (explained in Section 5.5.1), defined as fraction of the signal maximum. • truncate_sys: If set to 1, all samples outside of an automatically defined, fix-sized window are set to zero. The window is defined to contain the actual system impulse response. This can be used if the system impulse response was measured without an attenuator to reduce the influence of the peaks caused by nonlinear effects. If set to 0, the whole system response is used for the calibration.

Table 5.7: Description of the 'mdev' struct

5.4.3 Pulse Shaping (shaping)

Holds information on the pulse shaping.

Variable	**Description**
shaping.f_meas	Holds the frequency vector as returned by the `rcos_win` function.
shaping.md	A struct holding the pulse shaping parameters as described in Section 5.3.4.

Table 5.8: Description of the 'shaping' struct

5.4.4 Measurement Setup Description (description)

Holds a description of the measurement setup.

Variable	**Description**
rx	The position and the height of the used base stations.
pos_vec	A vector holding all positions on which measurements were done.
tx_len	Length of the cable to the transmitter in meters.
rx1_len rx2_len	Length of the two receiver cables in meters.
c_cable	The transmission speed in the cables for delay computation in meters / second.
tx_antenna_len	The length of the transmitter antenna in meters.
rx1_antenna_len rx2_antenna_len	The length of both receiver antennas in meters.
c_antenna	The transmission speed in the antenna for delay computation in meters / second.
scen_file	The name of the used scene file.
VA_file	The name of the used virtual anchor file.
tx_antenna_type rx1_antenna_type rx2_antenna_type	Description of the used antenna types.
location	Describes where the measurement was done.
measurement_description	Describes the purpose of the measurement and could for instance also include some remarks.
tx_cable_type rx1_cable_type rx2_cable_type	Description of the used cable types.
date	Date of the measurement.

Table 5.9: Description of the 'description' struct

5.4.5 Positioning Information (positioning)

This struct stores all the computed distance and position information (if two base stations are used).

Variable	Description
tx_all	Holds all computed (possible) positions of the transmitter. There are two possible positions per measurement as only the distance to the two base stations is used.
dist_all	The computed distance to each base station for each trajectory point.

Table 5.10: Description of the 'positioning' struct

5.5 Algorithms

5.5.1 Computing the Channel Impulse Response

To compute the channel impulse response, the crosstalk and the system response have to be measured first. This is described in Sec. 4.1.1 and Sec. 4.1.2, respectively.

The crosstalk is subtracted from the measured channel impulse response and from the system response. Fig. 5.2 shows examples of the raw measured channel impulse response $h_{\mathrm{ch,meas}}(\tau)$, the crosstalk $h_{\mathrm{cross}}(\tau)$ and the system impulse response $h_{\mathrm{sys,meas}}(\tau)$.

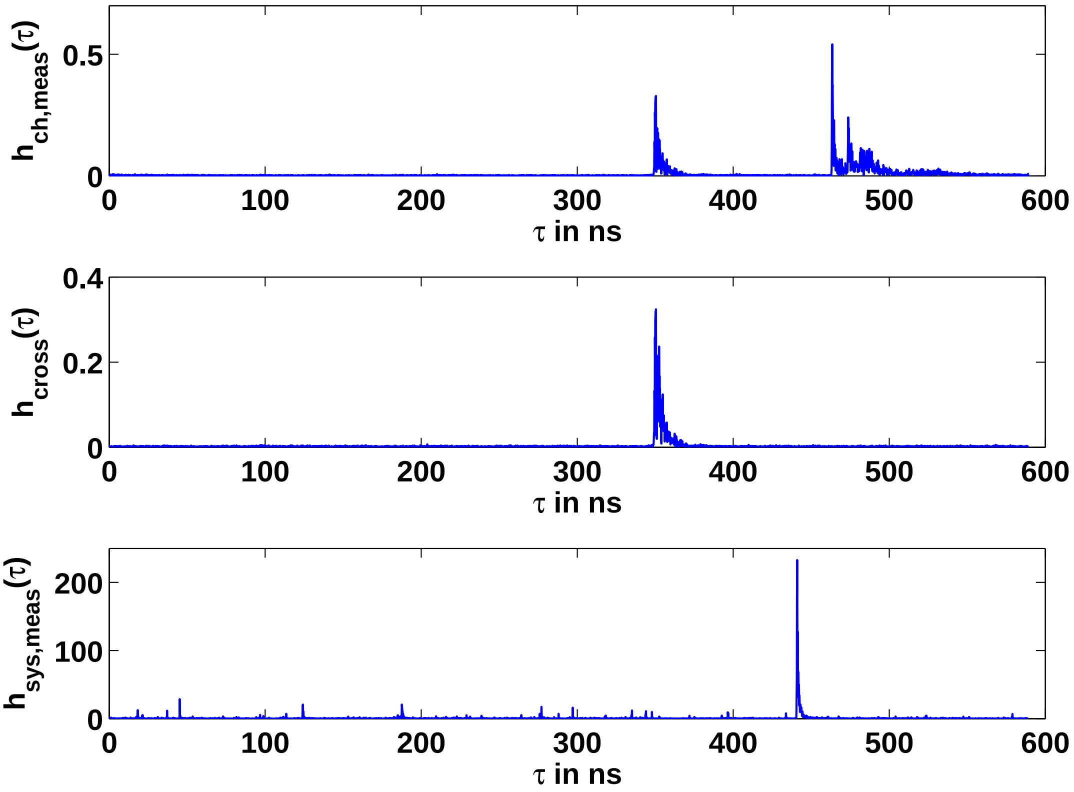


Figure 5.2: Examples of measured impulse responses

The small peaks in the lower plot (system response, $h_{\mathrm{ch,meas}}(\tau)$) stem from the measurement of the system response without an attenuator. Due to that, nonlinear distortions occur in the system and result in those peaks. By using an attenuator those peaks can be suppressed.

To compute the channel impulse response $h_{\mathrm{ch}}(\tau + \tau_{\mathrm{add}})$ use:

$$h_{\mathrm{ch}}(\tau+\tau_{\mathrm{add}}) = \mathrm{IDFT}\left[\frac{H_{\mathrm{ch,meas,corr}}(f)}{H_{\mathrm{sys,meas,corr}}(f)}\right]$$

where $H_{\mathrm{ch,meas,corr}}(f)$ is the measured channel impulse response with subtracted crosstalk and $H_{\mathrm{sys,meas,corr}}(f)$ is the system response with subtracted crosstalk, both in frequency domain. For the system response, samples below a certain defined threshold (defined by the user, see `mdev.p` in Tab. 5.7), which depends on the signal maximum, are set to zero in the time domain before the Fourier Transformation. This is a form of a Zero Forcing Equalizer and is used to reduce the noise in the signal as noise amplification occurs due to the division in the frequency domain. The division in frequency domain corresponds to a deconvolution in time domain. As the M-Sequence is a cyclic code and its generator does not necessarily start at the first sample, the start of the impulse response usually is at some arbitrary delay, called code delay. In order to use the impulse response to get distance information the starting delay must be known. The M-Sequence device does not allow reading that start sample value of the generator from the register and also does not shift the measured impulse response automatically.

The system response and the measured channel impulse response both have this unknown shift due to the code delay. Due to the division, this delay cancels out and the only delay remaining in the result is caused by the antennas, which were not considered in the system response measurements, and the connector used in those measurements. This delay is denoted as τ_{add} in $h_{\mathrm{ch}}(\tau+\tau_{add})$.

To account for the additional delay τ_{add}, shift the impulse response $h_{\mathrm{ch}}(\tau+\tau_{\mathrm{add}})$ to get $h_{\mathrm{ch}}(\tau)$. The shift is done in the sample domain, so the antenna and the connector lengths have to be converted to number of samples using:

$$\tau_{\mathrm{add,samples}} = \mathrm{round}\left((len_{\mathrm{tx}} + len_{\mathrm{rx}} + len_{\mathrm{conn}}) \cdot \frac{mClk}{c_{\mathrm{prop}}}\right)$$

This assumes that antennas and connector have the same propagation speed c_{prop}. $mClk$ is the sampling frequency of the channel sounder and len_{tx}, len_{rx} and len_{conn} are the lengths of the transmit antenna, receive antenna and the connector, respectively.

After this shift, the channel impulse response is positioned correctly and it is possible to use the delays starting at zero for distance measurements.

Fig. 5.3 shows the effect of the shift by the additional delay τ_{add}. The upper plot shows the correct signal, which is shifted by τ_{add}. In the lower plot, the signal without the shift is shown. Omitting the shift in this case results in very inaccurate results as each of the used antennas was 0.17 m long. This results in a shift of

$$\tau_{\mathrm{add}} = \frac{0.17+0.17}{0.7\cdot c} = 1.6\,\mathrm{ns}$$

where $0.7 \cdot c$ is the propagation speed in the antennas expressed as a fraction of the speed of light. In the spatial domain, this shift equals $1.6\,\mathrm{ns} \cdot c = 0.48\,\mathrm{m}$. A signal with a systematic error of 0.48 m would make it unsuitable for any positioning or tracking algorithm.

The dashed lines in Fig. 5.3 indicate expected multipath components.

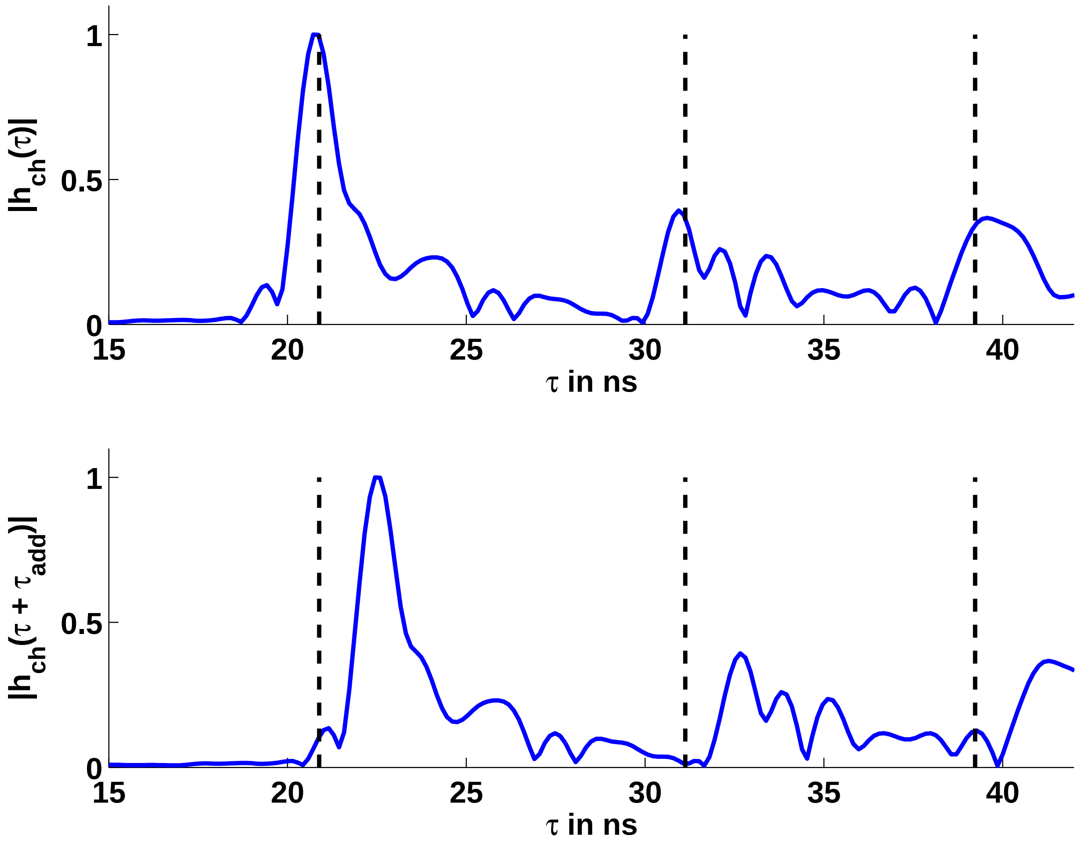


Figure 5.3: Computed channel impulse response: shifted (upper plot, $h_{\mathrm{ch}}(\tau)$) and not shifted (lower plot, $h_{\mathrm{ch}}(\tau + \tau_{\mathrm{add}})$) by the additional delay τ_{add}

5.5.2 Distance Measurement

To compute the distance from the transmitter to a base station the calibrated impulse response is used. The first step is to find the first arriving component of the signal via the direct path. A very popular algorithm for such operations is the Jump-Back-Search-Forward (JBSF) algorithm as described in [13]. JBSF is a threshold based algorithm and its functionality can be seen in Fig. 5.4. The algorithm assumes that the receiver is synchronized to the strongest path. From the strongest path at sample n_{max} the algorithm jumps back by $N_w^{(sb)}$ samples (the window size) to sample n_{sb}. From this sample the algorithm iterates towards the strongest component and compares each sample with the defined threshold ξ. The first sample n_{le} being greater than ξ is returned as the so called leading edge sample. Mathematically, the algorithm can be described in the following way:

$$\hat{n}_{JBSP} = min\{n|h[n] > \xi\}$$

where $n \in \{n_{sb}, n_{sb} + 1, ..., n_{max}\}$ and $h[n]$ is the impulse response in the sample domain.
The first arriving component is not necessarily the strongest one (e.g. in a NLOS scenario) therefore JBSF cannot be used here. To find the first arriving component an algorithm using a threshold value and clustering is used. The threshold defines which sample is part of a signal component (greater than or equal the threshold) and which is noise (less than the threshold). A cluster consists of at least three contiguous samples which are greater than the threshold. The algorithm assumes that the first cluster found includes the first signal component. The maximum peak of the cluster is considered the first arriving component. This sample is used to

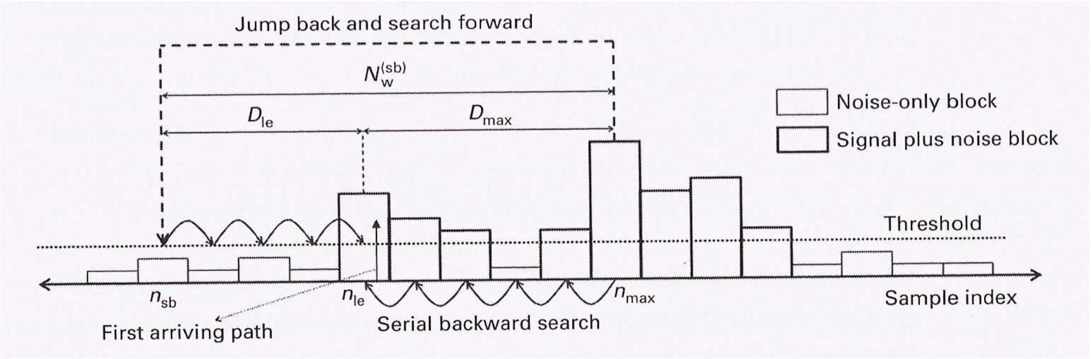


Figure 5.4: Illustration of the JBSF algorithm [13]

compute the delay and with that the distance between receiver and transmitter.

5.5.3 Transmitter Position Estimation

The distances from the transmitter to both receivers are computed right after the measurement and based on those distances the transmitter position is computed. This is done in order to check whether the measured impulse responses of the measurement point are plausible or not. During implementation of the script it sometimes happened that the M-Sequence device delivered impulse responses which resulted in the same computed distance as in the previous measurement point although the transmitter was moved more than half a meter, therefore the computed distance is also used to check if the measured impulse response is not an old one.

The estimated position (x_{TX}/y_{TX}) of the transmitter is computed in the following way: For both receivers RX1 (at position (x_{RX1}/y_{RX1}) and RX2 (at position (x_{RX2}/y_{RX2}) a circle with radius of the computed distance and the receiver position as the center can be drawn which results in two circle equations.

$$d_1^2 = (x_{TX} - x_{RX1})^2 + (y_{TX} - y_{RX1})^2$$

$$d_2^2 = (x_{TX} - x_{RX2})^2 + (y_{TX} - y_{RX2})^2$$

The circles have the two intersection points. By subtracting both circle equations from each other the quadratic terms of the transmitter position cancel out and a line equation is left over.

$$y_{TX} = \frac{d_1^2 - d_2^2 + 2x_{TX}x_{RX1} - x_{RX1}^2 - y_{RX1}^2 - 2x_{TX}x_{RX2} + x_{RX2}^2 + y_{RX2}^2}{-2y_{RX1} + 2y_{RX2}}$$

The parameters of the line equation $y_{TX} = kx_{TX} + d$ can be computed using

$$k = \frac{2x_{RX1} - 2x_{RX2}}{2y_{RX2} - 2y_{RX1}}$$

$$d = \frac{d_1^2 - d_2^2 - x_{RX1}^2 - y_{RX1}^2 + x_{RX2}^2 + y_{RX2}^2}{2y_{RX2} - 2y_{RX1}}$$

The line equation can be inserted in one of the circle equations (here shown for equation of RX1) which results in a quadratic equation with only the x-coordinate of the transmitter as an unknown variable.

$$0 = x_{TX}^2(1 + k^2) + x_{TX}(-2x_{RX1} + 2kd - 2ky_{RX1}) + (x_{RX1}^2 + d^2 - 2dy_{RX1} + y_{RX1}^2 - d_1^2)$$

Solving the quadratic equation and inserting the two resulting x-values into the line equation results in two coordinate pairs. For each coordinate pair the mean squared error with the known transmitter position is computed, the pair with the smaller error is the computed transmitter position. The result is presented to the user.

Fig. 5.5 shows all geometrical shapes for computing the two possible transmitter positions as described above.

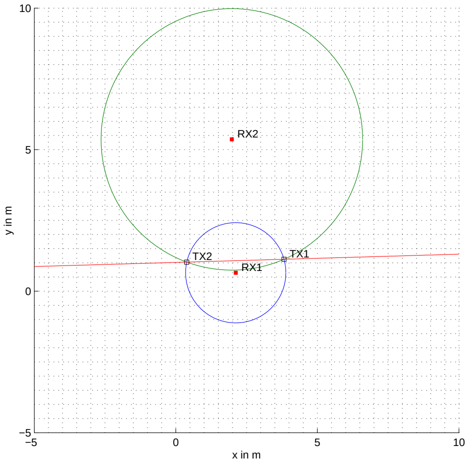


Figure 5.5: Computing the two possible transmitter positions

5.6 Functions for Using the M-Sequence Device

5.6.1 MDevInitAndCalibration

Description

Comprises all functions needed for the initialisation and calibration process (`MDevCreateParamsFile`, `MDevSetup`, `MDevInit`, `MDevDoCalibration`). The parameters will be stored in a `.mat` file.

Parameters

Parameter	Description
mdev_params_file	The name of the file in which the parameters are stored.
calibration_params	A struct containing some variables needed for the computations to get the calibrated impulse response. The variables are: • `c` The speed of light • `tx_antenna_len` The length of the transmitter antenna • `rx_antenna_len` The length of the receiver antenna (same length for all base stations) • `additional_len` Accounts for lengths by which the impulse response might be shifted (e.g. the connector used for the system response measurement) • `c_antenna` The speed of propagation in the antennas • `calibration_th` The threshold for setting samples of the system response to zero, calibration_th is a fraction of the system response maximum (e.g. 0.01)

Table 5.11: Parameters of the `MDevInitAndCalibration`- function

Return Values

Variable	Description
s	A struct holding all defined parameters, computed variables and measured system responses and crosstalk.

Table 5.12: Return values of the `MDevInitAndCalibration`- function

5.6.2 MDevCreateParamsFile

Description

The measurement parameters of the channel sounder are defined in this function. It creates the file in which all parameters are stored and the variables and calibration measurements will be added later.

Parameters

Parameter	Description
filename	The name of the file in which the parameters will be stored.

Table 5.13: Parameters of the `MDevCreateParamsFile`- function

Return Values

None

5.6.3 MDevSetup

Description

Computes all variables from the given parameters needed for operating the channel sounder.

Parameters

Parameter	**Description**
params_file	The name of the file in which the parameters are stored.

Table 5.14: Parameters of the `MDevSetup`- function

Return Values

Variable	**Description**
s	A struct holding all defined parameters, computed variables and measured system responses and crosstalk.

Table 5.15: Return values of the `MDevSetup`- function

5.6.4 MDevInit

Description

Initialises the channel sounder.

Parameters

Parameter	**Description**
s	The struct holding all parameters and variables for operating the channel sounder.

Table 5.16: Parameters of the `MDevInit`- function

Return Values

None

5.6.5 MDevDoCalibration

Description

Performs the calibration measurements. Prompts the user to measure crosstalk and system response.

Parameters

Parameter	Description
s	The struct holding all parameters and variables to operate the channel sounder.

Table 5.17: Parameters of the `MDevDoCalibration`- function

Return Values

Variable	Description
s	The struct given as parameter with the crosstalk and system response added.

Table 5.18: Return values of the `MDevDoCalibration`- function

5.6.6 MDevGetMeasurement

Description

Performs a measurement.

Parameters

Parameter	Description
s	The struct holding all parameters and variables to operate the channel sounder and the measured crosstalk and system response from the calibration measurement.
do_calibrate	Use crosstalk and system response to compute the calibrated impulse response.

Table 5.19: Parameters of the `MDevGetMeasurement`- function

Return Values

Variable	Description
CIR_orig	The raw measured impulse response.
CIR_calibrated	The calibrated impulse response computed from the raw impulse response.

Table 5.20: Return values of the `MDevGetMeasurement`- function

5.6.7 MDevDeinit

Description

Deinitialise the channel sounder. Stops the measurements and closes the API. After this function is called, a new calibration measurement is necessary if further measurements should be done.

Parameters

None

Return Values

None

5.7 Description of the Helper Functions

5.7.1 do_pulse_shaping

Description

Performs pulse shaping as theoretically described in Sec. 4.2.
The implementation uses the `rcos_win` and `CIR_IDFT` functions as described in [5]. To generate the raised cosine pulse, `rcos_win` is used. It takes the raised cosine parameters (center frequency f_c, roll-off factor β and the pulse duration T_p), and a frequency vector (see Sec. 5.3.4). For the frequency vector, the following conditions must hold:

$$f_{min} = f_c - \frac{\beta + 1}{2T_p} \geq f_{meas,lower}$$

$$f_{max} = f_c + \frac{\beta + 1}{2T_p} \leq f_{meas,upper}$$

where the lower and upper bound of the frequency vector are $f_{meas,lower}$ and $f_{meas,upper}$, respectively.
The return values are a vector holding the raised cosine pulse samples and the according frequency vector f_{meas} which is bounded by f_{min} and f_{max}.
This raised cosine pulse in frequency domain $P(f)$ is then shifted to be centred around zero:

$$P(f) = P(f) \cdot \mathrm{e}^{-j2\pi f \tau_{med}}$$

where τ_{med} is the median value of the τ- vector and is always a single value, as τ always has odd length. Without the shift, the time domain signal would look like a so-called bathtub signal. The transformation to time domain is done using the `CIR_IDFT` function.
The measured impulse response and the raised cosine pulse in time domain are convolved. As the result of the convolution exceeds the original length (e.g. impulse response length is 4095, raised cosine also has 4095 samples, length of convolution is $4095 + 4095 - 1 = 8189$) leading and trailing samples are dismissed to obtain an impulse response function of the original length again.

Parameters

Parameter	**Description**
ir	The impulse response which is convolved with the cosine pulse.
f_s	The sampling frequency (master clock) of the M-Sequence device, needed to create the τ vector and to determine τ_{max} for the frequency vector.
f_lower	The lower bound of the frequency vector.
f_upper	The upper bound of the frequency vector.
md	A struct containing the parameters of the cosine pulse (described in Sec. 5.3.4).

Table 5.21: Parameters of the `do_pulse_shaping`- function

Return Values

Variable	**Description**
shaped_ir	The shaped impulse response.
f_meas	The frequency vector returned by `rcos_win`.

Table 5.22: Return values of the `do_pulse_shaping`- function

5.7.2 compute_delay_from_zero

Description

Computes the time delay of the first incoming signal part. In order to work correct this function needs the calibrated impulse response.
The algorithm iterates over the samples of the impulse response and tries to find clusters of contiguous samples exceeding a certain threshold. If the cluster size is three or more a valid cluster has been found and the first sample of the cluster is considered the start of the signal. As only the first cluster is needed the iteration over the samples is stopped after that.
The delay is the value of the tau vector at the start of signal sample, the distance is the delay times the speed of light.

Parameters

Parameter	Description
ir	Vector with absolute values of the shifted impulse response.
tau	The tau vector of the measured impulse response.

Table 5.23: Parameters of the `compute_delay_from_zero`- function

Return Values

Variable	Description
delay	The computed delay in seconds.
distance	The computed distance in meters.

Table 5.24: Return values of the `compute_delay_from_zero`- function

5.7.3 compute_position

Description

This function computes the position of the transmitter for the given receiver positions and distances between them and the transmitter as described in Sec. 5.7.3. An additional function called `get_line_parameters` is used to get the parameters of the straight line equation.

Parameters

Parameter	Description
rx	A matrix holding x and y values of the receivers, x in first row, y in second row. $rx = \begin{pmatrix} x_{R1} & x_{R2} \\ y_{R1} & y_{R2} \end{pmatrix}$
dist	A vector holding the distances of each receiver to the transmitter. $dist = \begin{pmatrix} dist_{R1} & dist_{R2} \end{pmatrix}$

Table 5.25: Parameters of the `compute_position`- function

Return Values

Variable	Description
tx	A matrix holding the two possible transmitter positions, x values in the first row, y values in the second row. $tx = \begin{pmatrix} x_1 & x_2 \\ y_1 & y_2 \end{pmatrix}$

Table 5.26: Return values of the `compute_position`- function

5.7.4 get_line_parameters

Description

Computes the parameters k and d for the straight line equation when subtracting the two circle equations of the receivers.

Parameters

Parameter	Description
rx	A matrix holding x and y values of the receivers, x in first row, y in second row. $$rx = \begin{pmatrix} x_{R1} & x_{R2} \\ y_{R1} & y_{R2} \end{pmatrix}$$
dist	A vector holding the distances of each receiver to the transmitter. $$dist = \begin{pmatrix} dist_{R1} & dist_{R2} \end{pmatrix}$$

Table 5.27: Parameters of the `get_line_parameters`- function

Return Values

Variable	Description
k	The slope of the straight line equation.
d	The distance of the ordinate of the straight line equation.

Table 5.28: Return values of the `get_line_parameters`- function

5.7.5 plot_position

Description

Plots the receiver circles, the two possible transmitter positions, and the line for computing them in the 2D space for two given receivers.

Parameters

Parameter	Description
tx	A matrix holding the possible transmitter positions. Optional just one TX position ca be plotted, in that case *tx* must be a column vector. It holds the x values in the first row and y values in the second row. $$tx = \begin{pmatrix} x_{TX1} & x_{TX2} \\ y_{TX1} & y_{TX2} \end{pmatrix}$$
rx	A matrix holding x and y values of the receivers, x in first row, y in second row. $$rx = \begin{pmatrix} x_{R1} & x_{R2} \\ y_{R1} & y_{R2} \end{pmatrix}$$
dist	A vector holding the distances of each receiver to the transmitter. $$dist = \begin{pmatrix} dist_{R1} & dist_{R2} \end{pmatrix}$$

Table 5.29: Parameters of the `plot_position`- function

Return Values

None

6 Comparison to Results of the Vector Network Analyzer

In this section the results of some measurements in the seminar room of the SPSC lab are compared to the results of the measurements with the vector network analyzer (VNA) as performed in [5]. The pulse shaping of both methods were done with the same parameters which are a center frequency $f_c = 7\,\text{GHz}$, pulse duration $T_p = 1\,\text{GHz}$, roll-off factor $\beta = 0.5$ and with the same antenna height of $h_a = 1.3\,\text{m}$. The absolute values of the impulse responses of both devices are normalized to have a maximum amplitude of one to make them comparable.

The amplitudes of the peaks may differ because of the used coin antennas. They do not have a completely isotropic antenna pattern and therefore the direction of the antenna matters. It might happen that for the same measurements the peaks differ if the antennas are oriented differently.

Fig. 6.1 shows a part of the floor plan of the seminar room where the measurements were done. The plan shows three different transmitter positions, indicated as red stars, the base station (BS2) and two first order VAs for BS2 (VA #1182 and VA #1214). Those VAs were chosen as they are both visible over a long part of the trajectory. In the floor plan, the different signal paths for transmitter position two are plotted. They are plotted in different colors which match the colors in the impulse response plots in Fig. 6.2 to 6.4. The solid lines depict the physical signal paths with reflections and the dashed lines show the paths to the corresponding VA.

Fig. 6.2 to 6.4 show the absolute values of the shaped impulse responses $h(\tau)$ for both VNA and M-Sequence device. Additionally, there are lines indicating the expected delays of BS2 and the two VAs. The red line indicates the expected delay of the line-of-sight component and the other two lines the expected delay of the corresponding VA.

In the first plot (Fig. 6.2) the transmitter is at position (3/2.65). All expected delays match the pulses of the CIR quite well for both measurement devices. In this plot the difference of the amplitudes, when comparing VNA and M-Sequence, due to the antenna direction can be seen.

The second plot (Fig. 6.3) shows the transmitter at position (3/4). As the transmitter moves away from the receiver the delay increases which can be seen in the delay of the LOS component. The delays of the VAs do change a little. What can be observed is that VA #1214 has the highest amplitude in the M-Sequence device measurement although it has a three times larger delay compared to the LOS component and a reflection, which also attenuates the signal, happens. This can be due to the effect of the antennas as described beforehand.

At the position (5/4) in Fig. 6.4 the transmitter is again closer to the receiver and therefore the delay is smaller compared to position two. The amplitudes for VA #1214 are very small because it now has a longer delay and also due to the antenna influences. In the VNA measurement the pulse of VA #1182 is not clearly distinguishable.

Overall the results of VNA and M-Sequence are quite similar, most of the peaks matches quite well despite the above described circumstances (antenna patterns not isotropic).

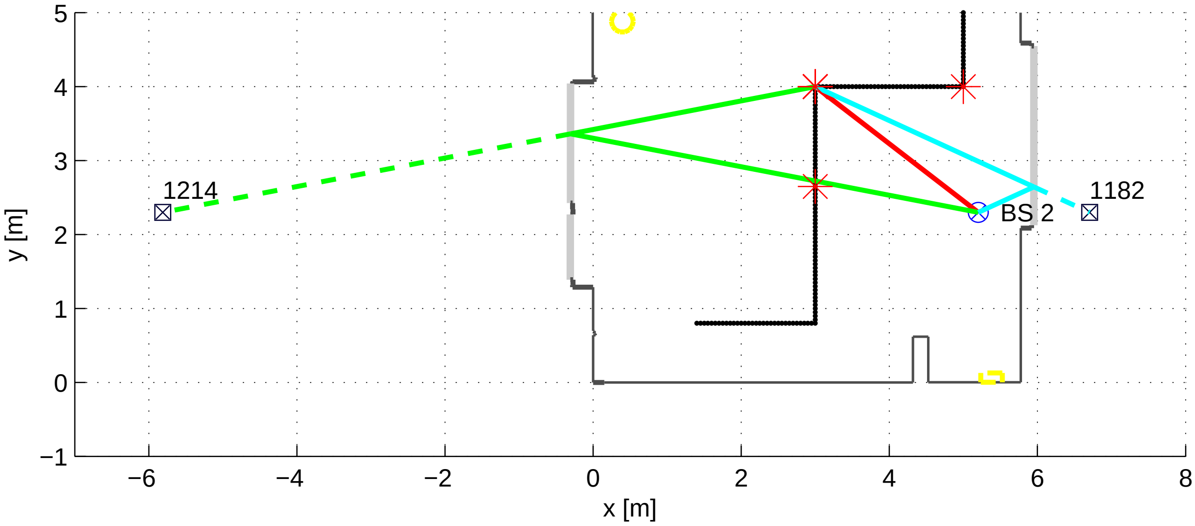


Figure 6.1: Floor plan of the HF lab with transmitter, receivers, VAs and their signal paths

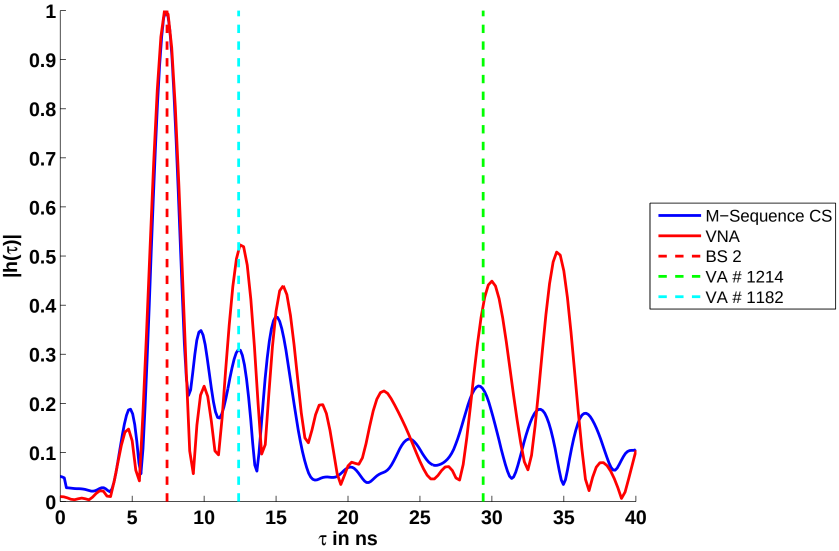


Figure 6.2: Shaped impulse responses of VNA and M-Sequence with TX at position (3/2.65)*. Pulse shaping parameters:* $f_c = 7$ *GHz,* $T_p = 1\,ns$*,* $\beta = 0.5$

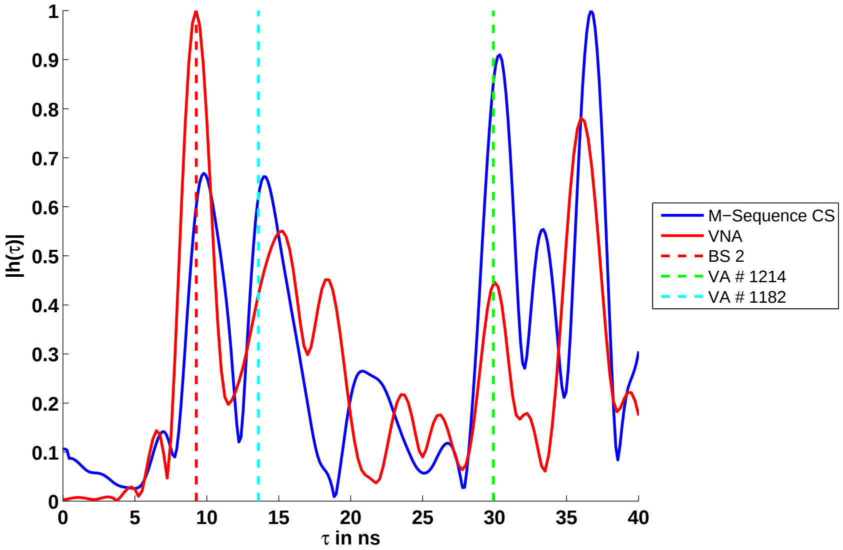


Figure 6.3: Shaped impulse responses of VNA and M-Sequence with TX at position (3/4). Pulse shaping parameters: $f_c = 7$ *GHz,* $T_p = 1$ *ns,* $\beta = 0.5$

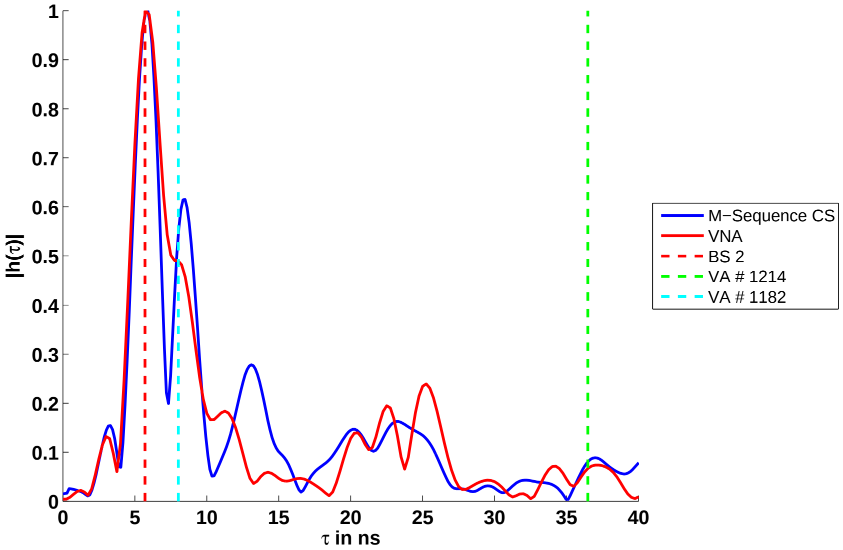


Figure 6.4: Shaped impulse responses of VNA and M-Sequence with TX at position (5/4). Pulse shaping parameters: $f_c = 7$ *GHz,* $T_p = 1$ *ns,* $\beta = 0.5$

Bibliography

[1] A. F. Molisch, "Ultrawideband propagation channels-theory, measurement, and modeling," *Vehicular Technology, IEEE Transactions on*, vol. 54, no. 5, pp. 1528 – 1545, 9 2005.

[2] ——, "Ultra-wide-band propagation channels," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 353 –371, 2 2009.

[3] B. Alavi, N. Alsindi, and K. Pahlavan, "Uwb channel measurements for accurate indoor localization," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, Oct., pp. 1–7.

[4] D. Cassioli and A. Durantini, "A time-domain propagation model of the uwb indoor channel in the fcc-compliant band 3.6 - 6 ghz based on pn-sequence channel measurements," in *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, vol. 1, May, pp. 213–217 Vol.1.

[5] P. Meissner, *UWB Channel Measurement Campaigns*, 9 2012. [Online]. Available: www.spsc.tugraz.at/tools/UWBmeasurements

[6] A. F. Molisch, *Wireless Communications*, 2nd ed. John Wiley & Sons Ltd., 2011.

[7] J. Sachs, R. Herrmann, M. Kmec, M. Helbig, and K. Schilling, "Recent advances and applications of m-sequence based ultra-wideband sensors," in *Ultra-Wideband, 2007. ICUWB 2007. IEEE International Conference on*, 9 2007, pp. 50 –55.

[8] D. D. Rife and J. Vanderkooy, "Transfer-function measurement with maximum- length sequences," in *Journal Of The Audio Engineering Sociery Audio/Acoustics/Applications*, vol. 37, no. 6, 6 1989, pp. 419–444.

[9] M. M. Rišková, M. J. Rovňáková, and I. M. Aftanas, "M-sequence uwb radar architecture for throughwall detection and localisation," pp. 29–30, 2007. [Online]. Available: http://www.aftanas.sk/aftanas/publications/DK2007.pdf

[10] TU Ilmenau Service GmbH, *Hardware Manual - Ultra Wideband M-Sequence Device*, 8 2012.

[11] ——, *Manual MatLab Software for Ultra Wideband M-Sequence Devices*, 8 2012.

[12] R. Cepeda, S. C. J. Parker, and M. Beach, "The measurement of frequency dependent path loss in residential los environments using time domain uwb channel sounding," in *Ultra-Wideband, 2007. ICUWB 2007. IEEE International Conference on*, 2007, pp. 328–333.

[13] Z. Sahinoglu, S. Gezici, and I. Guvenc, *Ultra- wideband Positioning Systems.* Cambridge University Press, 2008.

[14] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99 –110, 6 2006.