Wireless Wave Propagation in City Street Canyons

ANDRÉS FELIPE GARCIA ALBARRACIN
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Wireless Wave Propagation in City Street Canyons

Andrés Felipe García Albarracín
wir15aga@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor: Fredrik Tufvesson, Thomas Zemen

Examiner: Fredrik Rusek

June 20, 2017

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

This project aims to determine the characteristics of a wireless communication channel in city-street-canyon scenarios. The channel of a typical city intersection is modeled with a ray tracer and measured with a channel sounder. Then, the results of both tools are compared in terms of the local scattering function.

The channel sounder utilized for the project was initially developed by Rui Wang et al. at the University of Southern California in a software-defined radio (SDR). It has been updated from measuring 15 MHz to 50 MHz, which required the usage of equipment with larger bandwidth and hence, the improvement of the program running on the SDR FPGA. With the updated version of the channel sounder, a measurement campaign was run in an intersection of the city of Vienna.

Correspondingly, the ray tracer was developed by Mingming Gan during her Ph.D. thesis and later updated by Xuhong Li during her master thesis. The same intersection used to take measurements with the channel sounder has been modeled and simulated with the ray tracer. Then, the predictions of the ray tracer have been validated with the channel-sounder results. The comparison between both tools is used to provide a first verdict of whether the ray tracer can be utilized for wireless positioning.

To my parents, María del Pilar and Edgar Armando.

Popular Science Summary

At the same time that unmanned aerial vehicles (UAV), e.g. drones, increase in popularity, some concerns about their usage in illegal activities appear in government security agendas. Undesired cases, such as attacks in political gatherings or drug dealing, motivate an increasing interest in determining the position of eventual hostile UAVs. In principle, the location of an UAV can be found based on the signal between the UAV and its remote control. This problem, usually known as wireless positioning, has been already covered by several methods in the literature. The efficiency of the existing solutions could be significantly improved with the usage of geometric-based models, as for example ray tracing. In this project, we study the accuracy of a ray tracer when predicting the characteristics of the wireless communication channel that would exist between an UAV and its remote control. Our studies could be latter used to determine whether finding the position of the UAV can rely on ray tracing predictions.

Ray tracing is a deterministic channel model where the propagation of an electromagnetic wave is represented by multiple rays. Propagation mechanisms such as reflection, diffraction, diffuse scattering, penetration and line of sight are considered in a ray tracer to generate those rays. For this project, we have used the ray tracer developed by Mingming Gan, to simulate a wireless channel in a street intersection of the city of Vienna. In total, three sets of simulations were carried out, each of them representing a possible path that an UAV would take when flying at low altitude.

Simultaneously, we have run a measurement campaign in the same scenario simulated with the ray tracer. As a measurement device, we developed a channel sounder based on the original design of Rui Wang, et al. at the University of Southern California. This channel sounder consists of a transmitter that periodically sends a sounding sequence, and a receiver that collects the corresponding response of the channel. Both devices are synchronized with atomic clocks of rubidium that are disciplined by the GPS satellite system. The sounder is implemented on a software-defined radio that is a flexible device where the radio characteristics of the system can be programmed. We have upgraded the capabilities of the original channel sounder from measuring 15 MHz to 50 MHz, which required the improvement of the original device program.

The measurements taken with the channel sounder are then utilized to validate the predictions of the ray tracer. We have established a comparison between the results of both tools in terms of the local scattering function and some derived expressions. The local scattering function provides the distribution of the channel gain over the signal delay and its Doppler frequency shift. Although some of the details of the local scattering are correctly predicted by the ray tracer, some characteristics are misrepresented, especially in the gain distribution over the Doppler shift. Those inaccuracies suggest that some models of the ray tracer should be revised in order to use it as a wireless positioning tool.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Already since the first half of the twentieth century, early developments of unmanned aerial vehicles (UAV) were designed [1]. Those remotely-controlled devices were thought mainly for military purposes and today, thanks to the advances in several fields of engineering, its popularity among civilians applications has significantly increased. Such tendency seems to continue growing and their appearance in several commercial services is expected [2]. Usages such as delivery of products, remote assistance, monitoring, etc. will be facilitated by the usage of drones.

However, at the same time that UAVs gain popularity in the industry, there are several concerns about possible illegal activities that can be carried out with them. Some examples are drug traffic, terrorist attacks, violation of privacy, among several others [2]. The eventual misuse of UAV has originated interest in the development of technologies that prohibit their usage in certain zones and more importantly, that identify their position with accuracy.

One approach to identify the position of a drone consists of utilizing the signal that it uses to communicate with its remote control, which can be seen as a typical case of wireless positioning. In this case, a set of receiving stations can be deployed in a certain region in order to listen the transmitted signal and determine the drone location. The existing methods for wireless positioning typically use the information of angle of arrival (AoA), propagation time or received signal strength (RSS) to establish the position of a device [3]. Among these three variables, the last one is of special interest for the problem previously described because it does not require special signaling (as the propagation time does), nor several antennas at each station (as the AoA demands).

However, the characteristics of wireless channels offer several challenges to estimate the position of a terminal based on the RSS. In many scenarios, the environment lacks of a line of sight (LOS) propagation and then, assuming that the RSS depends exclusively on the distance by some polynomial decay is inaccurate. In practice, numerous rays participate in the propagation from the transmitter to the receiver and several propagation mechanisms such as reflections, diffractions, scatterings, etc. reduce the strength of each ray [4]. Under this picture, it is necessary to use a better model of the path loss that accurately represents the specific conditions of each environment.

The Austrian Institute of Technology (AIT)[1] has a ray tracer designed in the

[1] http://www.ait.ac.at/en/

Ph.D. degree of Minming Gan [5]. This simulation tool can be used to determine the possible locations of an UAV, based on its RSS. In principle, a map of the RSS can be elaborated for a specific scenario from the results of the ray tracer. However, it has been initially designed for wideband indoor, ultrawideband (UWB) indoor and tunnel scenarios, and so far, its accuracy in predicting the characteristics of wireless channels in outdoor environments and specifically, city street canyons has not been tested.

Another possibility to elaborate the relation between RSS and position consists of taking a measurement campaign in the particular scenario. Rui Wang *et al.* has developed a channel sounder in the University of Southern California (USC) [6], which can be utilized to take the measurements. This channel sounder is implemented on a software-defined-radio (SDR) system and measures up to 15 MHz.

As an initial stage to solve the localization of UAVs, in this project we explore the performance of the AIT ray tracer against the channel sounder when modeling city street canyons. Overall, we have worked on the basis of a general objective and four specific objectives:

General objective

Compare the performance of the AIT ray tracer for modeling the propagation of wireless waves in city street canyons, with an updated version of the channel sounder developed by Rui Wang, *et al.*

Specific objectives

1. Upgrade Wang's channel sounder from measuring 15 MHz to 50 MHz.
2. Model a city-street-canyon scenario in the AIT ray tracer and simulate it.
3. Take a short measurement campaign of the same scenario with the upgraded channel sounder.
4. Compare the results of both tools in terms of the local scattering function and establish a verdict of whether the ray tracer might be suitable for wireless positioning or not.

Observe that the specific problem of wireless positioning is not specifically addressed in this project. The document is organized as follows. In Chapter 2, we present a detailed description of the channel sounder, the enhancements that we introduced, and an explanation of the sounding signal. Then, Chapter 3 describes the AIT ray tracer and explains the models it utilizes to predict the propagation rays. Subsequently, we present the modeled scenario in Chapter 4, where additionally, the results obtained with both tools are shown. Finally, Chapter 5 collects the most important observations of this thesis and presents a list of future tasks.

Chapter 2

Channel Sounder

It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong.

Richard Feynman

From a fundamental perspective, any communication process consists of a transmitter, a receiver and a channel. The transmitter perturbs the channel, whose response is collected by the receiver. The communication highly depends on the channel and then, it is of ultimate importance to know its characteristics. This requires to measure the channel, which is known as *channel sounding* [4].

The channel sounder developed in this project is an extension of the one implemented by Rui Wang *et al.* at the University of Southern California [6]. Wang's sounder has been upgraded from using 15 MHz to 50 MHz. The sounder consists of a transmitter that periodically sends a multi-tone signal (see section 2.3) and a receiver that simultaneously takes time samples of the channel. In a later stage, the samples collected by the receiver are used to obtain the channel transfer function and its impulse response.

In this Chapter, the components of the channel sounder will be explained in detail. Section 2.1 will first present the implementation of the sounder, while Section 2.2 explains the sounding process. Section 2.3 introduces the characteristics of the sounding signal and the analysis that is performed to obtain the channel impulse response.

2.1 Measurement Setup

Both, the transmitter and receiver of the channel sounder consist of a software-defined radio (SDR) unit, a global-positioning-system (GPS) disciplined oscillator (GPSDO) based on a rubidium atomic clock, and a personal computer (PC) as illustrated in Fig. 2.1. The SDR implements the transmitter and receiver mechanisms, the GPSDO provides synchronization and the host PC is in charge of controlling the sounding process.

At the transmitter side, the sounding sequence is generated on the host PC,

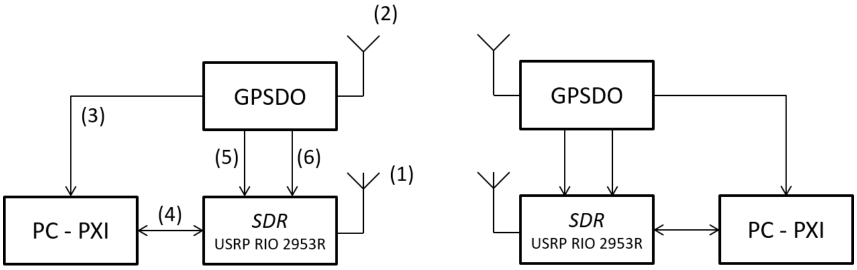

Figure 2.1: Measurement setup. The elements with numbers are: 1) Tx/Rx antenna. 2) GPS antenna. 3) Ethernet cable. 4) PCIe x4 cable. 5) SMA Coaxial cable: 1PPS. 6) SMA Coaxial cable: 10MHz sine wave.

which sends baseband samples of the sequence to the SDR by means of a PCIe x4[1] cable. The SDR performs digital-to-analog conversion and frequency modulation of the signal to be transmitted. On the receiver side, the SDR down-converts the received signal and generates digital samples of the baseband signal. Those samples are then sent to the host PC by using a PCIe x4 cable. On each terminal, the GPSDO maintains synchronization by providing a 1-pulse-per-second (1 PPS) signal and a 10-MHz signal as references in time and frequency, respectively. The GPSDO also supplies the coordinates and time of the current measurements to the PC through an Ethernet cable. The host PC stores these variables together with the channel measurements for further processing.

A deeper explanation of the three main components of Fig. 2.1, together with some other equipment will now be presented.

2.1.1 Software Defined Radio (SDR)

A SDR is a radio system where a large part of its functionalities are implemented digitally [7]. In this project, the universal software radio peripheral (USRP) RIO 2953R from National Instruments [8] has been used as SDR. Among other parts, it contains a field-programmable gate array (FPGA) Xilinx Kintex 7 K7410T [9], where the specific functions of the channel sounder are programmed [10], and a CBX-120 daughter board [11] that is in charge of the conversion from a digital baseband signal to an analog passband signal (or vice-versa).

At the transmitter, the FPGA contains a direct-memory-access (DMA) first-input-first-output (FIFO) queue, that temporally stores the sounding samples sent by the host PC as shown in Fig. 2.2. Each sample is then scaled and passed to a fractional interpolator, which generates an equivalent sounding signal with the same data rate at which the digital-to-analog converter (DAC) of the daughter board works. Meanwhile the rate adjustment is necessary as the signal sent by the host PC has in general a different rate from the DAC rate, the scaling is useful to assure that the samples lies in the dynamic range of the DAC. The signal is then

[1] PCIe stands for peripheral component interconnect express and is a high speed wired standard.

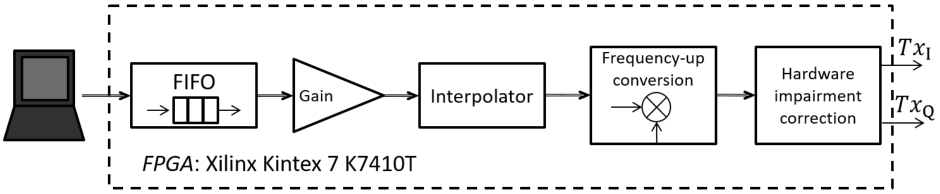

Figure 2.2: Functionalities implemented in the transmitter FPGA.

modulated to an intermediate frequency (f_IF). As it will be explained later, this up conversion is useful to avoid the leakage of the analog local oscillator. Finally, a partial hardware-impairment correction is implemented according to some factory calibration settings.

The resulting signals from the FPGA: Tx_I (in-phase) and Tx_Q (quadrature) are passed to two DACs of the daughter board, as it can be seen in Fig. 2.3. Then, the analog signals are modulated with a tunable local oscillator (f_LO). After adding the in-phase and quadrature components, the signal is amplified inside the USRP and then by an external wide-band power amplifier [12], to finally be transmitted by an omnidirectional antenna [13].

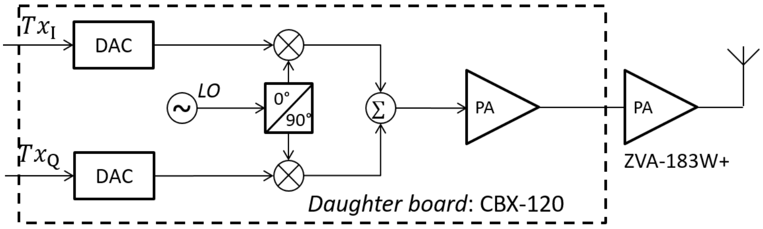

Figure 2.3: Block diagram of the transmitter daughter board.

Correspondingly, the daughter-board block diagram on the receiver is shown in Fig. 2.4. The signal captured by the antenna is first passed through a low-noise amplifier (LNA) and then by a variable amplifier (actually it is a constant amplifier and a variable attenuator), used to implement an automatic-gain control (AGC) (see Section 2.2.2). The signal is subsequently down-converted to an intermediate frequency (f_IF) and filtered to avoid aliasing in the analog-to-digital converter (ADC). Finally, the baseband signal is sampled and passed to the FPGA.

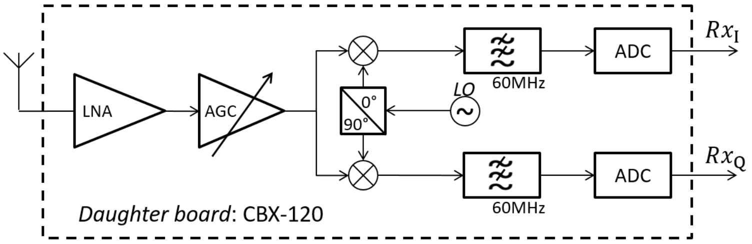

Figure 2.4: Block diagram of the receiver daughter board.

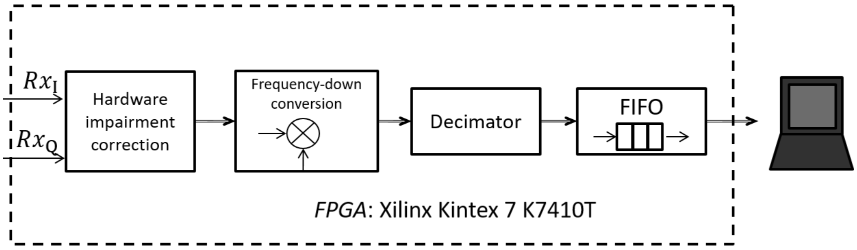
Rx_I
Rx_Q
Hardware impairment correction
Frequency-down conversion
Decimator
FIFO
FPGA: Xilinx Kintex 7 K7410T

Figure 2.5: Functionalities implemented in the receiver FPGA.

In the receiver FPGA, the samples are partially compensated for hardware impairment before being digitally down-converted (see Fig. 2.5). Like in the transmitter case, the measured sequence of samples is adjusted to match the required sampling rate by the host PC, which is done by a fractional decimator. Once the correct rate is achieved, the samples are placed in a DMA FIFO and read by the host PC.

Some of the characteristics of the used SDR are summarized in Table 2.1

USRP 2953R 120 MHz [8]	
FPGA	Xilinx Kintex 7 K7410T [9]
Daughter board	CBX-120 [11]
Bandwidth	120 MHz
Local oscillator frequency	1.2 – 6 GHz
ADC Sampling rate	200 MS/s
Maximum output power	7 – 20 dBm*

Table 2.1: Characteristics of the used SDR

* Depending on the local oscillator frequency

2.1.2 Global-Positioning-System (GPS) Disciplined Oscilator (GPSDO)

In order to accurately determine characteristics of the channel, such as the duration of the impulse response, its time delay, or the Doppler frequency shift, the transmitter and receiver need to be synchronized in time and use, with high precision, the same frequencies in their radio-frequency (RF) chains. To achieve this goal, a GPSDO is utilized at each terminal as an external reference.

A GPSDO is an oscillator that is disciplined by a global navigation satellite service (GNSS) [14], which means that it actively corrects its frequency according to the signals received from a selected GNSS (typically GPS). In this project, the GPS10eR [14] was used as GPSDO, which has a rubidium atomic clock as main oscillator. This device provides a 1-PPS signal as time reference and a 10-MHz tone as frequency reference. The characteristics of both signals are summarized in Table 2.2.

In addition to provide time and frequency references, the GPSDO is also used

GPS10eR [14]

Signal	Time reference	Frequency reference
Shape	1PPS - TTL square signal	10 MHz sine wave
Precision	Aligned ±20 ns with UTC*	$2 \times 10^{-13\dagger}$
Signal level	$3.3V_{\text{peak}}$	$3.3V_{\text{peak-to-peak}}$

Table 2.2: Characteristics of the used GPSDO [14]

* Coordinated Universal Time

† Allan Deviation

to acquire the coordinates and time instants of each measurement. This task is accomplished by connecting each GPSDO with the corresponding host PC by means of an Ethernet cable.

Finally, each GPSDO has an active quadrifiliar helix antenna [15] to capture the GPS signals.

2.1.3 Host PC

On each terminal, the sounding process is controlled by a PXIe-8135 (the host PC) [16] system controller, mounted in a PXIe-1082 [17] chassis. In addition to the host PC, the chassis also has a solid-state, redundant array of independent disks (RAID) (NI HDD-8261) to store the measured data, and a MXI-Express x4 link (NI PXIe-8374) to communicate with the SDR. These two are high speed modules for storing and communication.

As it will be presented later, the host PC of the transmitter and receiver have to communicate to each other in the beginning of each sounding signal. This procedure is done by deploying a wireless local area network (WLAN) between the two terminals over the 2.4 GHz industrial, scientific and medical (ISM) band. Specifically, each PC is connected to a Linksys WRT54GL [18] router, that implements the 802.11g standard.

2.1.4 Power Supply

Each station uses a DC/AC converter and a battery to run measurements without access to the electric network. The DC/AC converter [19] is capable of delivering up to 800 VA. The battery, in turn, is a Victron Energy gel cell [20], works at 12 V and stores up to 165 Ah of charge.

According to some conservative estimates on the power consumption ($\sim$ 500 W), each terminal can run up to 3.5 h on its battery.

2.2 Sounding Procedure

In this Section, the programs running on the FPGA and on the host PC are presented. Both programs were initially developed in LabVIEW [21] and required to be updated due to the increase in the sounding bandwidth. Several functionalities of those programs were written again in recent versions of LabVIEW. Overall, the software versions used for this project are:

1. LabVIEW 2016.
2. USRP drivers 16.10.
3. Device drivers 16.8.
4. LabVIEW 2016 Real-Time module.
5. LabVIEW 2016 FPGA module.

2.2.1 FPGA Program

The program running on the FPGA was designed by Wang *et al.* at the University of Southern California [6] and was implemented on a USRP with 40 MHz bandwidth. To increase the bandwidth of the channel sounder, we have adapted the program to a 120 MHz USRP. Overall, it is based on a streaming template [10] provided by LabVIEW and presents two additional functionalities that will be explained below.

The LabVIEW streaming template implements the state machine of Fig. 2.6 at both terminals: transmitter and receiver. It consists of an idle state, where the system does not transmit or process any signal, and an active state, where the system executes the digital signal processing functionalities of Figs. 2.2 (transmitter) or 2.5 (receiver). Normally, the system stays in the idle state until a trigger signal Tr is received. In such a case, it passes to the active state and processes N_s samples of the Tx/Rx signal, each of them during a clock cycle. After the samples have been processed ($n_\mathrm{s} = 1$) or whenever a reset command R is received, the machine goes back to the idle state. The trigger and reset signal can be generated externally, for example, from the host PC.

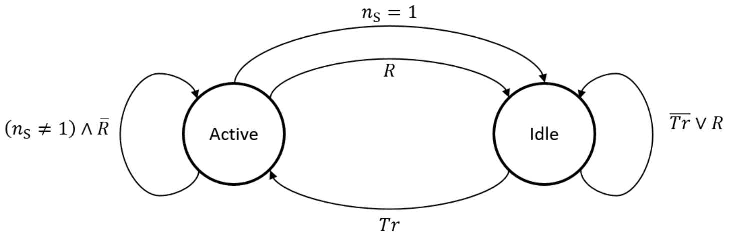

Figure 2.6: State machine running on the FPGA. n_s is the sample's down counter: $n_s = N_s \ldots 1$.

In addition to the state machine, the channel sounder implements two extra functionalities:

1. *PPS synchronization:* In order to keep time synchronization with the external GPSDO, the incoming PPS signal is read and a trigger is produced whenever it has a rising edge.

2. *Software and reset counters* Two counters periodically emit a software trigger and a reset signal. Their maximum values can be set at the host PC.

Notice that in the channel sounder program, the trigger signal of Fig. 2.6 is produced either by the external PPS or the software counter. The reset signal is produced by the respective counter and can also be generated from the host PC. With this configuration, the time structure between states is determined by the maximum values of the counters (N_{SW}: software and N_{R}: reset) and the frequency of the FPGA clock f_{ck}. The time interval between two transmissions is[2]:

$$T_{\mathrm{rep}} = (N_{\mathrm{SW}} + 1)/f_{\mathrm{ck}}, \tag{2.1}$$

while the time at which a reset signal is produced after the start of a transmission or reception is:

$$T_{\mathrm{Reset}} = (N_{\mathrm{R}} + 1)/f_{\mathrm{ck}}. \tag{2.2}$$

The active time is the minimum between the required time to process N_{s} samples at a sampling rate R_{s}, and T_{Reset}:

$$T_{\mathrm{active}} = \min\{N_{\mathrm{s}}/R_{\mathrm{s}},\ T_{\mathrm{Reset}}\}. \tag{2.3}$$

This time structure is summarized in Fig. 2.7. Notice the flexibility that this structure offers, as the number of samples and the maximum values of both counters can be set by the host PC. For the measurements taken in this project, the values of the variables of (2.1) - (2.3) are summarized in Table 2.3.

Variable	Value
N_{SW}	199 999
N_{R}	19 999
N_{s}	800
f_{ck}	200 MHz
T_{rep}	1 ms
T_{Reset}	100 µs
T_{active}	13.79 µs

Table 2.3: Maximum values of the counters and time parameters involved in the FPGA structure. Here, $R_{\mathrm{s}} = 58$ MS/s.

As previously stated, the program has been upgraded to run on 120 MHz USRPs. This required the update of the USRP drivers to the 2016.10 version and the usage of LabVIEW 2016. Due to the larger bandwidth of the USRP, the sampling rate of the ADC is also higher ($f_{\mathrm{ck}} = 200$ MHz in the 120 MHz USRP, compared to $f_{\mathrm{ck}} = 120$ MHz in the 40 MHz USRP). The same clock used in the ADC is utilized by the FPGA, which naturally, has to process more samples per second. As a result, the digital signal processing of the FGPA has a tighter

[2] Each counter starts counting at 0.

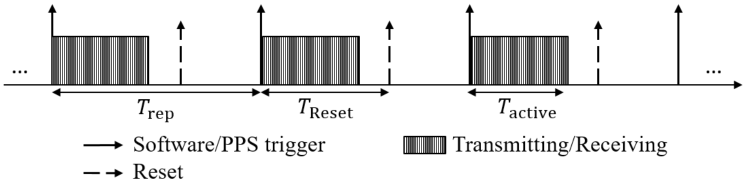

Figure 2.7: Time structure on the FPGA.

time schedule, which the original code could not achieve. We needed to rewrite the FPGA code in the new software versions, which consumed most of the time in the development of the project. To achieve the new timing requirements, we introduced the following two modifications:

1. The execution line was separated by some shift registers, which is known as pipeline execution [22]. Specifically, the software and reset counters now work in parallel to the digital signal processing.
2. The DMA FIFO length at the transmitter and receiver was reduced to 1023 elements.

2.2.2 Program at the host PC

In the same way as the FPGA code, the program at the host PC was initially developed by Wang *et al.* at USC [6], and was later used by Aleksandar Dichev at AIT [23]. We have updated it to run in LabVIEW 2016, which also required several modifications in the code. In general, the code can be seen as two main parts: an initial stage where the channel sounder is configured, and a main loop where it actually runs.

During the initial stage, the USRP at each end is configured to execute the channel sounder functionalities. This process consists of opening an USRP RIO session, sending the bit file with the FPGA program, and defining parameters according to the user settings such as the software and reset counters, the sampling rate and local oscillator frequency. Additionally, the connection between the host PC and the corresponding GPSDO is also established during the initial stage. The transmitter and receiver exchange their GPSDO status (whether it is locked or not) by an external link (the WLAN network of section 2.1.3). This link is also useful to share some variables as the GPS coordinates and some flags that indicate that the measurement can start. Once everything is ready, the user can start the measurement at the transmitter side, and both programs will pass to the main loop of the code.

In the main loop, the procedure is different on the transmitter and receiver side. In the former, the signal to be transmitted is sent to the DMA FIFO while the position and time from the GPSDO are stored in a binary file. In the latter, the data is firstly read from the DMA FIFO and then three simultaneous operations are executed: the data is written in a binary file, the AGC is implemented and the current gain is stored as illustrated in Fig. 2.8. Another cycle is executed in

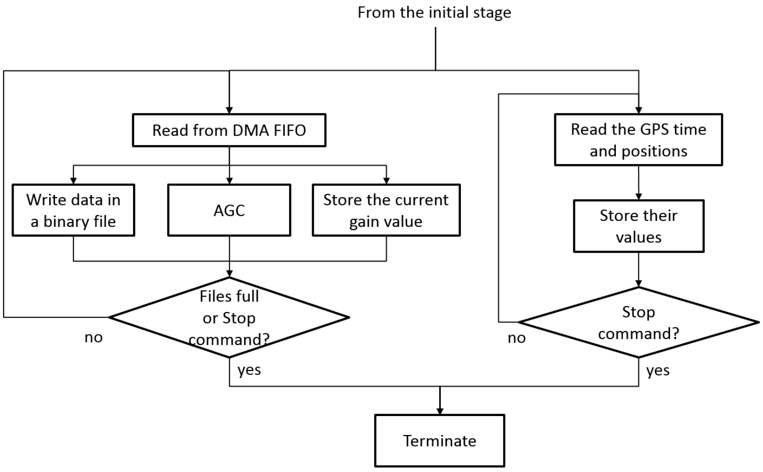
From the initial stage
Read from DMA FIFO
Write data in a binary file
AGC
Store the current gain value
Files full or Stop command?
no
yes
Read the GPS time and positions
Store their values
Stop command?
no
yes
Terminate

Figure 2.8: Main cycle at the reciver host program.

parallel, which reads data from the GPSDO and stores it together with the current measurement number.

The cycles at both terminals are controlled by the FPGA time structure of Fig. 2.7. Observe that the receiver can only implement the three operations that come after reading the DMA FIFO during a time equal to:

$$T_{\mathrm{idle}} = T_{\mathrm{rep}} - T_{\mathrm{Reset}}, \tag{2.4}$$

that is, during the time where the FPGA is not collecting data. This imposes some restrictions on the values of N_{s}, N_{SW} and R_{s}, and an error appears in case the timing is violated.

Regarding the AGC, it consists of an on-off control, that checks whether the maximum value of the signal magnitude is within the ADC dynamic range. If the value is less than a minimum threshold, the gain increases by 2 dB, while it decreases by 2 dB in case that the maximum of the signal magnitude is above the upper threshold. This procedure avoids a high quantization noise when the signal has low power and prevents clipping on the contrary case. Naturally, in a latter processing stage, this gain needs to be considered and that is why the current gain of the AGC is also stored. We had to modify some parts of the initial AGC to make it faster and compatible with the new drivers.

Finally, the program at the receiver finishes either when all the desired data has been collected or when the user stops the data acquisition with a special command. The transmitter will be continuously sending the sounding signal until the user stops the program.

2.3 Sounding signal

2.3.1 Signal description

Suppose that it is desired to sound a bandwidth BW, with a resolution Δf. A simple approach to achieve this consists of building a signal with $N_{\mathrm{q}} = BW/\Delta f$

tones, spaced Δf apart. A complex baseband representation of such a signal is[3],

$$\tilde{x}(t) = \sum_{q=-(N_\mathrm{q}-1)/2}^{(N_\mathrm{q}-1)/2} x_q e^{j2\pi q\Delta f t}, \tag{2.5}$$

where x_q is the complex coefficient of the q-th tone. In principle, there is no reason why one would prefer one frequency component over another but instead, it is desired to measure each of them with the same power. Therefore, x_q is restricted to have the same magnitude, regardless of the tone,

$$|x_q| = 1, \qquad \forall\, q\ \in \{-\tfrac{N_\mathrm{q}-1}{2}, \tfrac{N_\mathrm{q}-1}{2}\}\,. \tag{2.6}$$

The signal $\tilde{x}$ is then determined by the phases of the tones, which are not arbitrary but constrained by another condition. Because the passband representation of (2.5) will be later amplified at the transmitter, it is of interest that it keeps a low crest factor C_c to avoid non linearities of the power amplifier [4]. All together, the complex coefficients of the sounding signal can be found by solving the optimization problem

$$\begin{aligned} \underset{\{x_q\}}{\text{minimize}} \quad & C_\mathrm{c} = \max_{t\in[0,T]} \{\tilde{x}(t)\} / \sqrt{\frac{1}{T}\int_0^T |\tilde{x}(t)|^2 dt} \\ \text{subject to} \quad & |x_q| = 1, \quad q = -\tfrac{N_\mathrm{q}-1}{2}, \dots, \tfrac{N_\mathrm{q}-1}{2}, \end{aligned} \tag{2.7}$$

where C_c is the baseband crest factor of the signal, and T is its period.

To solve the problem of (2.7), the algorithm proposed by Mathias Friese in [24] has been used. The method defines an error function in time and then uses its frequency representation to determine the phase of the tones that minimizes it. This method has been run for several number of tones and the results of the crest factor can be found in Table 2.4.

N_q	C_c	BW[MHz] [9]
61	1.184	15.25
161	1.167	40.25
181	1.138	45.25
201	1.212	50.25
221	1.182	55.25

Table 2.4: Crest factor C_c and bandwidth BW (when $\Delta f = 250$ kHz), for sounding signals with different number of tones N_q. The 15 MHz signal was calculated again because the original signal had a large crest factor: $C_\mathrm{c} = 1.49$.

[3] The number of tones has been restricted to be odd for simplicity. It does not have to be odd, though.

Regarding the frequency resolution, the signals have been constructed with $\Delta f = 250$ kHz. This value, which can be conveniently modified, determines the maximum delay that the channel sounder can measure,

$$\tau_{\max} = \frac{N_{\mathrm{q}} - 1}{N_{\mathrm{q}}} \frac{1}{\Delta f} \approx \frac{1}{\Delta f} = 4\ \mathrm{\mu s}. \tag{2.8}$$

Additionally, observe that by construction, the signal in (2.5) has period

$$T = \frac{1}{\Delta f} = 4\ \mathrm{\mu s}. \tag{2.9}$$

Once the coefficients of the tones have been calculated according to the procedure above, a digital version of the baseband signal is obtained by sampling (2.5) with rate R_{s}. Those samples are then sent by the host PC to the DMA FIFO of the USRP, as explained in section 2.1.1.

2.3.2 Frequency analysis of the signal

As presented before, the signal passes through several modules from the DMA FIFO at the transmitter to the receiver FIFO. In the following, the frequency content of the signal will be studied when it passes by each of those modules (Figs. 2.2 - 2.5).

The spectrum of the digital baseband signal that is stored in the DMA FIFO at the transmitter is centered around zero, as defined in (2.5). This can be seen in the upper picture of Fig. 2.9, where the magnitude of the spectrum is shown asymmetric to emphasize that in general, $\tilde{X}(f) \neq \tilde{X}(-f)$.

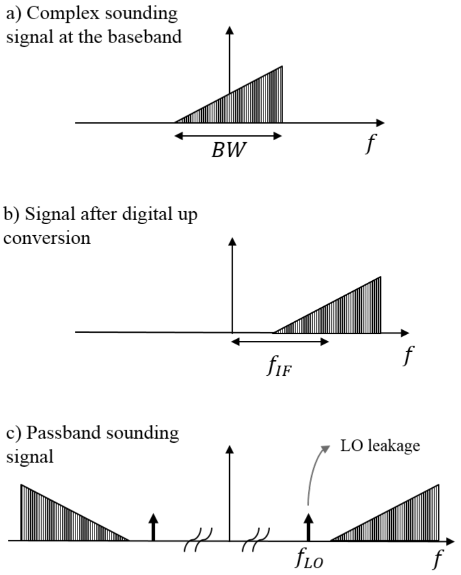
a) Complex sounding signal at the baseband
BW
f
b) Signal after digital up conversion
f_{IF}
f
c) Passband sounding signal
LO leakage
f_{LO}
f

Figure 2.9: Frequency sketch of the signal at the transmitter.

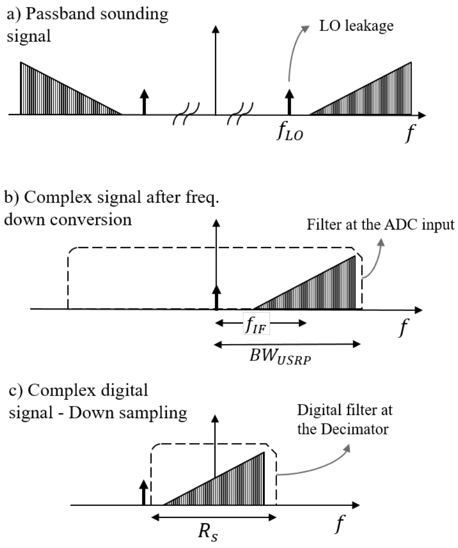
a) Passband sounding signal
LO leakage
f_{LO}
f
b) Complex signal after freq. down conversion
Filter at the ADC input
f_{IF}
f
BW_{USRP}
c) Complex digital signal - Down sampling
Digital filter at the Decimator
R_s
f

Figure 2.10: Noiseless frequency sketch of the signal at the receiver.

After going through the interpolator (see Fig. 2.2), the signal is then digitally up converted to an intermediate frequency f_{IF} (see Fig. 2.9.b) and then converted into an analog signal. Once being analog, it is frequency shifted, i.e. modulated, by a local oscillator (f_{LO}) to obtain the single-side-band (SSB) signal at the lower picture of Fig. 2.9. Observe that the possible leakage of the local oscillator remains outside the sounding signal, avoiding interference. This approach, however, limits the useful bandwidth of the USRP to a half, which explains why only less than 60 MHz of the 120 MHz can be used to sound the wireless channel.

As previously explained, the signal captured by the receiver is firstly amplified and then frequency down converted by a local oscillator tuned at the same frequency f_{LO}, as shown in Figs. 2.10 a, b. Notice that the local oscillator leakage of both, transmitter and receiver, are now at the center of the spectrum, while the signal is located around f_{IF}. To convert from analog to digital, a low pass filter is needed to avoid aliasing, which has a $BW_{USRP} = 60$ MHz bandwidth in the case of the 120 MHz USRP. As a result, the maximum useful bandwidth[4] BW is constrained by

$$BW < 2(BW_{\mathrm{USRP}} - f_{\mathrm{IF}}). \tag{2.10}$$

After the signal is sampled, it is digitally down-converted to obtain the spectrum around zero (see Fig. 2.10.c). With this shift, the local oscillator leakage is moved to $-f_{\mathrm{IF}}$, and can be filtered by the fractional decimator, which consists of (an up-sampler,) a filter and a down-sampler [25]. To achieve this, the requested

[4] In this project, the signal bandwidth is always the frequency width of the passband signal.

sampling rate R_s at the receiver must satisfy

$$R_\mathrm{s} < 2f_\mathrm{IF}. \tag{2.11}$$

However, it cannot be arbitrarily low, otherwise the desired signal would be lost,

$$BW < R_\mathrm{s}. \tag{2.12}$$

All together, (2.10) - (2.12) limit the values that can be chosen on the intermediate frequency and the sampling rate. Of the possible bandwidths of Table 2.4, the largest is also the most restrictive. To satisfy the fulfillment of the three requirements for such bandwidth, the following values have been selected:

$$\begin{aligned} f_\mathrm{IF} &= 31 \text{ MHz}, \\ R_\mathrm{s} &= 58 \text{ MS/s}. \end{aligned} \tag{2.13}$$

They can be relaxed for other bandwidths to avoid being close to the filters' edges. Although the sampling rate at the transmitter has not to be the same that at the receiver, they have been made equal for simplicity.

Finally, the baseband samples in the end of the digital signal processing at the receiver FPGA are stored in the corresponding DMA FIFO. They are later read by the host PC and stored for further processing.

2.3.3 Signal processing after the measurement

In the initial version of the channel sounder, the transmitter repeatedly sends a single period T of the baseband signal as illustrated in Fig. 2.11, a. In order to identify components with a maximum delay T, a $2T$-long sequence y_meas of the received signal is utilized for the analysis carried out after the measurement. With this sequence, it is possible to identify frequency components separated

$$f_\mathrm{res} = \frac{1}{2T} = \frac{\Delta f}{2} \tag{2.14}$$

apart, i.e. the frequency resolution is half of the tone spacing. Therefore, to obtain the frequency components where the tones are sent Y_2, it is sufficient to take every second frequency sample of the measured signal, Y_meas,

$$Y_2[q] = Y_\mathrm{meas}[2q] \qquad q = -\tfrac{TR_\mathrm{s}}{2}, \ldots, \tfrac{TR_\mathrm{s}}{2} - 1, \tag{2.15}$$

where R_s is the sampling rate.

In order that the sounding components appear exactly at the samples of (2.15), the period T of the signal must be an integer of the sampling period $t_\mathrm{s} = 1/R_\mathrm{s}$,

$$T = N_T t_\mathrm{s} = \frac{N_T}{R_\mathrm{s}}, \qquad N_T \in \mathbb{Z}^+. \tag{2.16}$$

Otherwise, spectrum leakage would appear and the sounding components could not be independently identified (they are no longer orthogonal). Observe that the sampling rate in (2.13) satisfies (2.16), and in our case N_T is even. However, the

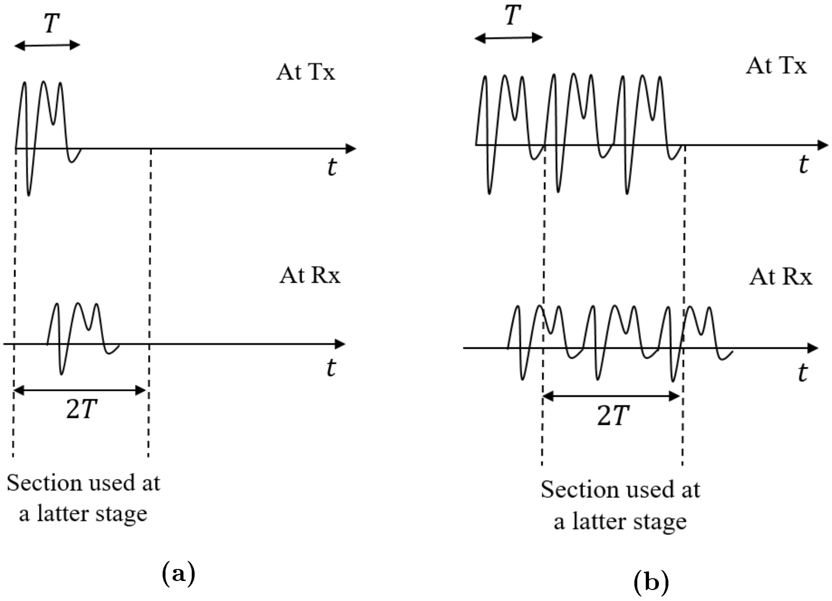

Figure 2.11: Noiseless graphical representation of the transmitted and received signals in (a) the initial version of the channel sounder and (b) our modified version.

integer N_T does not coincide with the number of tones N_q, and hence, $(N_T - N_\mathrm{q} + 1)/2$ frequency samples in the beginning and $(N_T - N_\mathrm{q} - 1)/2$ in the end are eliminated from Y_2 to obtain the sounded tones,

$$Y[q] = Y(f_\mathrm{c} + q\Delta f) \quad q = -\tfrac{N_\mathrm{q}-1}{2}, \ldots, \tfrac{N_\mathrm{q}-1}{2}. \tag{2.17}$$

Notice that with this sounding scheme, around half of the taken time samples are just noise. To increase the signal-to-noise ratio (SNR), we proposed a modification in which three periods of the signal were transmitted instead of one. The receiver still takes a $2T$-long signal but one period later as shown in Fig. 2.11, b.

The difference now is that we are always taking samples with the signal. The noise is still the same and corresponds to the samples during the $2T$ interval. However, the resultant amplitude of the signal in (2.17) is twice as much as before, and hence, the SNR increases by 6 dB. Of course, transmitting the signal three times implies an increase of 4.77 dB of the energy of each sounding pattern. Finally, observe that this analysis can be seen as an orthogonal-frequency-division-multiplexing (OFDM) signal whose cyclic prefix is one period T of the sounding sequence, and whose actual length is $2T$. The receiver discards the cyclic prefix to obtain a circular convolution between the channel and the signal [4].

2.3.4 Calibration

The resultant signal from the measurement of the previous section contains the transfer function of the channel and the antennas H_CH, the RF chains at the transmitter and receiver H_{RF}, and the transmitted signal X. It can be written as:

$$Y[q] = H_{\mathrm{CH}}[q]H_{\mathrm{RF}}[q]X[q] + N[q], \quad q = -\frac{N_{\mathrm{q}}-1}{2}\dots\frac{N_{\mathrm{q}}-1}{2}. \tag{2.18}$$

where N is the additive noise and is assumed to be white and Gaussian distributed. The fact that each tone can be studied independent from the others is a consequence of the OFDM-like structure of the sounding signal, which avoids inter-carrier interference and inter-symbol interference (ISI)[5] [4].

As it can be seen from (2.18), the measured signal at the receiver also contains the effect of the RF chains and hence cannot be directly used to obtain the channel transfer function. A calibration is implemented to determine the RF-chain response by directly connecting the transmitter and receiver with a coaxial cable and an attenuator. The same cable, without attenuation, is also used during the measurements (before the transmitter antenna) and hence its effect is included inside H_{RF}, which also includes the response of the external power amplifier. The calibration is repeated 1000 times before each measurement to average each realization and reduce the signal to noise ratio (SNR). As a result, a calibration signal Y_{CAL} is obtained,

$$Y_{\mathrm{CAL}}[q] \approx H_{\mathrm{RF}}[q]X[q], \qquad q = -(N_{\mathrm{q}}-1)/2\dots(N_{\mathrm{q}}-1)/2. \tag{2.19}$$

The transfer function of the channel (including the antennas) can then be estimated by calculating the ratio, tone by tone, of the measured signal Y and the calibration reference Y_{CAL},

$$\hat{H}_{\mathrm{CH}}[q] = \frac{Y[q]}{Y_{\mathrm{CAL}}[q]}, \qquad m = -\frac{N_{\mathrm{q}}-1}{2}\dots\frac{N_{\mathrm{q}}-1}{2}. \tag{2.20}$$

Observe that the noise of (2.18) is also divided by Y_{CAL} and hence, some of the frequency components suffer from noise enhancement. In fact, there is a difference of up to $1.73dB$ between some tones in the RF-chain transfer function, as illustrated in Fig. 2.12. The main difference is at the right side of the spectrum, where the effect of the USRP filter becomes noticeable (see Fig. 2.10 b).

The estimated channel transfer function of (2.20) corresponds to the channel response at a particular time t_m, at which the transmitter sent the sounding signal. Recall from section 2.2.1 that t_m is a multiple of the repetition time T_{rep}. Therefore, each measurement of the channel can be indexed by two variables:

$$H_{\mathrm{CH}}[m,q] = H_{\mathrm{CH}}(mT_{\mathrm{rep}}, f_{\mathrm{c}} + q\Delta f), \tag{2.21}$$

such that $H_{\mathrm{CH}}[m,q]$ represents the time-variant channel transfer function at the instant $t_m = mT_{\mathrm{rep}}$, and q varies over the sounding tones: $q = -(N_{\mathrm{q}}-1)/2\dots(N_{\mathrm{q}}-1)/2$. This assumes that the variations of the channel during each measurement are small enough to be neglected. The assumption allows to interpret the channel as linear and time-invariant (LTI) for each measurement [4].

[5] The absence of ISI is also a consequence of the selection of T_{rep} (see (2.1))

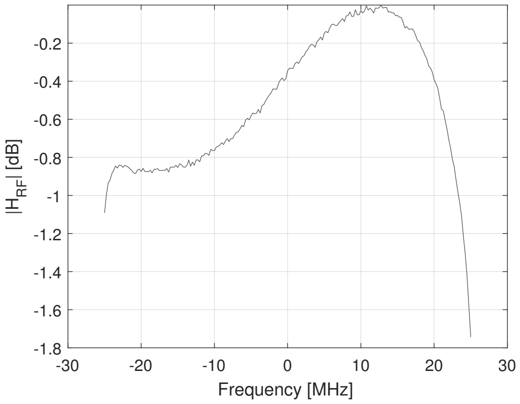

Figure 2.12: RF-chain transfer function H_{RF} when using $BW =$ 50.25 MHz. It is plotted relative to the tone with the highest gain.

Chapter 3

Ray Tracer Channel Modeling

> I do not think that the wireless waves I have discovered will have any practical application
>
> *Heinrich R. Hertz*

The properties of a communication channel can be best determined by empirical measurements. However, as it is evident from the previous chapter, measurements require the usage of specialized equipments and great logistic efforts that impose limitations on budget and time. Moreover, to characterize a certain environment, it is necessary to take several measurements of similar scenarios and thus have a significant sample for a statistical analysis. This picture promotes the usage of channel models that allow an easier prediction of the characteristics of a wireless channel in a given geometry.

One type of channel models are deterministic methods where the propagation properties of a channel are determined by solving the Maxwell equations constrained to the boundary conditions of a specific scenario [4]. There are several approaches to this problem such as the finite-difference time-domain (FTTD) method, the finite element method (FEM) and the method of moments (MoM), that pursuit to derive exact expressions of the electromagnetic fields [26]. In spite of their accuracy, they usually require large computational resources and long execution times, which make them unfeasible, specially in large scenarios whose dimensions are hundreds of wavelengths.

Other deterministic models look for an approximate solution of the Maxwell equations that reduce the inherent complexity of the problem. A common simplification, the so called high-frequency approximation [4], consists of modeling the propagating electric field as a set of several rays, each of them traveling through a specific path. The magnitude of each ray at the receiver is determined from the propagation mechanisms involved in the corresponding path (reflection, transmission, etc.), and are calculated using geometrical optics [27]. Some examples of those approximate models are ray tracing [28] and ray launching [27].

The main difference between those two methods lies on the approach to generate the rays. In ray launching, several rays are uniformly *launched* around the transmitter according to its antenna radiation pattern. Then, the propagation of each ray is studied until it eventually reaches the receiver or is sufficiently attenuated to be neglected [4]. In ray tracing, all the possible paths between transmitter and receiver are determined geometrically[1] and then studied according to the propagation mechanisms involved in each of them.

In this project, we have used the ray tracer developed by Mingming Gan during her Ph.D studies [5], which was later optimized by Xuhong Li [29]. The code, implemented in Matlab [30], takes as inputs the geometric information of a particular scenario and the electromagnetic properties of its components to calculate the possible paths between transmitter and receiver and their magnitude. Throughout this chapter, the main steps of the algorithm will be presented as well as their relation with the corresponding propagation mechanisms. The following, however, is a brief overview of the code and a more detail presentation can be found in Mingming's dissertation [5].

3.1 Preparation of the ray tracer

The ray tracer algorithm starts by reading the geometrical information of the scenario to simulate. Each environment is modeled as a set of rectangular parallelepipeds whose upper and lower faces are always horizontal and parallel to the ground. The description of each block is passed inside a matrix `WedgesInput` that includes the following information:

1. The X and Y coordinates of each of the eight corners of the block, and the height of the lower and upper faces.
2. Some flags that enable or disable scattering and penetration in the block. Another flag to set the block as perfect electric conductor (PEC).
3. The electrical characteristics of the block in terms of its permittivity and conductivity.
4. In case that scattering is enabled, some characteristics of the used model.

In addition, the ray tracer also receives as parameter a matrix `outwedges` that presents:

1. The properties of the ground. Specifically, it contains the characteristics 2 to 4 of the previous list for the ground.
2. The position of the transmitter and receiver.
3. The reflection order, i.e. the maximum number of reflections of a ray.
4. The center frequency of the simulation.
5. The type of antenna used at the transmitter and receiver. So far, just a few antennas are supported and we have always used $\lambda/2$ dipole antennas.

Once the ray tracer has read the input parameters, it finds all the rays in the scenario. This procedure is oriented towards the propagation mechanisms, which means that for each mechanism (except penetration), the algorithm finds the rays that are affected by it.

[1] Some rays are randomly generated, but anyway constrained to geometric conditions.

3.2 Calculation of the line of sight (LOS) component

In this ray tracer, LOS denotes the path that joins the transmitter and receiver by a straight line. To calculate the electric field at the receiver position $\mathbf{E}(\mathbf{r}, f_c)^2$, the free-space loss formula is used [31],

$$\mathbf{E}(\mathbf{r}, f_c) = \mathbf{E}(\mathbf{r_0}, f_c)\frac{c}{4\pi f_c|\mathbf{r}-\mathbf{r_0}|}e^{-j2\pi f_c|\mathbf{r}-\mathbf{r_0}|/c}, \tag{3.1}$$

where $\mathbf{E}(\mathbf{r_0})$ is the electric field at the transmitter position $\mathbf{r_0}$, f_c is the center frequency and c is the wave speed. The effect of the antennas' radiation pattern and polarization is included by introducing two factors, $\mathbf{g_{Tx}} = \mathbf{g_{Tx}}(\mathbf{r}-\mathbf{r_0}, f_c)$ and $\mathbf{g_{Rx}} = \mathbf{g_{Rx}}(\mathbf{r_0}-\mathbf{r}, f_c)$, which account for the amplitude gains and polarizations of the transmitter antenna at the direction of departure $\mathbf{r}-\mathbf{r_0}$, and receiver at the direction of arrival $\mathbf{r_0}-\mathbf{r}$. With this factors, the scalar value of the electric field at the receiver can be written as:

$$E(\mathbf{r}, f_c) = E(\mathbf{r_0}, f_c)\frac{c}{4\pi f_c|\mathbf{r}-\mathbf{r_0}|}e^{-j2\pi f_c|\mathbf{r}-\mathbf{r_0}|/c}\mathbf{g_{Tx}}^{\dagger}\mathbf{g_{Rx}}. \tag{3.2}$$

By using (3.2), the ray tracer calculates the ray corresponding to the LOS component. It is implemented in the function `CalcELOS.m`.

3.3 Calculation of reflected components

The interaction between an electromagnetic wave and a surface can be modeled in terms of a reflected and a transmitted wave [32]. In wireless communications, the reflection of the electromagnetic wave with the objects in a scenario give raise to several components that significantly contribute to the signal captured at the receiver. In principle, each object of the scenario is a potential reflector and hence it is very important to accurately model this mechanism. In the ray tracer, the generation of reflected components is done by a two-step process. First, a reflection tree is built from all the possible reflections in the scenario. Then, the field of the rays going through each possible reflection is calculated.

To calculate all possible reflections in the scenario, the method of images is implemented. This method has been utilized in electrostatics to solve the Poisson equation of the electrical potential when a charge is over an infinite plane [33]. The effect of the plane is replaced by a charge of opposite sign, located on the other side of the plane at the same distance from the boundary that the initial charge is (see Fig. 3.1). In the ray tracer, a similar principle is utilized to model the interaction between the transmitted wave and a surface. Each surface is replaced by an image of the transmitter that becomes the source of a ray traveling to the receiver.

Naturally, each image source can be also reflected by other planes and here is where the concept of the reflection tree appears. The ray tracer first calculates all the possible images that appear from each of the surfaces around the transmitter.

[2] All vectors here are columns, except naturally, if they are transposed (^{T}) or conjugated and transposed (†)

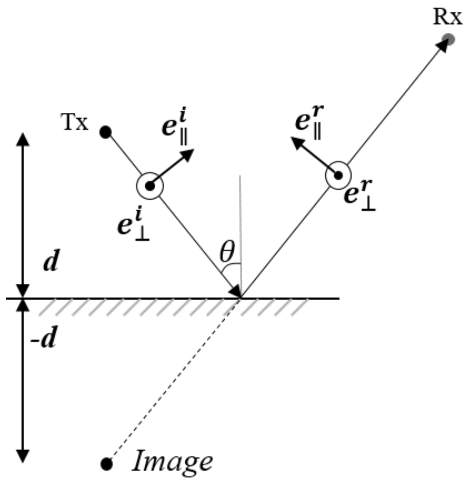

Figure 3.1: Illustration of an image source.

Then, it calculates a second set of images that appear from the first set, and so on until completing the maximum reflection order of the simulation. Each level of the tree consists of the set of images of the same reflection order. Once the images have been obtained, only those whose rays intersect their corresponding surface are considered valid reflections. The initial process to obtain the tree was later improved by Xuhong Li [29] and her result is the current version of the code: `Construct.m`.

After finding the image tree, the algorithm then calculates the electric field associated with each branch of the reflection tree (one path). The calculation utilizes the solution of the Maxwell equations when an electromagnetic wave is constrained to the boundaries of an infinite and smooth surface [32]. This solution can be utilized to approximate the interaction with any block of the environment as long as its dimensions are much larger than the wavelength λ, and its roughnesses are much smaller than λ [4]. In the case of a single reflection, the reflected field at position $\mathbf{r}$ can be written as:

$$E_{\mathrm{rf}}(\mathbf{r}) = E(\mathbf{r_0}, f_c)\frac{c}{4\pi f_c s}e^{-j2\pi f_c s/c}\mathbf{g_{Tx}}^{\dagger}\mathbf{R}\mathbf{g_{Rx}}, \tag{3.3}$$

where s is the total length of the path, while $\mathbf{g_{Tx}}$ and $\mathbf{g_{Rx}}$ are the coefficients of the antenna gains and polarization at the transmitter and receiver, respectively. $\mathbf{R}$ corresponds to the dyadic reflection coefficient,

$$\mathbf{R} = \mathbf{e}^{\mathbf{i}}_{\|}\mathbf{e}^{\mathbf{r}\dagger}_{\|}R_{\|} + \mathbf{e}_{\perp}\mathbf{e}^{\dagger}_{\perp}R_{\perp}, \tag{3.4}$$

being $\mathbf{e}_{\|}$ and $\mathbf{e}_{\perp}$ the unitary vectors parallel and perpendicular to the propagation plane of the incident or reflected wave. They also correspond to the possible polarizations of the electric field (see Fig. 3.1). The perpendicular vector $\mathbf{e}_{\perp}$ is the same for the incident and reflected planes. The variables $R_{\|}$ and $R_{\perp}$ are the reflection parameters [34] when the complex effective permittivity of the incident and reflecting mediums are $\epsilon_\mathrm{i} = 1$ and ϵ_r,

$$\begin{aligned} R_\parallel &= \frac{\epsilon_\mathrm{r}\cos(\theta) - \sqrt{\epsilon_\mathrm{r} - \sin^2(\theta)}}{\epsilon_\mathrm{r}\cos(\theta) + \sqrt{\epsilon_\mathrm{r} - \sin^2(\theta)}}, \\ R_\perp &= \frac{\cos(\theta) - \sqrt{\epsilon_\mathrm{r} - \sin^2(\theta)}}{\cos(\theta) + \sqrt{\epsilon_\mathrm{r} - \sin^2(\theta)}}. \end{aligned} \tag{3.5}$$

These calculations are implemented in the function `CalcEref.m`.

3.4 Calculation of diffracted components

Diffraction is a wave phenomenon that has been exhaustively studied in optics [35] and electromagnetism [34]. It covers the interaction of a wavefront with an obstacle and the resulting components that propagate in several directions in addition to the incoming one. In wireless communications, this phenomenon is particularly important because it is responsible that waves *bend* around obstacles and reach zones where the pure line of sight component is unable to reach. Therefore, diffraction needs to be considered in any ray tracing algorithm to obtain smooth transitions of the electric field around obstacles and thus get realistic results.

The calculation of diffracted rays is implemented in the function `CalcEDiff.m`, which executes three main steps. Initially, the visible edges between transmitter and receiver are found with the subroutine `VisibleEdges.m`. Then, a diffracted point is associated to each edge in the function `CalcDiffPt.m`. Finally, the electric field corresponding to each diffracted point is calculated in `DiffField.m`. The ray tracer utilizes an extension of the geometrical theory of diffraction [34], called the uniform theory of diffraction (UTD) [36], to perform this calculation. In short, the electric field is calculated as,

$$E_\mathrm{diff}(\mathbf{r}) = E(\mathbf{r_0}, f_\mathrm{c})\frac{c}{4\pi f_\mathrm{c} s} e^{-j2\pi f_\mathrm{c} s/c} \mathbf{g_{Tx}}^\dagger \mathbf{D} \mathbf{g_{Rx}} \sqrt{\frac{s}{s_\mathrm{Tx} s_\mathrm{Rx}}}, \tag{3.6}$$

where, as before, $\mathbf{g_{Tx}}$ and $\mathbf{g_{Rx}}$ are the coefficients of the antenna gains and polarization at the transmitter and receiver, and $\mathbf{D}$ is the dyadic diffraction matrix, with units $[\ \mathrm{m}^{1/2}]$. The distances from the diffraction point to the transmitter and receiver are s_Tx and s_Rx, meanwhile $s = s_\mathrm{Tx} + s_\mathrm{Rx}$ is the total length of the ray.

3.5 Modeling of diffuse scattering

In practice, many surfaces are far from being smooth [37] and hence, approximating the interaction of an incident wave with them by the pure reflection of (3.3) is not accurate. In the wireless communication literature [38], the resulting components from the interaction between a electromagnetic wave and a rough surface that propagate in directions different from the specular reflection are denoted by

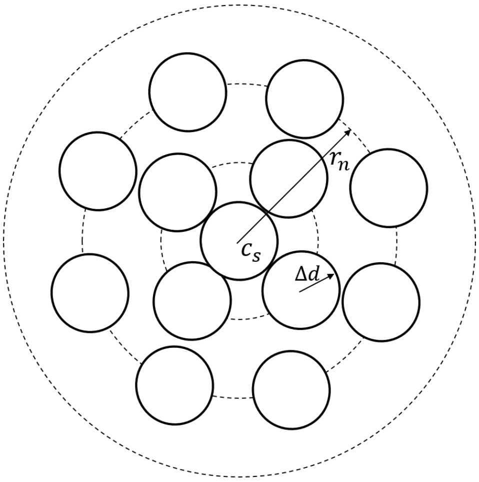

Figure 3.2: Tile generation in the ray tracer.

scattering. Usually, the existence of scattering is assumed when the height of the roughnesses on the surface Δh exceed the so called Rayleigh criterion [39] [40],

$$\Delta h > \frac{\lambda}{8\cos\theta}, \tag{3.7}$$

where θ is the angle between the incident wave and the vector normal to the surface. This can be considered as a practical boundary for using pure reflection (Section 3.3) and introducing scattering.

An approximation to this mechanism consists of dividing the surface into multiple tiles [41] [42], such that each of them represents a generator of a scattered ray. The size of the tiles is upper bounded by the far field condition [31],

$$d_{\mathrm{max}} \leq \sqrt{\frac{d_{\mathrm{s}}\lambda}{2}}, \tag{3.8}$$

where d_{s} is the distance from the center of the tile to the transmitter or receiver. If scattering is allowed for a certain block, the ray tracer generates circular tiles with radius $\Delta d = c/2BW$ over the block [29], checking that (3.8) is satisfied. As illustrated in Fig. 3.2, the center of the tiles lie on concentric circumferences with radius $r_n = 2n\Delta d$, which in turn are located around a random point c_{s} on the rough surface. The tiles on the same circumference are equally spaced but the angle of the first of them in the circle is random.

It is worth pointing out that the generation of diffuse scattering tiles is random as both, the position of the central tile and the distribution of the tiles in each circle are randomly generated in each simulation. This construction imposes some limitations when the ray tracer is used to simulate a time-variant channel. The scattered rays generated in one simulation cannot be tracked in the next one, as a entire new set of rays is generated again. As a result, Doppler shifts associated to diffuse scattering cannot be predicted with the ray tracer as will be evident in Chapter 4.

Two functions are in charge of the generation of scattered rays: `CalcLEds.m` and `CalcFEds.m`. In the first one, all possible scattered rays that result after a reflection are considered. The function starts by using the reflection tree that was initially obtained when finding the reflected components (see Section 3.3) and then obtains all the possible tiles of those images in the function `dspointsQP.m`. Once the tiles have been found, it generates the diffuse scattering field with the function `dsfield.m`.

On the other hand, the function `CalcFEds.m` considers all the cases where diffuse scattering is the first interaction, i.e. when there is only scattering or scattering and then reflection. It initially obtains the diffuse scattering tiles around the transmitter with the function `scatterers.m` and then `construct2.m` is used to find the image tree associated to each tile. As in the previous case, `CalcFEds.m` concludes by calculating the electric field of each scattered path in the function `dsfield.m`.

The calculation of the scattered field inside `dsfield.m` is implemented by using one of three models of the direction of the scattered radiation. In the *Lambertian* model the scattering direction is perpendicular to the surface tile, while it points towards the specular reflection in the *directive* model. The *backscattering* model is similar to the directive one, but additionally introduces a scattered ray in the incident direction. The selection of the model is an input parameter and defined for each block.

As mentioned before, scattering can be seen as a special case of reflection when the surface is rough and hence, considering it should reduce the power of the specular reflection. This effect is modeled by means of the scattering coefficient [38],

$$S = \left.\frac{|E_{\mathrm{ds}}|}{|E_{\mathrm{ref}}|}\right|_{dS}, \tag{3.9}$$

which is the ratio between the scattered electric field $|E_{\mathrm{ds}}|$ and the specular reflected field $|E_{\mathrm{ref}}|$ on a surface dS. This is a parameter defined as input for each block and lies in the range [0,1]. Observe that $R_{\mathrm{ref}} = \sqrt{1-S^2}$ is the factor that describes the losses of the reflected component by scattering, and hence, the field in (3.3) is multiplied by R_{ref} when scattering is included.

3.6 Calculation of penetration

In the ray tracer, penetration is always considered in the analysis of others propagation mechanisms. After finding the set of rays of a certain mechanism, the ray tracer checks whether each ray penetrates the dielectric blocks of the environment with the function `masking.m`. In case that penetration exits, the resultant electric field after crossing the block is calculated in a two-step process.

First, the angle of the transmitted field inside the block is calculated using the Snell's law [34] [35],

$$n_1 \sin\theta_1 = n_2 \sin\theta_2, \tag{3.10}$$

that establishes the relation between the propagation angles of the incident θ_1

and transmitted field θ_2[3], with the refraction indexes n_1 and n_2 of the materials. The refraction index is the ratio between the wave speed in vacuum and inside a medium.

As it can be seen from (3.10), if $n_1 > n_2$, there is an incident angle θ_1^{cri} from which penetration is no longer possible. This occurs when $\theta_2 = \pi/2$ and hence the critical angle is

$$\theta_1^{\mathrm{cri}} = \arcsin\left(\frac{n_2}{n_1}\right), \tag{3.11}$$

which is utilized by the ray tracer to verify that penetration is possible.

In case that penetration is valid, the ray tracer executes the second step where it actually calculates the transmitted field. As in the reflection case, the field is obtained from the solution of the Maxwell equations constrained to the boundary conditions of the surfaces,

$$E_{\mathrm{trans}}(\mathbf{r}, f_{\mathrm{c}}) = E(\mathbf{r_0}, f_{\mathrm{c}})\frac{c}{4\pi f_{\mathrm{c}} s} e^{-j2\pi f_{\mathrm{c}} s/c} \mathbf{g_{Tx}}^{\dagger} \mathbf{T}_{\mathrm{in}}(f_{\mathrm{c}}) \mathbf{T}_{\mathrm{out}}(f_{\mathrm{c}}) \mathbf{g_{Rx}}, \tag{3.12}$$

where, again, $\mathbf{g_{Tx}}$ and $\mathbf{g_{Rx}}$ are the coefficients of the antenna gains and polarization at the transmitter and receiver, and s is the total length of the path. $\mathbf{T}_{\mathrm{in}}(f_{\mathrm{c}})$ and $\mathbf{T}_{\mathrm{out}}(f_{\mathrm{c}})$ are the dyadic penetration matrices at the input and output of the block [34].

3.7 Calculation of the transfer function and impulse response

So far, we have presented the propagation mechanisms included in the ray tracer. For each of the propagation paths, the ray tracer gives its delay τ_l and the amplitude of the path's field α_l when the electric field at the transmitter is $E(\mathbf{r}_0, f_{\mathrm{c}}) = 1$, i.e. the attenuation and rotation of the field throughout that path. From these variables, we can write the transfer function of the channel for a particular simulation as,

$$H(f) = \sum_{l=1}^{N_{\mathrm{l}}} \alpha_l e^{-j2\pi f \tau_l}, \tag{3.13}$$

where N_{l} is the total number of paths that result in a specific simulation. This function is continuous (up to the numerical precision of the computer), and a discrete expression can be found by sampling it,

$$H[q] = H(f_{\mathrm{c}} + q\Delta f), \tag{3.14}$$

such that the sample interval, Δf, coincide with the spacing between tones of the channel sounder (see Section 2.3.1).

Subsequently, the channel impulse response is found by implementing an inverse discrete Fourier transform,

$$h[n] = \mathcal{F}^{-1}\{H[q]W[q]\}. \tag{3.15}$$

[3] As before, these angles are defined between the propagation direction and the vector normal to the surface (see Fig. 3.1).

A Hann window W is utilized to reduce the effect of spectral leakage[4] that appears due to the discontinuities in the initial and final portions of the spectrum [43]. The Hann window is selected due to its good spectrum properties. For example, the difference between the first and second lobe of the spectrum of a Hann window is 31.5 dB, which results in much less leakage that the usual rectangular window (13.4 dB).

The ray tracer can also be used to study a time-variant channel. This is implemented by running several simulations of a specific scenario, such that some characteristics of the channel (e.g. the positions of some blocks) change from one simulation to the next one. If the simulations represent the state of the channel at fixed intervals T_{rep}, a time-variant transfer function and impulse response can be obtained,

$$\begin{aligned} H[m,q] &= H[mT_{\text{rep}}, f_{\text{c}} + q\Delta f), \\ h[m,n] &= \mathcal{F}^{-1}\{H[m,q]W[q]\}_q, \end{aligned} \tag{3.16}$$

where m is the simulation time index. As it will be presented later, each simulation of the scenario under test corresponds to a different position of the transmitter. Because the positions are equally spaced, the simulations correspond to a transmitter moving at a constant speed.

[4] In this case it is actually delay leakage.

Chapter 4

Modeling of a city intersection

In order to test the channel sounder implementation, a city intersection has been selected to run a short measurement campaign. The same intersection has been simulated with the ray tracer to check its performance in this kind of environments. So far, the AIT ray tracer has been utilized in indoor scenarios and vehicular tunnels and this is the first time we use it in a city intersection.

In Section 4.1, the selected intersection will be presented together with the characteristics of the simulation and the measurement campaign. The results from both tools are compared in terms of the local scattering function, whose calculations and derived functions are explained in Section 4.2. Finally, Section 4.3 presents a comparison of the results obtained with each tool.

4.1 Modeled scenario

The intersection of the streets *Sinagasse* and *Bellegardegasse*, located in the zone of *Kaisermühlen*, Vienna, Austria was the selected location to run the measurement campaign and the simulations with the ray tracer. It is illustrated in Fig. 4.1 and consists of buildings with two to six floors, some of them with several elements in their facades such as balconies and metallic frames. The intersection also has several cars on each side of the streets, that are usually located within fixed spaces. As it will be presented below, all these elements are important when modeling the ray-tracer simulation.

In the scenario, the receiver is always at a fixed location on the southwest street (see Fig. 4.2), while the transmitter follows one of three paths. In the first one, corresponding to the black squares in Fig. 4.2, it starts in the footpath in front of the transmitter and crosses the intersection towards northeast. In the second one, the transmitter crosses the intersection when moving from northwest to southeast as indicated by the green circles in the figure. In the third one, the transmitter starts in the northeast street and once it arrives to the corner, it turns right and moves to northwest (red diamonds). These three paths emulate a situation when an UAV flies around a static base station.

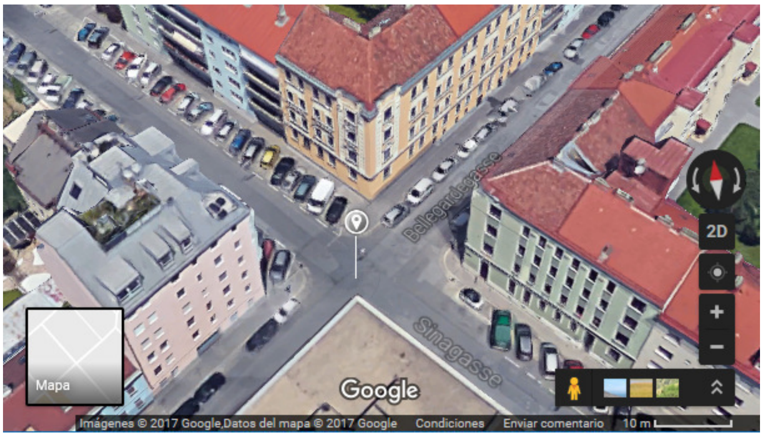

Figure 4.1: Selected scenario to run the measurement campaign and the simulation [44].

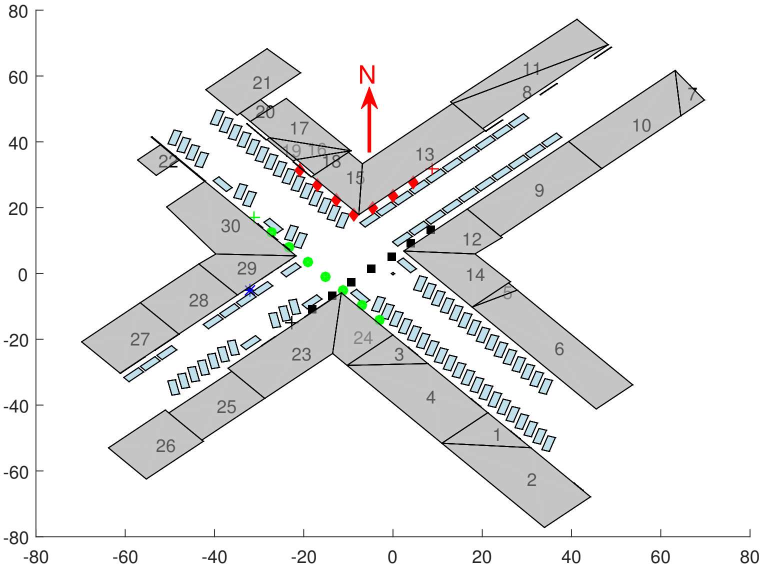

Figure 4.2: Scenario simulated with the ray tracer. The receiver is the blue asterisk on the southwest street, while the transmitter moves over the three paths that start with a cross.

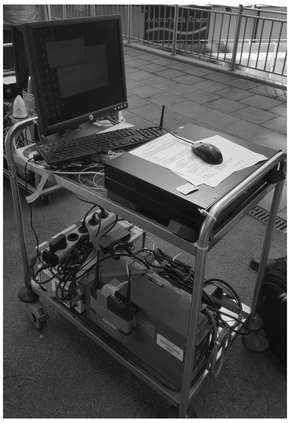

Figure 4.3: Overview of the equipment for the measurement campaign. Receiver.

4.1.1 Description of the measurement campaign

The measurement campaign was carried out on April 25th, 2017. The equipment of each terminal (Fig. 2.1) was arranged on trolleys to alloy the movement of the transmitter over the three paths described above. This set-up can be seen in Fig. 4.3. The antenna of the receiver was close to the trolley, at 0.85 m over the ground, while the transmitter antenna was 1.75 m over the ground to represent an UAV flying at low altitude.

For each of the paths followed by the transmitter, eight measurements were taken and a calibration was stored each second or fourth measurement. For each run, a video was recorded to qualitatively validate the results in the analysis. The calibration data served to consider possible variations in the RF chains (e.g. by temperature changes) and drifts in the GPSDO clocks. During the measurements, the GPSDOs were always disciplined by the GPS system and hence, they presented an accuracy of ±20 ns with the coordinated universal time (see Table. 2.2). The drift between the two clocks can be up to ±40 ns, that corresponds to a distance of ±12 m for the propagating wave. By running calibrations before each measurement, this error can be mitigated as both, the calibration and the measurement would include the same drift.

The sounding sequence followed the characteristics of Chapter 2, which are summarized again in Table 4.1.

4.1.2 Simulation with the ray tracer

The same paths followed by the measurement were implemented in the ray tracer. In fact, Fig. 4.2 corresponds to the blocks that are passed to the ray tracer as inputs. The big blocks with numbers represent the buildings in the intersection. Their dimensions were obtained from a public database of the city of Vienna [45] and were latter processed with the programs QGIS [46] and Blender$^{\mathrm{TM}}$ [47]. This procedure has been developed by David Löschenbrand at AIT and allows to obtain the geometry of the scenario in some simple steps.

Characteristic	Value
Frequency separation between tones Δf [KHz]	250
Number of tones N_q	201
Sounding bandwidth BW [MHz]	50.25
Minimum resolvable delay τ_min [ns]	19.9
Maximum resolvable delay τ_max [ns]	$3.98 \approx 4.0$
Sounding repetition T_rep [ms]	1
Maximum resolvable Doppler shift ν_max[Hz]	500

Table 4.1: Characteristics of the sounding signal.

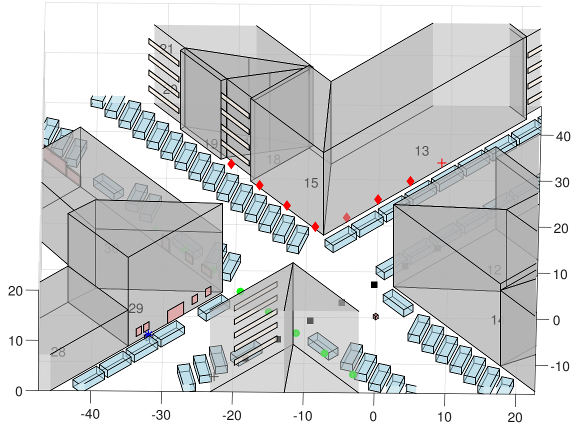

Figure 4.4: 3D view of the scenario simulated with the ray tracer. Observe the rectangles on the buildings, representing balconies, doors and fences.

Before building the model, we visited the environment in order to identify possible sources of strong reflections in the buildings' facades. Big metallic objects such as balconies, fences and doors were included as additional PEC blocks as can be seen in Fig. 4.4. In PEC materials, the incident waves are always reflected and there is not penetration. Those blocks are specifically modeled because they could originate strong reflections that could be misrepresented by considering them with the same material as the buildings. Other details were considered with the diffuse scattering mechanism, which used the directive model (see Section 3.5). We set the electric properties of the buildings as sandstone.

There are several cars in the scenario, which were modeled as rectangular PEC blocks of the same size. They were located within fixed positions, marked by lines on the street, which eased the modeling in the ray tracer. The videos of the measurement allowed to see if some spots were empty, which was not the case in most of the measurements. Additionally, the trolleys were also modeled as small PEC blocks as can be seen in the coordinate $(0, 0)$ of Fig. 4.4, where the transmitter trolley is depicted for clarification.

For each of the three paths of the transmitter, 15000 simulations were executed in a parallelized version of the code. Each of them corresponded to a different position of the transmitter, spaced 3 mm ($\sim 0.06\lambda$) from the previous one. Observe that contrary to the channel sounder, each snapshot corresponds to a different position instead of another time instant. Naturally, they can also be translated into time by assuming a constant speed of the transmitter, that in our case was 1.5 m/s, i.e. a typical walking speed. With these values, the repetition period of the ray tracer snapshots is $T_{\text{rep}} = 2$ ms, which gives a maximum resolvable Doppler shift of $\nu_{\max} = 250$ Hz. The bandwidth and frequency resolution of the simulation with the ray tracer coincided with the values of Table 4.1.

4.2 Calculation of the scattering function

The comparison between the channel-sounder results and the ray-tracer prediction is done in terms of the local scattering function. In the following, a brief introduction to this function will be presented as well as the procedure that we use to calculate it.

For a deterministic, linear, time-variant channel, the transfer function $H(t, f)$ is sufficient to determine the frequency content of the output signal when the channel is excited by an arbitrary input signal. In a stochastic channel, the best knowledge of the channel would be provided by a joint probability distribution of $H(t, f)$, which in practice is unfeasible to obtain [4]. A simpler approach consists of determining second-order functions, such as the autocorrelation functions,

$$\begin{aligned} R_{\text{H}}(t, t', f, f') &= \text{E}\{H^*(t, f)H(t', f')\}, \\ R_{\text{s}}(\nu, \nu', \tau, \tau') &= \text{E}\{s^*(\nu, \tau)s(\nu', \tau')\}, \end{aligned} \tag{4.1}$$

where $s(\nu, \tau)$ is the Doppler-variant impulse response and is equal to the double Fourier transform of $H(t, f)$ from time t to Doppler shift ν, and from frequency f to delay τ.

If the autocorrelation function of the channel does not depend on the specific values of t and t', but only on the difference $\Delta t = t' - t$, the channel is called wide sense stationary (WSS) [48]. Similarly, when it depends on the difference $\Delta f = f' - f$ and not on the specific values f and f' (stationary in frequency), it is said that the channel satisfies the uncorrelated-scatterers (UC) condition. Together, the WSS-US assumptions simplify the expressions of the channel autocorrelation functions [4],

$$\begin{aligned} R_{\text{H}}(t, t', f, f') &= R_{\text{H}}(\Delta t, \Delta f), \\ R_{\text{s}}(\nu, \nu', \tau, \tau') &= P_{\text{s}}(\nu, \tau)\delta(\nu' - \nu)\delta(\tau' - \tau), \end{aligned} \tag{4.2}$$

and introduce the so called scattering function, P_{s}, which conveniently contains the distribution of the channel gain along the delay τ and Doppler-shift ν domain,

$$P_{\text{s}}(\nu, \tau) = E\{|s(\nu, \tau)|^2\}. \tag{4.3}$$

In principle, any channel is stationary in the limit of very short times and very narrow bandwidths. However, when we consider long time intervals or large bandwidths, the channels (and thus the statistics) are not stationary. This scenario promotes the definition of stationary regions, in frequency and time, where the statistics of the channel are in fact stationary. For each region, a local Doppler-variant impulse response $s(t,f;\nu,\tau)$, and a local scattering function can be defined,

$$P_{\mathrm{s}}(t,f;\nu,\tau) = E\{|s(t,f;\nu,\tau)|^2\}, \tag{4.4}$$

where t and f are the centers in time and frequency of the stationary regions. For the scenarios under analysis, we have taken stationary regions of size 201 ms by 50.25 MHz based on studies performed in road crossings [49] [50]. In [49] the width of the stationary windows is determined from the collinearity (a spectral distance metric) of the local scattering function, while in [49] the validity of the WSS-US assumptions is studied with the spectral divergence.

The procedure to obtain the local scattering function from the channel sounder and the ray tracer data is the following. First, recall that both, the simulated data and the measured results can be expressed as a time-variant discrete transfer function,

$$H[m,q] = H(mT_{\mathrm{rep}}, f_{\mathrm{c}} + q\Delta f), \quad q = -\tfrac{N_{\mathrm{q}}-1}{2} \ldots \tfrac{N_{\mathrm{q}}-1}{2}, \quad m = 0 \ldots K_m, \tag{4.5}$$

where again, m is the index over each measurement or simulation and q is the index over the N_{q} tones. The local Doppler-variant impulse response can be obtained from (4.5) by moving a window function G_w over the domain of H and then taking a two dimensional discrete Fourier transform,

$$s^{G_w}[m,q;n,p] = \sum_{m'=-M/2}^{M/2} \sum_{q'=-N/2}^{N/2} H[m+m', q+q'] G_w[m',q'] e^{-j2\pi(pm'-nq')}. \tag{4.6}$$

This way, the Doppler impulse response in (4.6) is centered around the sample with index m in time and q in frequency, and contains M points in the Doppler domain (index p) and N in the delay domain (index n). Observe that by using stationary regions of 50.25 MHz in frequency, N coincides with the number of tones $N_{\mathrm{q}} = 201$ for both, the ray tracer and the channel sounder results. Moreover, only one window fits into the frequency domain and hence, the index q can be dropped. On the contrary, the value of M is not the same in each tool. For the channel sounder, $M = 201$ because $T_{\mathrm{rep}} = 1$ ms and the stationary window is 201 ms long. The ray tracer uses a repetition period $T_{\mathrm{rep}} = 2$ ms and 202 ms-long stationary windows, and hence $M = 101$.

With the local Doppler-variant impulse response, the local scattering function can be calculated as the expected value of the power of the former. The problem, however, is that we only have one realization of the measurements (or the simulations) which makes an ensemble average impossible. Instead, we utilize a multi-taper estimator [51] [52] based on discrete prolate spheroidal (DPS) sequences [53].

The multi-taper estimator consists of obtaining several versions of the local Doppler-variant impulse response by using orthogonal windows in their calculation. Thus, (4.6) corresponds to the impulse response when using the window G_w, for example. Because (4.6) contains two Fourier transforms, each window G_w is actually the product of two windows, one over the time domain and the other over frequency. If I windows are used in time and J in frequency, an estimate of the local scattering function would be,

$$\hat{P}_{\mathrm{s}}[m;n,p] = \frac{1}{IJ}\sum_{w=0}^{IJ-1} |s^{G_w}[m;n,p]|^2. \tag{4.7}$$

This procedure reduces the variance of the estimation compared to the case when only one window is utilized. The problem now consists of selecting an appropriate set of windows. In the dual domain of the Fourier transform, a good window decays fast, presents low power in its lobes and hence reduces the leakage of strong components across the spectrum. In principle, this could be achieved by selecting windows whose energy is mostly contained over certain region of the dual variable. For example, in the Fourier transform from time to Doppler domain, we would like to maximize the ratio,

$$\lambda(M,W) = \frac{\int_{-W}^{W} |G(\nu)|^2 d\nu}{\int_{-1/2}^{1/2} |G(\nu)|^2 d\nu}, \tag{4.8}$$

i.e. we would like the spectrum of the window G to be mostly contained within certain normalized bandwidth W. The solution to this optimization problem are the DPS sequences, which can be obtained from the solution of an eigenvalue equation [53]. Thus, we just define certain concentration region for each domain and obtain the corresponding DPS windows. As mentioned before, we then use them to calculate different versions of the Doppler-variant impulse response according to (4.6) and finally, we estimate the local scattering function with (4.7). This procedure is based on the descriptions in [49] and [54].

Because in our case the local scattering function depends on three variables, we build simpler functions that allows an easier comparison between the channel sounder and the ray tracer results. One of them is the channel-gain distribution over the delay, also called the (time-variant) power delay profile (PDP), which is obtained as a sum of the local scattering function along the Doppler indexes,

$$\hat{P}_{\mathrm{h}}[m;n] = \sum_{p} \hat{P}_{\mathrm{s}}[m;n,p]\Delta\nu. \tag{4.9}$$

Similarly, the time-variant Doppler spectral density (DSD), i.e. the gain distribution over the Doppler shift, is obtained from the integral of the local scattering function along the delay indexes,

$$\hat{P}_{\mathrm{B}}[m;p] = \sum_{n} \hat{P}_{\mathrm{s}}[m;n,p]\Delta\tau. \tag{4.10}$$

These functions are just projections on the delay or Doppler axis, respectively. Observe that meanwhile the local scattering function has units [$\mathrm{Hz}^{-1}\,\mathrm{s}^{-1}$], the PDP has units [$\mathrm{s}^{-1}$] and the DSD [$\mathrm{Hz}^{-1}$]. From the last two, we calculate condensed parameters that help to characterize the channel. We utilize the following functions, defined in [4].

1. The zeroth order moment of each function. In both cases, the PDP and DSD, this function is equal to the time-variant gain of the channel,

$$\begin{aligned} P_{\mathrm{m}}[m] &= \sum_{n} \hat{P}_{\mathrm{h}}[m;n]\Delta\tau, \\ P_{\mathrm{m}}[m] &= \sum_{p} \hat{P}_{\mathrm{B}}[m;p]\Delta\tau. \end{aligned} \tag{4.11}$$

2. The normalized-first order moment, which in the case of the PDP, it is called mean delay,

$$T_{\mathrm{m}}[m] = \frac{\sum_n \hat{P}_{\mathrm{h}}[m;n]\tau_n \Delta\tau}{P_{\mathrm{m}}}. \tag{4.12}$$

   In the DSD, it is known as mean Doppler shift,

$$\nu_{\mathrm{m}}[m] = \frac{\sum_p \hat{P}_{\mathrm{B}}[m;p]\nu_p \Delta\nu}{P_{\mathrm{m}}}. \tag{4.13}$$

3. The normalized second central moment, or root-mean-square (rms) spread. It is called rms delay spread for the PDP case,

$$S_{\tau}[m] = \sqrt{\frac{\sum_n \hat{P}_{\mathrm{h}}[m;n]\tau_n^2 \Delta\tau}{P_{\mathrm{m}}} - T_{\mathrm{m}}^2[m]}, \tag{4.14}$$

   and rms Doppler spread in the DSD,

$$S_{\nu}[m] = \sqrt{\frac{\sum_p \hat{P}_{\mathrm{B}}[m;p]\nu_p^2 \Delta\nu}{P_{\mathrm{m}}} - \nu_{\mathrm{m}}^2[m]}. \tag{4.15}$$

In a strict sense, we do not use the local scattering function to compare the channel sounder results and the ray tracer predictions. Instead, the comparison is implemented in terms of the derived functions (4.9 - 4.15). This is the topic of the next Section.

4.3 Comparison of the results

As it was mentioned above, the predictions of the ray tracer are now compared against the measurements taken with the channel sounder. Before showing the results, it is worth mentioning some comments about them. There is a comparison for each path taken by the transmitter, that from now on will be called scenarios. Because eight measurements (runs) were taken for each scenario, the one closest to a calibration that did not present any warning when taking the data is presented in the comparison.

As the simulation with the ray tracer is noiseless, artificial additive white Gaussian noise was generated with the same power than the noise from the measurements. Additionally, to compensate the fact that the channel gain with the ray tracer was always below the measurements, a *calibration* was implemented such that the mean gain from the simulations matched the gain from the measurements.

Finally, the time axes in the figures of this Section are in general not the same in the measurements than in the simulations. This is because the former were executed based on fixed positions of the transmitter, while the latter were taken at fixed intervals in time. As it was mentioned before, the positions in the ray tracer are converted into time by considering a constant speed of the transmitter.

4.3.1 First scenario

Analysis of the delay

In the first scenario, the transmitter is moving on the footpath in front of the receiver, crossing the intersection from southwest to northeast, as shown in Fig. 4.2. The corresponding PDP obtained from the measurements and the simulations are presented in Figs. 4.5 and 4.6, respectively. In both cases, a strong cluster, marked with number 1, can be distinguished. The cluster is ~3-tap wide (~ 60 ns), and increases its delay time as the transmitter moves away from the receiver.

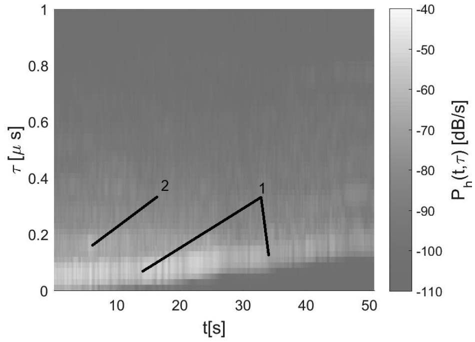

Figure 4.5: Measured power delay profile of the first scenario.

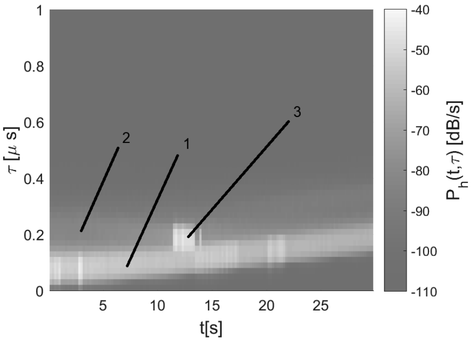

Figure 4.6: Simulated power delay profile of the first scenario.

A second cluster, marked with number 2 in the figures, arrives after the first one between 140 ns and 200 ns, which corresponds to multi-path components (MPC) that travel distances around 42 m and 60 m. Notice that contrary to the first cluster, this one does not seem to increase the arrival time as the transmitter advances. Checking the results of the simulation, the second cluster mainly arises from diffuse scattering with the surrounding buildings, except around $t \sim 12$ s (see Fig. 4.7, when strong reflected components appear from the vehicles in the north corner (number 3 in Fig. 4.6).

As it can be seen in the figures, the difference between the second-cluster power and the first one is larger in the simulation than in the measurements. This would indicate that the total power on those long[1] diffuse-scattered paths is not correctly modeled, possibly because each of them has low power, there are a reduced number of diffuse tiles, or the components in the first cluster are very powerful.

[1] By long we mean paths whose total length is larger than those of the first cluster

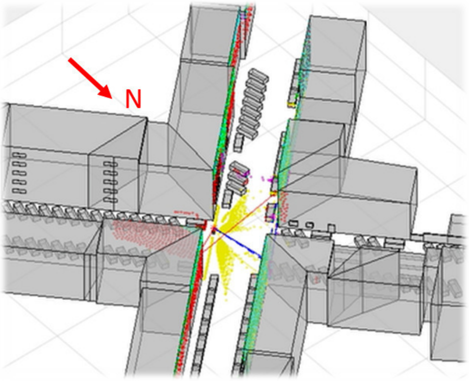

Figure 4.7: Simulated MPCs in the first scenario at $t = 12.2$ s. Each point represents a scattering source, while the thick lines connecting the transmitter (red star) and receiver corresponds to LOS or pure reflection. Notice the strong reflection with the cars at the north corner.

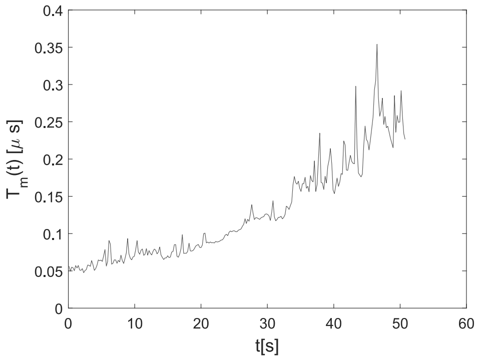

Figure 4.8: Measured mean delay of the first scenario.

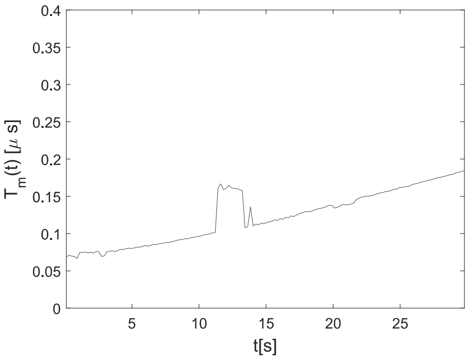

Figure 4.9: Simulated mean delay of the first scenario.

The problem with long rays is also observed in the tail of the PDP, which decays faster in the simulation than in the measurements. A bigger tail also implies a larger mean delay as it can be seen in Figs. 4.8 and 4.9, which confirms the absence of late MPCs in the ray tracer. This deficiency could be caused by the restriction of reflections up to second order (due to the simulation complexity) that limits both, specular and scattering reflections. Moreover, notice that in contrast to the simulations, the mean delay of the measurements is not smooth, specially when the transmitter is far away from the receiver. The predictions of long MPCs with the ray tracer should not only be improved in their power, but also in their random behavior.

Figs. 4.10 and 4.11 present the rms delay spread of this scenario. As in the case of the mean delay, there is a significant difference in the function amplitude and smoothness. The large spread on the measurements corresponds with the large tail of the PDP and the existence of several MPCs much later than the main

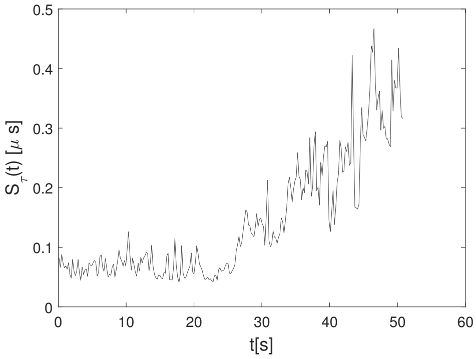

Figure 4.10: Measured rms delay spread of the first scenario

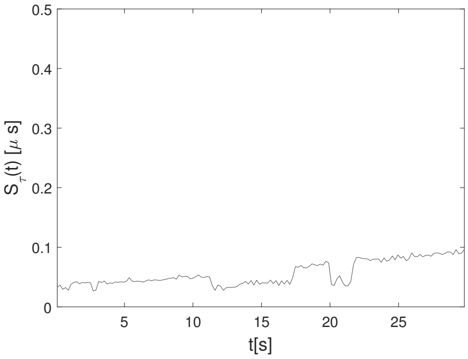

Figure 4.11: Simulated rms delay spread of the first scenario

cluster. This is not the case of the simulated data, where the spread of the delay is basically the width of the main cluster. Interestingly, the existence of reflected components around $\sim$ 12 s result in a smaller Doppler spread than without them. Because the power of those components is $\sim$ 12 dB larger than the MPCs of the first cluster, they dominate in the calculation of the delay spread and hence, the spread around $\sim$ 12 s is basically the spread of those reflected components, which is smaller than the one of the first cluster.

The differences in the rms delay spread are summarized in Table 4.2, where its mean value over time is presented before and after the transmitter crosses the intersection (case a) and b) respectively). In both cases, the measured delay spread is larger than the simulated case. Such a difference is significantly larger after the transmitter has crossed the intersection (the mean S_τ in the measurements is more than three times in the simulations), where multiple reflections (not-predicted by the ray tracer) participate in the wave propagation.

Case	**Delay spread** [ns]		**Doppler spread** [Hz]	
	Measured $\mu;\sigma$	Simulated $\mu;\sigma$	Measured $\mu;\sigma$	Simulated $\mu;\sigma$
a)	66.1; 16.2	41.8; 6.9	22.63; 5.19	115.78; 32.94
b)	218.6; 93.3	65.7; 21.2	36.49; 12.59	100.09; 26.49

Table 4.2: Mean values and standard deviations of the rms spreads in the first scenario when the transmitter is a) before and b) after the intersection.

Analysis of the Doppler shift

In case of pure free space propagation, the Doppler shift at the frequency f_c due to movement of the transmitter in this scenario would be given by,

$$\nu(t) = -\frac{v^2 t f_\mathrm{c}}{c\sqrt{(vt)^2 + d_\mathrm{min}}}, \tag{4.16}$$

where v is the speed of the transmitter (1.5 m/s in the ray tracer) and d_{min} is the closest distance between the receiver and transmitter. This function is depicted as the upper dotted line in Fig. 4.13 where the simulated DSD of the first scenario is presented. As it can be seen, most of the components (marked with number 1) do not follow (4.16) but instead an scaled version of it. This behavior suggests that their paths contain at least one reflection. The dashed lines in the figure correspond to twice the maximum Doppler shift due to the transmitter speed, and as it can be seen, no components exceed them.

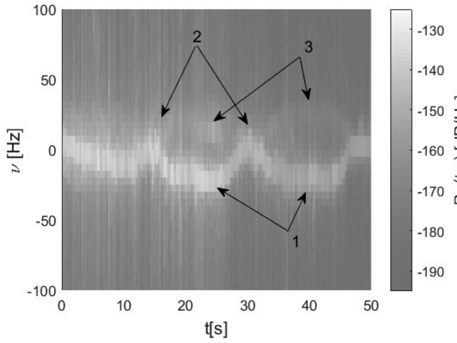

Figure 4.12: Measured Doppler spectral density of the first scenario.

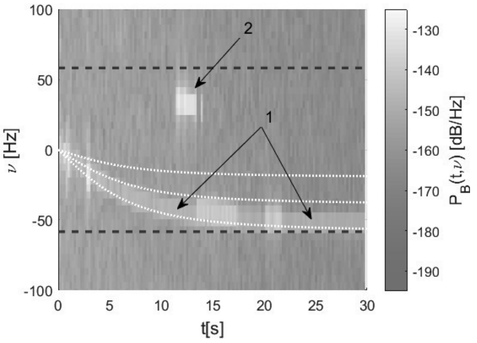

Figure 4.13: Simulated Doppler spectral density of the first scenario.

As comparison, Fig. 4.12 shows the DSD obtained from the measurement campaign. Like in the ray tracer, a large portion (marked with number 1) of the MPCs present a negative Doppler and follow a shape similar to (4.16). When taking the measurements, we had to stop at the corners of the intersection to lift and lower the trolley with the transmitter equipment. This effect can be distinguished in Fig. 4.12 in the sections marked with number 2, where the Doppler shift goes to zero.

Additionally, the measurements present some positive Doppler components that *mimic* the negative ones. They correspond to paths that initially propagate in the movement direction of the transmitter and then are reflected back to the receiver. These components cannot be seen in the simulation except around the second marker of Fig. 4.13. Studying the data of the simulation, this cluster corresponds to the strong reflection with vehicles at the north corner in the scenario (Recall marker 3 in Fig. 4.6). Therefore, the absence of other components with positive Doppler shift might be also related to the poor prediction of long rays.

As it will be more evident when studying scenarios 2 and 3, this problem is also a consequence of the way that diffuse scattering components are modeled by the ray tracer. Recall from Section 3.5 that the diffuse tiles are randomly generated in each simulation. This implies that all of the scattered MPCs created in one simulation are discarded for the next one and a new random set is generated again [5] [29]. As a result, it is impossible to track scattered components in the Doppler shift and their aggregate effect is white noise. The latter is also the reason why the noise floor in the simulation is higher than in the measurement (around 20 dB or more).

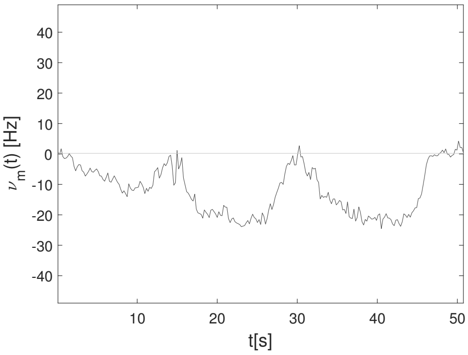

Figure 4.14: Measured mean Doppler shift of the first scenario.

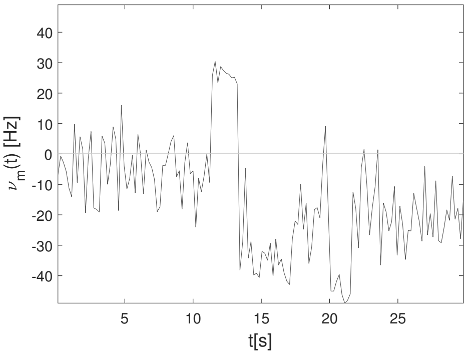

Figure 4.15: Simulated mean Doppler shift of the first scenario.

From Figs. 4.12 and 4.13, it is clear that the Doppler shift in the simulations is larger than in the measurement. The reason lies in the fact that the simulations were run at a constant speed of 1.5 m/s, while the measurements were slower (around 1.2 m/s according to the video we took). This effect is evidenced in the mean Doppler shift, illustrated in Figs. 4.14 and 4.15, where the measured case has smaller values than those well-defined regions of the simulations (e.g. $\sim$ 12 s or $\sim$ 15 s). Additionally, it can also be seen when the mean Doppler goes to zero in the measured data, corresponding to the instants when the trolley stopped at the corners.

The simulated mean Doppler shift also fluctuates more in the simulations than in the measurements and there are some intervals when it is actually positive. This effect is closely related with the inability of the ray tracer to track the Doppler shift from scattered MPCs. The random and noisy effect of those components deviate the mean Doppler from the shift produced by other propagation mechanisms. This occurs in most of the cases, except when the other MPCs are very powerful, like in the case of the strong reflection around 12 s. This problem is also reflected in the rms Doppler spread of Figs. 4.16 and 4.17, which is in general larger for the simulated case (except again at 12 s).

Like in the rms delay spread, the mean value of the rms Doppler spread before and after the transmitter crosses the intersection is presented in Table 4.2. As expected from Figs. 4.16 and 4.17, the Doppler spread is larger in the simulations than in the measurements. In the latter case, the spread is larger when the transmitter is far away from the receiver as the effect of large-reflection-order paths is more significant.

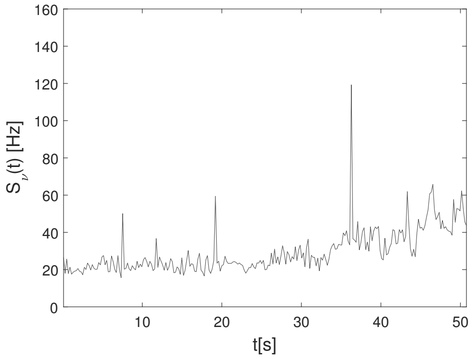

Figure 4.16: Measured rms Doppler spread of the first scenario.

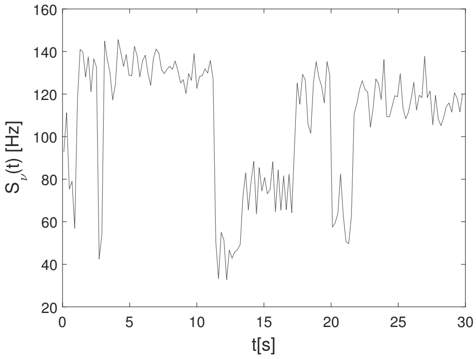

Figure 4.17: Simulated rms Doppler spread of the first scenario.

4.3.2 Second scenario

Analysis of the delay

The transmitter crosses the intersection from northwest to southeast during the second scenario. It starts in a highly obstructed zone, then moves on the less blocked region of the intersection, and finishes obstructed again. This transition is reflected in the PDP of the scenario where the power changes accordingly. In both, the measurements and simulations, the power is significantly larger during the intersection than when the transmitter is blocked by the buildings, as can be verified in Figs. 4.18 and 4.19.

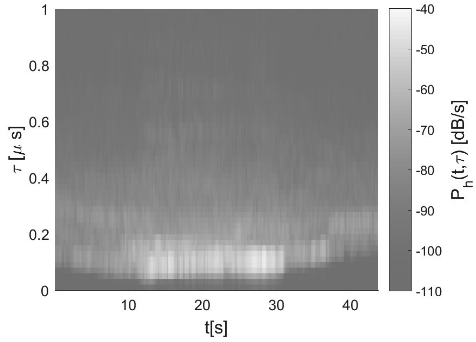

Figure 4.18: Measured power delay profile of the second scenario.

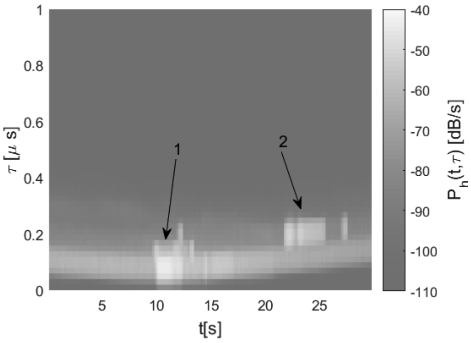

Figure 4.19: Simulated power delay profile of the second scenario.

In both cases, one can identify a large cluster whose delay time reduces as the transmitter approaches to the corner and then increases when it moves away from the receiver. In the simulation, this cluster is made of rays resulting from all propagation mechanisms, which suggests that their prediction is accurate (they match the measurement results). Observe that after the cluster arrival, the PDP of the measurements is richer than the PDP of the simulations. In the measurements, some components can be distinguished even after $\tau = 0.6$ µs, while already at 0.4 µs the decay of the simulated PDP is determined by the window used in the Fourier transform. Again, this difference shows the reduced number of long rays that the ray tracer can simulate.

Some strong components in the ray tracer appear around $t = 11$ s and 23 s. The first, marked with number 1 in Fig. 4.19, arises when LOS and reflected rays appear between the two terminals once the transmitter reaches the corner. They disappear a few seconds later when the transmitter is hidden behind the cars in the scenario. After being blocked by the buildings in the second corner, the components of the second marker in Fig. 4.19 appear due to reflections with cars at the north corner. Some snapshots of the simulation, where those two clusters appear, are shown in Fig. 4.20.

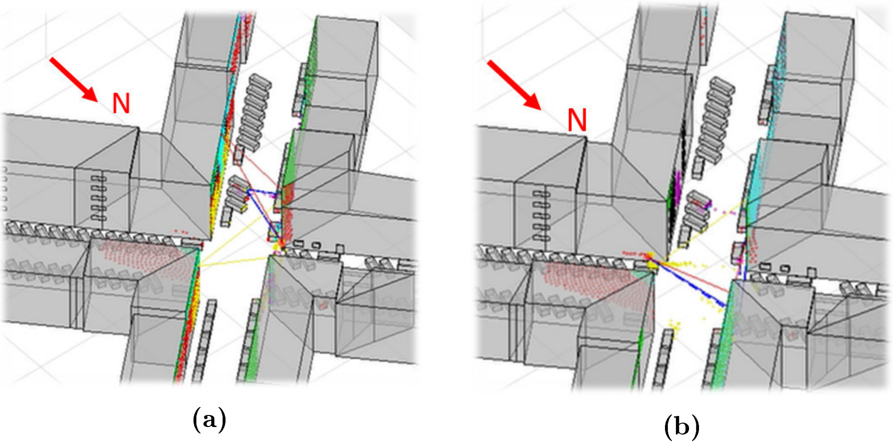

Figure 4.20: MPCs of the second scenario at (a) 9.8 s and (b) at 22.4 s. Observe the reflected rays connecting the trasnmitter and receiver.

Figs. 4.21 and 4.22 present the mean delay of the scenario for the measurements and the simulations, respectively. As in the case of the first scenario, the mean delay is in general larger in the measurements than in the simulations, which as before, is a consequence of the poor prediction of long MPCs in the ray tracer. This difference is specially noticeable when the transmitter is obstructed by the buildings and the measured delay is much larger than in the intersection. During those regions, the main propagation mechanism in the ray tracer is diffuse scattering, and hence, the modeling of this mechanism should be revised to improve its performance. On the other hand, the ray tracer predicts a very smooth delay function, which again is not the case in reality. Nevertheless, the mean delay of both tools is larger than the time that a LOS ray would take, which shows that in spite of the fact that the ray tracer does not model all the rays of the scenario, the ones that are simulated present a reasonable delay.

Finally, the rms delay spread is illustrated in Figs. 4.23 and 4.24. As expected, the same behavior of the first scenario is obtained: the ray tracer predicts a smooth

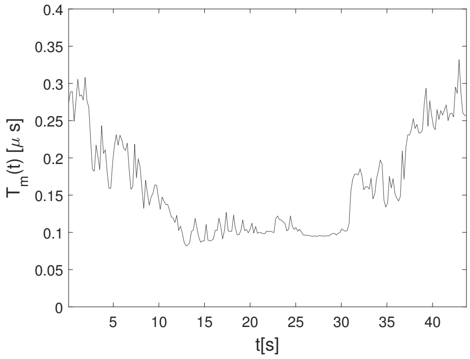

Figure 4.21: Measured mean delay of the second scenario.

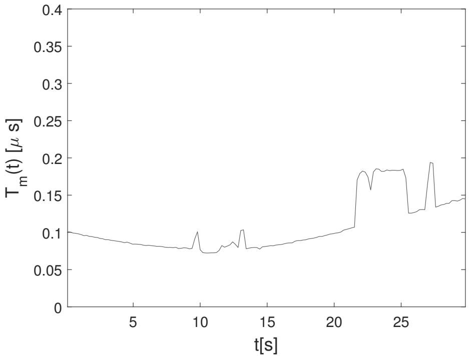

Figure 4.22: Simulated mean delay of the second scenario.

and reduced spread, which does not coincide with the fluctuating function of the measurements. Observe from Fig. 4.23 that when the transmitter is obstructed, the delay spread is in general proportional to the distance of the transmitter to the corner. Again, this could be a consequence of the relevance that multi-reflected paths gain in the street canyon. This is confirmed in Table 4.3 that presents the mean values of the rms delay spread before a), during b) and after c) the intersection. Naturally, the mean value obtained from the measurements is larger than the one calculated from the simulations, specially in cases a) and c), where the transmitter is obstructed and long paths are significant in the wave propagation.

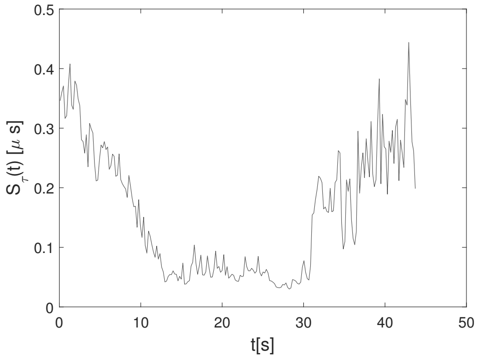

Figure 4.23: Measured rms delay spread of the second scenario.

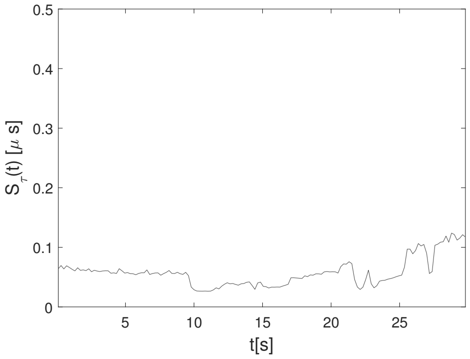

Figure 4.24: Simulated rms delay spread of the second scenario.

Analysis of the Doppler shift

The DSD of the second scenario presents significant differences between the measured data and the simulations. The DSD calculated from the measurements is presented in Fig. 4.25. When the transmitter is approaching to the corner, most of the components have a positive shift because it is getting closer to the receiver

Case	**Delay spread** [ns]		**Doppler spread** [Hz]	
	Measured $\mu; \sigma$	Simulated $\mu; \sigma$	Measured $\mu; \sigma$	Simulated $\mu; \sigma$
a)	255.6; 73.3	57.5; 8.0	40.29; 8.67	134.47; 24.89
b)	60.2; 22.0	40.8; 10.4	22.11; 3.49	93.36; 36.15
c)	228.8; 70.2	74.8; 30.3	38.38; 10.14	105.82; 44.77

Table 4.3: Mean values and standard deviations of the rms spreads in the second scenario when the transmitter moves a) before, b) during and b) after the intersection.

(mark 1 in Fig. 4.25). Some other components present a negative shift in this region and correspond to reflections whose path lengths are increasing. A similar behavior is observed in the third region of the plot, when the transmitter has already crossed the intersection. In this case, however, most of the power corresponds to components with negative shifts because the transmitter is moving away from the receiver. In the region of the intersection (mark 2 in Fig. 4.25), the Doppler shift is not clearly positive or negative but instead, the power spreads on both sides of the axis. This means that the propagation of the wave uses different types of paths, some of them increase their length and some others reduce it.

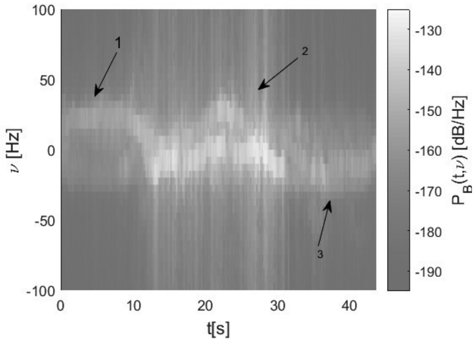

Figure 4.25: Measured Doppler spectral density of the second scenario.

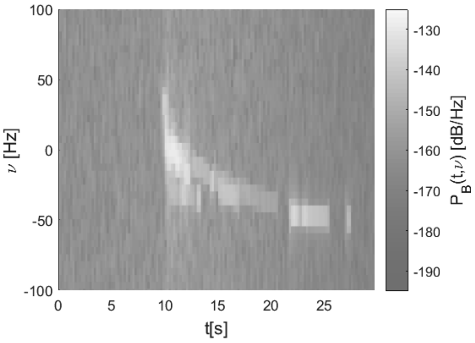

Figure 4.26: Simulated Doppler spectral density of the second scenario.

On the other hand, the DSD obtained from the simulations is very poor. In the first region, it only has white noise and no Doppler shift can be clearly distinguished. As mentioned before, this region is mostly modeled by diffuse scattering and a single diffracted ray from the corner of the building, which anyway does not have too much power as the bending angle is almost 90°. Therefore, we have here the same issue that was described in the first scenario regarding the impossibility of modeling the Doppler shift with the ray tracer. The random generation of diffuse tiles from one simulation snapshot to the next one makes impossible to track any Doppler component.

Once the transmitter enters the intersection, other propagation mechanisms appear in the scene and a clear Doppler shift becomes evident. However, the DSD in this region is still misrepresented by the ray tracer when compared to the measurements. In fact, only negative components are observed and reflected (or scattered) paths whose length diminish are not predicted. The powerful components studied in the PDP of Fig. 4.19 present a clear Doppler shift in Fig. 4.26. This confirms that LOS and pure reflected components are correctly modeled by the ray tracer in both, the PDP and DSD.

This inaccuracy in the prediction of the Doppler shift can also be seen in the mean Doppler function of Fig. 4.28. Contrary to the measured data of Fig. 4.27, the simulated Doppler shift do not follow a tendency but presents random and large variations (except when strong reflection components exist). On the other hand, the mean Doppler shift of the measurements are coherent with the movement of the transmitter. Before the corner, the shift is positive as the transmitter approaches to the receiver and many MPCs reduce their length. Once it reaches the corner, the shift is suddenly negative as the transmitter moves away from the receiver. Then, during the intersection, many MPCs with different Doppler shifts participate in the propagation and the sign of the mean shift depends on which components dominate. After the second corner, the Doppler shift is then negative as the transmitter moves away from the receiver.

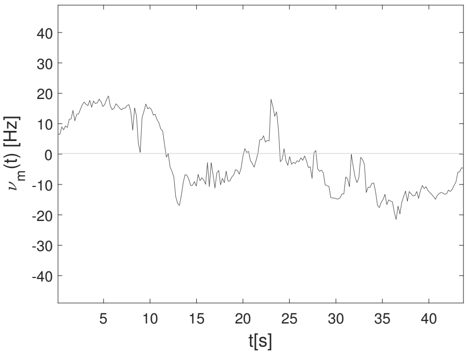

Figure 4.27: Measured mean Doppler shift of the second scenario.

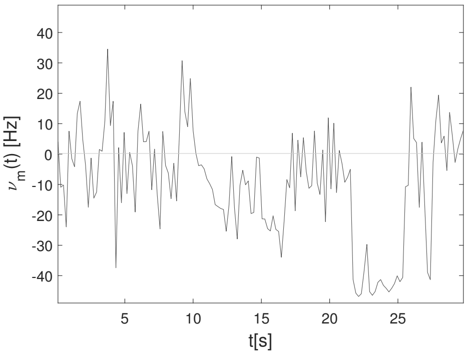

Figure 4.28: Simulated mean Doppler shift of the second scenario.

Regarding the rms Doppler spread, the simulation results present large values (see Fig.4.30), as expected from the uniform distribution of the power in the DSD. In the measurements, the spread is lower and increases in the beginning and in the end. On the extremes of the measurement, the arriving MPCs have gone through a different number of reflections and hence, their paths change at different speeds. The variance of the Doppler is then larger than when the propagation is mostly dominated by paths with a reduced number of reflections. Table 4.3 corroborates these observations as the mean value of the measured Doppler spread when the transmitter is obstructed (regions a) and c)) is almost twice as much as in region b). The mean value of the rms Doppler spread is larger in the simulations as a result of the uniform DSD.

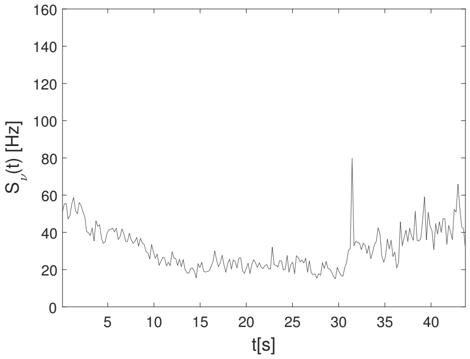

Figure 4.29: Measured rms Doppler spread of the second scenario.

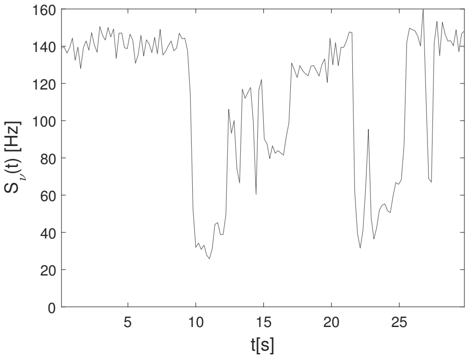

Figure 4.30: Simulated rms Doppler spread of the second scenario.

4.3.3 Third scenario

Analysis of the delay

In the third scenario, the transmitter is initially located in the same street of the receiver but on the other side of the intersection. It moves towards the receiver until reaching the corner, when it turns right and moves to northwest. Therefore, the expected behaviour of the delay consists of a reduction during the first part of the scenario and an increase after the transmitter reaches the corner. In fact, the PDP of the measurements and the simulations correspond to those expectations as illustrated in Figs. 4.31 and 4.32, where a large cluster is observed following the described behavior (mark 1 in Fig. 4.31).

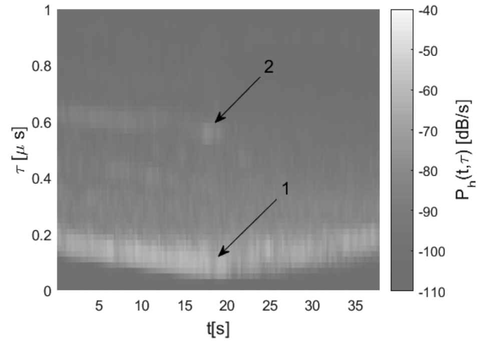

Figure 4.31: Measured power delay profile of the third scenario.

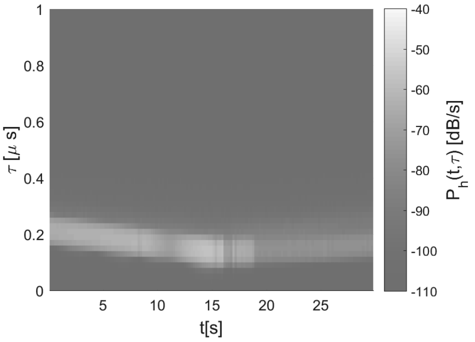

Figure 4.32: Simulated power delay profile of the third scenario.

Interestingly, the power variation of that cluster before and after the corner is less significant in the measurements than in the simulations. In the former, the power drops around 6 dB, while it reduces by around 10 dB in the latter. Again, this suggests a deficiency in the power of diffuse-scattered MPCs, which are the main propagation mechanism when the transmitter is obstructed. As before, this could also be caused by the limitation on the maximum reflection order.

Like in previous scenarios, the decay of the PDP is slower in the measurements than in the simulations. Additionally, a second cluster is distinguished around the marker 2 of Fig. 4.31, which is absent in the simulations. In spite of those differences, the mean delay of both cases follow the same tendency as illustrated in Figs. 4.33 and 4.34. In the measurement, however, the function presents some peaks, possibly due to the random arrival of some late components, while it is smooth in the ray tracer.

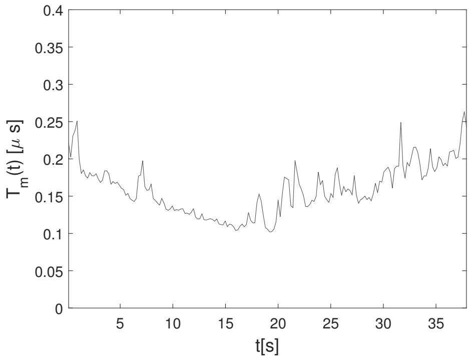

Figure 4.33: Measured mean delay of the third scenario.

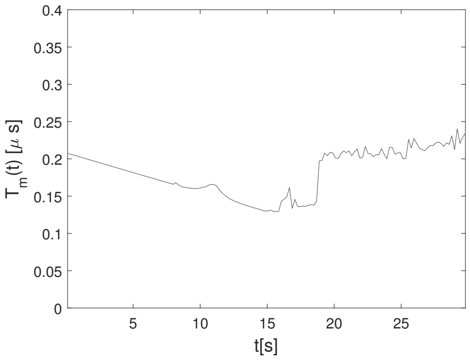

Figure 4.34: Simulated mean delay of the third scenario.

The arrival of late components in the measurements is evidenced with sudden peaks in the rms delay spread of Fig. 4.35. Some of those peaks match the increase of the mean delay mentioned before (c.f. Fig. 4.33). On the other hand, the smother rms delay spread of the simulations is clearly different when there is LOS than when the transmitter is obstructed by the building. The huge gap between the two cases might be caused by the propagation mechanisms. When the LOS is obstructed, the propagation is mainly ruled by scattering and diffraction. The rays produced by those mechanisms present different lengths, which results in a larger delay spread. On the contrary, the first part of the propagation is ruled by the LOS case, which is responsible for most of the power. The power spread around this component is naturally smaller.

The large increase of the delay spread during the second part of the simulation is strange when comparing with the second scenario (see Fig. 4.24). The beginning of the second scenario is similar to the end of the third one because in both situations, the transmitter is blocked by the same building (on different sides of the street, though). The spread should be somehow similar because the same propagation mechanisms, scattering and diffraction, are dominant in both scenarios. One possible reason of such a difference might be the acute angle between the building facade and the transmitter, which is larger in the third scenario than in the second one. This might allow a larger number of MPCs in the former. Some simulations over the southeast corner can be executed to check these ideas and verify whether the behavior is the same.

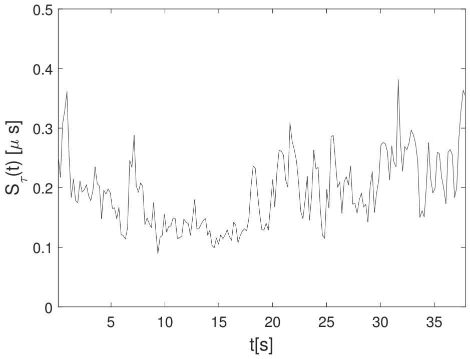

Figure 4.35: Measured rms delay spread of the third scenario.

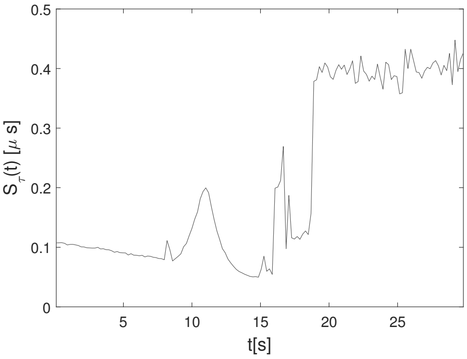

Figure 4.36: Simulated rms delay spread of the third scenario.

Table 4.4 presents the mean values of the rms spreads when the transmitter is a) before and b) after the intersection. As observed in the second scenario, the delay spread is larger when the transmitter is obstructed than in the LOS case. However, meanwhile the increase in the measurements lies around 33%, the mean delay spread increases by 235% in the simulations. As mentioned above, this behaviour is anomalous when comparing with the results of the second scenario.

Case	**Delay spread** [ns]		**Doppler spread** [Hz]	
	Measured $\mu; \sigma$	Simulated $\mu; \sigma$	Measured $\mu; \sigma$	Simulated $\mu; \sigma$
a)	164.6; 51.8	98.1; 32.2	29.42; 6.03	38.64; 14.98
b)	220.2; 55.8	328.8; 121.9	35.90; 6.88	118.15; 42.90

Table 4.4: Mean values and standard deviations of the rms spreads in the third scenario when the transmitter a) moves towards and b) from the corner.

Analysis of the Doppler shift

The DSD of the scenario is plotted in Fig. 4.37 for the measurements and Fig. 4.38 for the simulations. In both cases, there is a clear positive shift when the transmitter is approaching to the corner. The value of the shift is different as the speed of the transmitter in the simulation differs from the speed in the measurement. Once the transmitter turns around the corner, the shift becomes negative as most of the rays increase in length. However, this effect is only seen in the measurements because, as before, the model we use for diffuse scattering does not allow to predict Doppler shifts. The same happens with the *mirror* components, that have negative shifts in the beginning and positive in the end. They can be clearly distinguished in the measurements of the channel sounder but are absent in the ray tracer predictions.

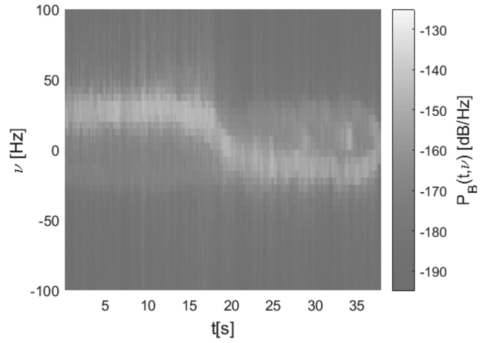

Figure 4.37: Measured Doppler spectral density of the third scenario.

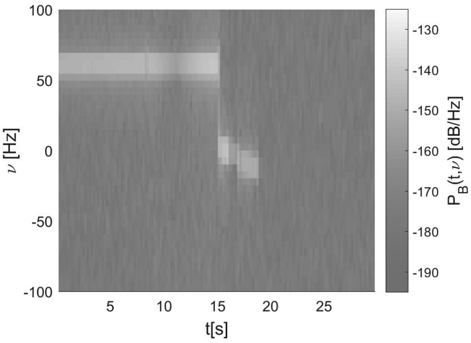

Figure 4.38: Simulated Doppler spectral density of the third scenario.

The mean Doppler shift of the scenario is depicted in Figs. 4.39 and 4.40. In the simulations, the Doppler shift is clearly positive before the corner and then it is just random noise with zero mean. In the measurements, the mean shift is always coherent with the movement of the transmitter. Observe that before the corner, the magnitude of the shift is larger because the movement is parallel to the propagation of the LOS component. After turning around the corner, the movement is not longer parallel to the rays' propagation direction and hence, the shift is smaller.

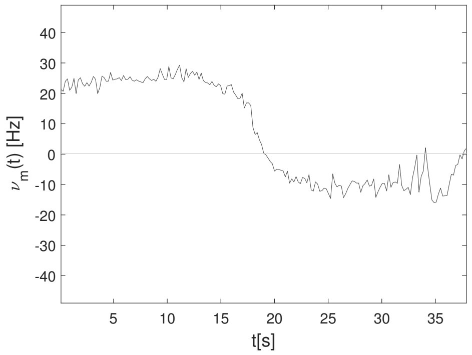

Figure 4.39: Measured mean Doppler shift of the third scenario.

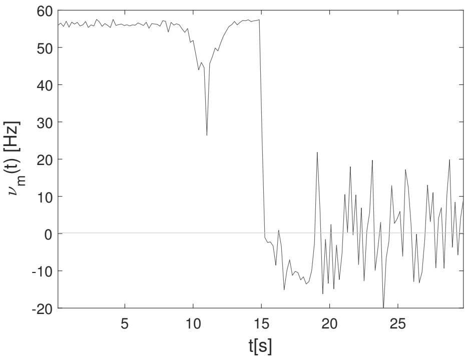

Figure 4.40: Simulated mean Doppler shift of the third scenario.

Figs. 4.41 and 4.42 present the Doppler spread of the third scenario for the measurements and the simulations, respectively. In the measurements, the Doppler spread seems to be constant, meaning that all the components are similarly spread during the entire run.

In the simulations, the region whose DSD did not show any defined shift has a very large rms Doppler spread, that does not provide any information of the envi-

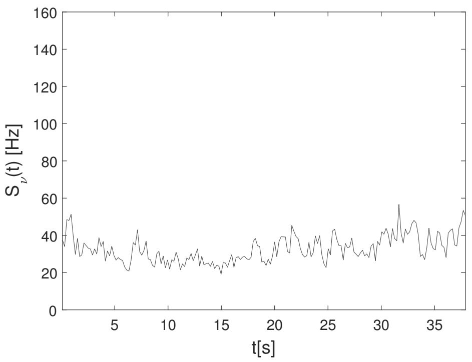

Figure 4.41: Measured rms Doppler spread of the third scenario.

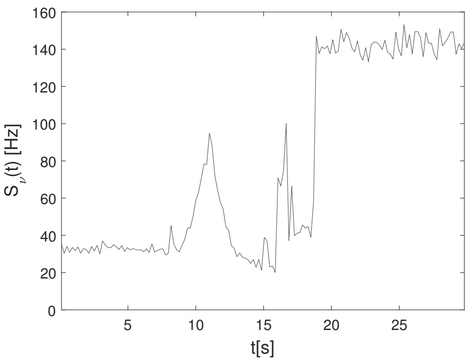

Figure 4.42: Simulated rms Doppler spread of the third scenario.

ronment. Actually, the observed value in the second part of Fig. 4.42 corresponds to the standard deviation of a uniform distribution between $[-250, 250]$ Hz, that is the range of the Doppler spectrum in the simulation,

$$\sigma_{\mathrm{u}} = \frac{\nu_{\max} - \nu_{\min}}{2\sqrt{3}} \approx 144.3. \tag{4.17}$$

Observe that this is the same value of the rms Doppler spread of the second scenario (see Fig. 4.30) when no components can be distinguished in the corresponding DSD. For the third scenario, this observation can be confirmed in Table 4.4, where the simulated mean Doppler spread after the corner is 118.5 Hz. It is not exactly 144.3 because immediately after the transmitter turns around the corner, there are still some MPCs resulting from non-scattering propagation mechanisms. In the measurements, the results coincide with the previous observations: the mean Doppler spread is larger when the transmitter is obstructed.

Lastly, it is worth mentioning the existence of a peak around 11 s in the first and second moments of the PDP and DSD. Those peaks coincide with a reduction of the signal power due to a destructive interference between the LOS component and some scattered MPCs, which makes visible other components with less power. The remaining power of the signal has naturally a larger spread in the delay domain. In the Doppler, the power approaches to the white noise of the diffuse scattering and hence, the mean shift gets closer to zero, while the rms spread approaches to (4.17), (see Figs. 4.40 and 4.42).

Chapter 5

Conclusions and Future Work

5.1 Conclusions

One of the most important contributions of this thesis has been the upgrade of the channel sounder developed by Wang [6] *et al.* at USC from sounding 15 MHz bandwidth to 50 MHz. To achieve this objective, we characterized the initial sounder and implemented its functionalities in a SDR with better capabilities. Our work in this field can be summarized in three points:

1. The original code running on the FPGA was re-written to satisfy the tighter timing requirements of the new hardware. We utilized pipeline execution to include the extra functionalities that the sounding program has with respect to the LabVIEW streaming template [10]. The additional functionalities are: the synchronization with the GPSDO's 1-PPS signal, and the modification of the streaming timing structure by means of two counters.

2. We updated the code on the host PC to a newer version of LabVIEW. Additionally, certain sections of the code, such as the automatic gain control, were modified to improve their speed.

3. To measure a larger bandwidth, we constructed new sounding sequences. We utilize multi-tone signals whose crest factor has been maximized with the Friese algorithm [24]. Additionally, the sounding scheme has been modified to obtain a larger signal to noise ratio.

The channel sounder has been utilized to measure the characteristics of a wireless communication channel on a city intersection. Three scenarios were defined, corresponding to three possible paths that a transmitter UAV would take when flying at low altitude. In all of them, the receiver is stationary as a defensive base station would be. We simulated the same scenarios with the AIT ray tracer [5] in order to check its performance in city street canyons. As it was mentioned in Chapter 3, the ray tracer is able to generate rays from LOS, reflection, diffraction, diffuse scattering and penetration.

The measured data with the channel sounder and the predictions of the ray tracer were compared in terms of the local scattering function. We used a multitaper estimator based on DPS sequences to obtain a time-variant version of the local scattering function. From it, we calculated the time-variant PDP and DSD, together with the mean and rms spread of the delay and Doppler shifts. Overall, our observations can be listed as follows.

1. In all three scenarios, the ray tracer and the channel sounder results coincided in the existence of an initial cluster, whose delay changed according to the transmitter position. In the simulation, this cluster was conformed by paths resulting from all the propagation mechanisms in the ray tracer, which suggests that their prediction is accurate.
2. After the main cluster, the PDP of the measurements always decayed slower than the PDP of the simulations, whose prediction of late clusters was poor or in-existent. The deficiency in predicting long paths could be caused by the used diffuse-scattering model or the constrain of reflections up to second order (used to reduce the computation complexity).
3. The mean delay obtained from the predictions with the ray tracer was always smother and usually smaller than the measured mean delay. Additionally, the measurements always presented a larger rms delay spread. Together, these results could be originated by the poor prediction of long rays, or an incorrect distribution of the power in the propagation mechanisms. The latter idea could be tested by changing the scattering coefficient of (3.9) in the simulation parameters.
4. In the analysis of the DSD, the simulations only predicted a few Doppler shifts of those obtained by the measurements. The correct predictions were always associated to LOS, reflection or diffraction. However, it was impossible to track any Doppler component resulting from diffuse scattering due to the model that we used. For each simulation snapshot, the diffuse tiles were randomly generated and did not have any relation with the set of the previous simulation. As a result, each simulation consisted of a different superposition of random phasors and hence, the DSD of scattered rays was white noise.
5. When the simulations were dominated by diffuse scattering, the mean and rms spread of the Doppler shift corresponded to the mean and standard deviation of the white-noise uniform distribution. When other propagation mechanisms dominated, the rms Doppler spread of the simulations was similar to the one of the measurements.
6. Overall, the deficiencies of the ray tracer, both in the PDP and the DSD, should be corrected before it can be used for wireless positioning. The local scattering function is not completely predicted for city street canyons.

5.2 Future work

Our current version of the channel sounder can be improved in several ways. Among the possible future projects, we can list the following:

1. It is important to characterize the phase stability of the channel sounder. In principle, it is impossible for the frequency references of both GPSDOs to be

exactly the same. According to the measurements that we have taken so far, the frequency difference between the local oscillators is at most 1.5 Hz, i.e. an error of $2.6 \cdot 10^{-8}\%$ with respect to the center frequency $f_c = 5.831$ GHz. These initial results need to be further explored in order to characterize the limitations in the channel sounder results.

2. Similarly, the time precision of the GPSDOs needs to be properly characterized. During the measurement campaign, the GPSDOs were always disciplined to the GPS, and hence, their precision lied around ±20 ns with respect to the UTC. So far, we have mitigated this error by running calibrations just before the measurements, which might be unfeasible in other circumstances. One possibility to improve the current precision consists of using a PPS correction circuit, that according to the manufacturer [14], might improve the precision below ±3 ns.

   Another option consists of implementing a back-to-back calibration before the measurements, where one clock is disciplined from the other. In an initial study of this procedure, we obtained that the drift between clocks after the calibration was less than 22 ns/h. However, when we repeated this calibration with some measurements, the clocks drifted more than 80 ns/h, possibly due to the movement of the GPSDOs during the measurements. We need to run more tests to determine which is and how to obtain the best precision in time.

3. AIT has now acquired 160 MHz USRPs. A natural extension of our work would be to increase the sounding bandwidth to 70 MHz.

4. The user interface of the sounder could be also improved to execute measurements faster. It is worth mentioning that we have already improved it and for example, we can now visualize in real time the Doppler-variant impulse response.

Regarding the ray tracer, the following is a list of future tasks:

1. To get a better idea of the effect of each propagation mechanism, we can use our current results to calculate the local scattering function for each of them. For example, the effect of the scattering coefficient could be checked to adjust the power differences between reflections and scattered rays.

2. The GPS coordinates of the measurements could be utilized as inputs for the ray tracer simulations. This way, each simulation snapshot could be associated to an actual time index and a closer comparison between the two tools could be obtained.

3. As it was observed in the comparison, one of the most evident failures of the ray tracer was the prediction of long MPCs. This deficiency is caused by the constrain of building the reflection tree up to second order, which also limits the order of scattered rays (see Section 3.5). We could include many already-proposed solutions as high-order reflection algorithms [55] or using propagation graphs to model multiple reflections [56] [57].

4. Another important aspect to improve is the performance of the scattering model when predicting the DSD. One possible solution would be the usage of the same diffuse tiles during a stationary region, which Mingming Gan has already explored. The problem with this implementation is the transition between stationary regions, because a new set of diffuse tiles would have to be calculated again. If this solution is implemented, we would have to correctly model how the diffuse scattering tiles change during the stationary region. For example, some of the tiles might be obstructed and in general, the reflection tree of a specific simulation snapshot of the stationary region might not be valid for the entire region.

References

[1] J. F. Keane and S. S. Carr, “A brief history of early unmanned aircraft,” *Johns Hopkins APL Technical Digest*, 2013.

[2] N. H. Motlagh, T. Taleb, and O. Arouk, “Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives,” *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2016.

[3] P. Keikhosrokiani, N. Mustaffa, N. Zakaria, and M. I. Sarwar, *Multimedia and Ubiquitous Engineering: MUE 2013.* Springer Netherlands, 2013, ch. Wireless Positioning Techniques and Location-Based Services: A Literature Review, pp. 785–797. [Online]. Available: http://dx.doi.org/10.1007/978-94-007-6738-6_97

[4] A. F. Molisch, *Wireless Communications*, 2nd ed. Wiley Publishing, 2011.

[5] M. Gan, “Accurate and low-complexity ray tracing channel modeling,” Ph.D. dissertation, Technische Universität Wien, Univ., Vienna, Austria, November 2015.

[6] R. Wang, C. U. Bas, O. Renaudin, S. Sangodoyin, U. T. Virk, and A. F. Molisch, “A real-time MIMO channel sounder for vehicle-to-vehicle propagation channel at 5.9 GHz,” in *2017 IEEE International Conference on Communications (ICC)*, 2017.

[7] H. Tuttlebee, *Software Defined Radio*, 1st ed. JOHN WILEY AND SONS LTD, 2002.

[8] *NI USRP-2953R Specifications*, National Instruments. [Online]. Available: http://www.ni.com/pdf/manuals/374197c.pdf

[9] *Xilinx 7 Series FPGAs Data Sheet: Overview. v2.4*, Xilinx. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf

[10] *Prototyping Software Defined Radio Systems with LabVIEW FPGA, Hands-On Seminar*, National Instruments. [Online]. Available: ftp://ftp.ni.com/pub/events/seminar_elit/prototyping_software_defined_radio_systems_manual.pdf

[11] *CBX DautherBoard Properties*, Ettus Research. [Online]. Available: http://files.ettus.com/manual/page_dboards.html#dboards_cbx

[12] *Super Ultra Wideband Amplifier ZVA-183W+*, Mini-Circuits. [Online]. Available: https://www.minicircuits.com/pdfs/ZVA-183W+.pdf

[13] *RD2458-5; Tri-Band 2.4, 5.3, 5.8 GHz Antenna*, Laird Technologies. [Online]. Available: https://assets.lairdtech.com/home/brandworld/files/ANT-DS-RD2458-5%200611.pdf

[14] *GPS10eR, 10 MHz GNSS Disciplined, Rubidium Frequency/Time Standard, Instruction Manual*, Precision Test Systems.

[15] *Multi GNSS Antenna: L1 GPS/Galileo/GLONASS/Beidou Antenna*, Meinberg. [Online]. Available: https://www.meinbergglobal.com/download/docs/shortinfo/english/info_gps-gln-l1-antenna.pdf

[16] *PXI Express, NI PXIe-8135 User Manual*, National Instruments. [Online]. Available: http://www.ni.com/pdf/manuals/373716b.pdf

[17] *PXI Express, NI PXIe-1082 User Manual*, National Instruments. [Online]. Available: http://www.ni.com/pdf/manuals/372752b.pdf

[18] *User Guide, Wireless-G Broadband Router, Model: WRT54GL*, Linksys by Cisco. [Online]. Available: http://downloads.linksys.com/downloads/userguide/WRT54GL_V11_UG_C-Web.pdf

[19] *MultiPlus Compact Manual*, Victron energy. [Online]. Available: https://www.victronenergy.de/upload/documents/Manual-MultiPlus-Compact-800-1200-1600-EN-NL-FR-DE-ES.pdf

[20] *Gel and AGM batteries*, Victron energy. [Online]. Available: https://www.victronenergy.com/upload/documents/Datasheet-GEL-and-AGM-Batteries-EN.pdf

[21] *LabVIEW*, National Instruments. [Online]. Available: http://www.ni.com/labview/

[22] Pipelining to optimize FPGA VIs (FPGA module). National Instruments. [Online]. Available: http://zone.ni.com/reference/en-XX/help/371599G-01/lvfpgaconcepts/fpga_pipelining/

[23] A. Dichev, "Vehicular channel sounding with a software-defined radio platform," Master thesis, Austrian Institute of Technology (AIT), 2017.

[24] M. Friese, "Multitone signals with low crest factor," *IEEE Transactions on Communications*, vol. 45, no. 10, pp. 1338–1344, Oct 1997.

[25] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, 3rd ed. Prentice-Hall International (UK) Limited, London, 1996.

[26] A. Bondeson, T. Rylander, and P. Ingelström, *Computational Electromagnetics*, 1st ed. Springer Science+Business Media, Inc, 2005.

[27] M. C. Lawton and J. P. McGeehan, “The application of a deterministic ray launching algorithm for the prediction of radio channel characteristics in small-cell environments,” *IEEE Transactions on Vehicular Technology*, vol. 43, no. 4, pp. 955–969, Nov 1994.

[28] R. Valenzuela, “A ray tracing approach to predicting indoor wireless transmission,” *Vehicular Technology Conference, 1993., 43rd IEEE*, pp. 214–218, May 1993.

[29] X. Li, “Efficient ray tracing simulation,” Master thesis, Lund Univ., and Forschungszentrum Telekommunikation Wien GmbH, Lund, Sweden, 2014.

[30] *Matlab*, Mathworks. [Online]. Available: https://www.mathworks.com/help/matlab/

[31] S. Orfanidis, *Electromagnetic Waves and Antennas*, 2016. [Online]. Available: www.ece.rutgers.edu/~orfanidi/ewa

[32] S. Ramo, J. Whinnery, and T. Duzer, *Fields and Waves in Communications Electronics*, 3rd ed. John Wiley and Sons, Inc, 1994.

[33] D. Griffiths, *Introduction to Electrodynamics*, 3rd ed. Prentice Hall, 1999.

[34] C. Balanis, *Advanced Engineering Electromagnetics*, 1st ed. John Wiley and Sons, Inc, 1989.

[35] E. Hecht, *Optics*, 4th ed. Addison Wesley, 2002.

[36] R. G. Kouyoumjian and P. H. Pathak, “A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface,” *Proceedings of the IEEE*, vol. 62, no. 11, pp. 1448–1461, Nov 1974.

[37] F. G. Bass, I. M. Fuks, and D. t. Haar, *Wave Scattering from Statistically Rough Surfaces*, 1st ed. Elsevier, 1979.

[38] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, and G. Falciasecca, “Measurement and modelling of scattering from buildings,” *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 1, pp. 143–153, Jan 2007.

[39] S. Saunders and A. Aragón-Zavala, *Antennas and Propagation for Wireless Communication Systems*, 2nd ed. John Wiley and Sons, Inc, 2007.

[40] J. W. Rayleigh, *The Theory of Sound*, 1st ed. London, Macmillan and co., 1877.

[41] V. Degli-Esposti, “A diffuse scattering model for urban propagation prediction,” *IEEE Transactions on Antennas and Propagation*, vol. 49, no. 7, pp. 1111–1113, Jul 2001.

[42] V. Degli-Esposti and H. L. Bertoni, “Evaluation of the role of diffuse scattering in urban microcellular propagation,” in *Gateway to 21st Century Communications Village. VTC 1999-Fall. IEEE VTS 50th Vehicular Technology Conference (Cat. No.99CH36324)*, vol. 3, 1999, pp. 1392–1396 vol.3.

[43] F. J. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, Jan 1978.

[44] Google. Google maps 3d view of the studied intersection. [Online]. Available: https://www.google.se/maps/@48.2274999,16.4229039,81a,35y,7h,39.24t/data=!3m1!1e3

[45] S. Wien. Geodatenviewer der stadtvermessung wien. [Online]. Available: https://www.wien.gv.at/ma41datenviewer/public/

[46] Qgis, a free and open source geographic information system, v.2.18.7. [Online]. Available: http://www.qgis.org

[47] S. B. Foundation. Blender. [Online]. Available: https://www.blender.org/

[48] S. Haykin, *Communication Systems*, 4th ed. John Wiley and Sons, Inc, 2001.

[49] L. Bernadó, T. Zemen, F. Tufvesson, A. F. Molisch, and C. F. Mecklenbräuker, "The (in-) validity of the WSSUS assumption in vehicular radio channels," in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, Sept 2012, pp. 1757–1762.

[50] L. Bernadó, T. Zemen, A. Paier, G. Matz, J. Kåredal, N. Czink, C. Dumard, F. Tufvesson, M. Hagenauer, A. Molisch, and C. F. Mecklenbräuker, "Non-WSSUS Vehicular Channel Characterization at 5.2 GHz - Spectral Divergence and Time-Variant Coherence Parameters," 2008.

[51] D. J. Thomson, "Spectrum estimation and harmonic analysis," *Proceedings of the IEEE*, vol. 70, no. 9, pp. 1055–1096, Sept 1982.

[52] W. van Drongelen. Multitaper power spectrum estimation, class material. [Online]. Available: https://epilepsylab.uchicago.edu/page/teaching

[53] D. Slepian, "Prolate spheroidal wave functions, fourier analysis, and uncertainty; V: the discrete case," *The Bell System Technical Journal*, vol. 57, no. 5, pp. 1371–1430, May 1978.

[54] L. Bernadó, T. Zemen, F. Tufvesson, A. F. Molisch, and C. F. Mecklenbräuker, "Delay and doppler spreads of nonstationary vehicular channels for safety-relevant scenarios," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 82–93, Jan 2014.

[55] M. Gan, Z. Xu, V. Shivaldova, A. Paier, F. Tufvesson, and T. Zemen, "A ray tracing algorithm for intelligent transport systems in tunnels," in *2014 IEEE 6th International Symposium on Wireless Vehicular Communications (WiVeC 2014)*, Sept 2014, pp. 1–5.

[56] T. Pedersen, G. Steinbock, and B. H. Fleury, "Modeling of reverberant radio channels using propagation graphs," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 12, pp. 5978–5988, Dec 2012.

[57] G. Steinböck, M. Gan, P. Meissner, E. Leitinger, K. Witrisal, T. Zemen, and T. Pedersen, "Hybrid model for reverberant indoor radio channels using rays and graphs," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 9, pp. 4036–4048, Sept 2016.

Printed by Tryckeriet i E-huset, Lund 2017

LUND UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2017-588

http://www.eit.lth.se