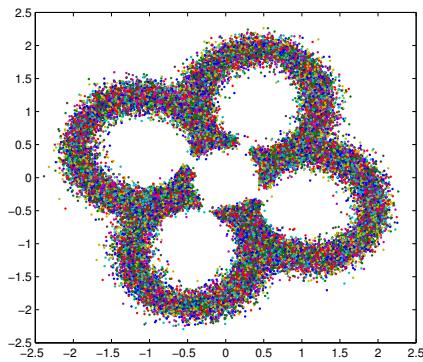


KUNGLIGA TEKNISKA HÖGSKOLAN

PROJECT REPORT

Wireless Communication over 60 GHz



Authors:

Johan LOVÉN

João LEMOS

Mathieu NAVAUX

Peter ABEL

Frédéric DE PORET

Adrià CASAMITJANA DÍAZ

Daniel ANFELT

Natalia DEMENTIEVA

Course Responsible:

Per ZETTERBERG

Project Assistant:

Senay NEGUSSE

A project report submitted in fulfilment of the requirements

of the course EQ2440/EQ2435

in the

Signalbehandling

School of Electrical Engineering

May 2014

“Apparently there is nothing that cannot happen today.”

Mark Twain

KUNGLIGA TEKNISKA HÖGSKOLAN

Abstract

Wireless Communication over 60 GHz

The purpose of this report is to document the findings and learning outcomes of a small group of students completing the course EQ2440/EQ2435 at the Royal Institute of Technology KTH in Sweden. The project required the students to build a fully functioning communications system, operating at 60 GHz frequency, with a data rate of 100 kbits/s. An added advanced goal of the project was to be able to send files of 10 kbyte size, in real-time.

During project three systems were designed and simulated in Matlab two of these systems were implemented in C++. The first system was built as a simple single-carrier system operating at 5 GHz and served as a practice system. The second system was built using OFDM based on the skills learned in the first system and operated at 60 GHz. The third system was constructed using conventional methods, then simulated and tested in Matlab.

The first single carrier system achieved a data rate of 12.5 Mbits/s and required 16 ms for decoding with 0 BER. The second system achieved a data rate of 72 Mbits/s in Matlab transmission. However, since channel coding was not implemented in C++, the system, implemented in C++ reached an effective rate of 30 Mbits/s with close to zero BER. These results were obtained at 5 GHz. At 60 GHz, the second system, with slightly altered parameters, performed at a data rate of 45 Mbits/s with 0 BER in offline mode. In real-time, the system's effective data rate reached 18 Mbits/s, with 0 BER and a decoding time of 16 ms.

The third system did not manage to perform a reliable transmission over 60 GHz. It has been left for future research to complete.

Due to the limited time duration of this project, the ambitiously set advanced real-time requirements was not reached. However, when we got official feedback, the course responsible was satisfied with our achievements i.e. achieve a real time transmission considering a data rate of 1 Mbps.

Acknowledgements

This project have been a great experience and a lot of fun for all the members of the group. We would like to thank Per Zetterberg, our course responsible, and Senay Negusse, our project assistant, for helping us when we got into trouble, for guiding us when we needed help and for providing the necessary hardware and software. We all consider this course one of the best courses that we have taken and a great course to summarize what we have learned so far. Our only regret is that we could not arrange for Senay to attend our group after-work and our party at Daniel's apartment.

We would also like to thank Bluewave Microsystems for providing us with the circuit board-design necessary to complete the project.

Contents

Abstract	ii
Acknowledgements	iii
Contents	iv
List of Figures	x
1 Background	1
1.1 Historical overview	2
1.2 60 GHz Communication	2
1.2.1 Applications	3
1.2.2 Engineering Constraints	4
1.2.3 Previous Work	4
1.3 Goal	5
1.4 Organization and Resources	6
1.4.1 Human Resources	6
1.4.2 Bluewave Microsystems	6
1.4.3 Course responsible and Assistance	6
1.4.4 Hardware	6
1.4.5 Software and Related Links	7
1.4.6 Organization	7
2 Methodology	8
2.1 The different groups	8
2.1.1 Theory Group	8
2.1.2 Implementation/USRP-Group	9
2.1.3 Project Manager	9
2.2 Meetings	9
2.3 Gantt chart	10
3 Theory	12
3.1 Channel characteristics	12
3.1.1 Noise	12
3.1.2 Signal fading	13

3.1.3	Path loss and Shadow loss	14
3.1.4	Local Oscillators	15
3.2	Modulation	15
3.2.1	Phase Shift Keying	15
3.2.2	Amplitude Modulation	16
3.2.3	Quadrature Amplitude Modulation	17
3.3	Orthogonal Frequency Division Multiplexing	18
3.3.1	Interleaver	19
3.3.2	Implementation tricks	19
3.3.3	Packet detection	20
3.3.4	DFT precoding	21
3.4	Fast Fourier Transform (FFT)	22
3.5	Kalman filter	22
3.6	Channel coding	22
3.6.1	Convolutional codes	23
3.6.2	Turbo codes	23
3.7	Antenna	24
4	Technical specification	25
4.1	The USRP	25
4.1.1	N210 technical description	26
4.1.1.1	Features	26
4.1.2	The millimeterwave transmitter and receiver	27
4.1.2.1	Transmitter's features	27
4.1.2.2	Receiver's features	28
4.2	Description of software and functions	28
4.2.1	USRP Class	28
4.2.2	Multi_USRP Class	29
4.2.3	Device Pointer	29
4.2.3.1	USRP Address	29
4.2.4	USRP Clock	29
4.2.5	USRP Parameters	29
4.2.6	USRP RF Setting	30
4.2.7	USRP tx_stream/rx_stream	30
4.2.7.1	metadata_t	30
4.2.7.2	stream_cmd_t	30
5	Single-Carrier system	31
5.1	Transmitter	31
5.2	Receiver	32
5.2.1	Local Oscillators revisited	34
5.3	Real-time Implementation	35
5.3.1	Transmission	35
5.3.2	Reception	36
5.3.2.1	The USRP thread (usrpT)	36
5.3.2.2	The detection thread (detectionT)	36
5.3.2.3	The processing threads	37

5.3.3	Received Sequence	37
5.3.4	Processed Sequence	38
5.4	System Performance	40
5.5	Results and conclusions	40
5.6	Lessons learned from system 1	41
5.6.1	C++ Coding, Git, ITPP and Harness Methodology	41
5.6.2	Over- and Underflows	42
5.6.3	Transient Problem	42
5.6.4	Root Mean Square (RMS) of the Signal	43
5.6.5	Conjugate Problem	44
6	OFDM system	45
6.1	Transmitter design and specifications	46
6.1.1	Data and Source Coding	46
6.1.2	Channel coding	48
6.1.2.1	Encoder	48
6.1.2.2	Decoder	49
6.1.3	Constellation mapping	52
6.1.4	Padding with random symbols	53
6.1.5	Carrier and pilot allocation	54
6.1.6	Cyclic prefix and suffix insertion	54
6.1.7	Training Sequence	55
6.1.8	Packet detection	56
6.2	Receiver Part	59
6.2.1	Sampling time for the synchronization	61
6.2.2	Filtering of the pilot's channel information	63
6.2.3	Channel interpolation	64
6.2.3.1	Linear interpolation	65
6.3	Channel description	65
6.3.1	Dealing with phase noise	67
6.3.2	Channel measurements	71
6.4	From 5 to 60 GHz	74
6.4.1	Frequency offset	75
6.4.2	Channel measurements	76
6.4.3	Modifications in the algorithms	77
6.4.3.1	Synchronization	77
6.4.3.2	Frequency offset	79
6.4.3.3	Channel Equalization	80
6.4.4	Real environment characterization	81
6.4.4.1	Distance	81
6.4.4.2	Saturation issue	81
6.4.4.3	Frequency and Impulse responses for 60 GHz channel .	83
6.4.5	PAPR	85
6.5	Performance analysis	85
6.5.1	Kalman Filtering	86
6.5.2	Constellations and phase noise	88
6.5.3	Gains, RMS and BER	91

6.6	Summary	96
7	OFDM system in C++	98
7.1	Modifications from Theoretical Second System	98
7.1.1	Transmitter	99
7.1.2	Receiver	99
7.2	Real-time Implementation	99
7.2.1	Transmission	100
7.2.2	Reception	102
7.2.3	Received Sequence	103
7.2.4	Receiver Implementation on C++	104
7.2.4.1	Catching Training Sequence and Coarse Channel Estimation	104
7.2.4.2	Getting OFDM Data	105
7.2.4.3	Data Separation and Channel Information	105
7.2.4.4	Kalman Filtering of the Channel	105
7.2.4.5	Channel Interpolation	106
7.2.4.6	Channel Correction and Data Detection	108
7.2.5	Results and Parameters	108
7.2.5.1	USRP Parameters	108
7.2.5.2	Channel Estimation	109
7.2.5.3	Received Constellation	109
7.2.6	Implementation Requirements	112
8	DFT-precoded OFDM (System 3)	115
8.1	DFT precoded OFDM	116
8.2	Channel estimation	119
8.3	Simulation results	121
8.3.1	Distribution of CFO estimations	121
8.3.2	Estimation of PHN	122
8.3.3	Estimation of the channel	123
8.4	Tracking of the residual CFO and CPE	124
8.4.1	Theoretical description	125
8.4.2	Comparison of the two algorithms	127
8.4.3	Performances of the Kalman filter	128
8.5	Simulation results of the system 3	130
8.5.1	Bit Error Rate evolution with SNR	130
8.5.2	Performance analysis for different CFO	131
8.5.3	Received constellation	132
8.6	Real transmission results	134
9	Conclusion	137
9.1	Project Management	137
9.2	60 GHz communications	137
9.3	System 1	138
9.4	System 2	138
9.5	System 3	139

9.6 Discussion	140
9.7 Future work	140
9.7.1 System delay	141
9.7.2 Antennas	141
10 Technology Economy Leadership	143
10.0.3 Structure	143
10.1 Methodology	144
10.1.1 Theoretical framework	144
10.2 Background	145
10.2.1 60 GHz communication	146
10.2.2 Communication system	146
10.2.3 Modular Smartphone	146
10.3 Business plan introduction	147
10.3.1 Proposed product	147
10.4 Management summary	148
10.4.1 Team gaps and Other personnel	148
10.5 Market analysis	149
10.5.1 Market size (current and future)	149
10.5.2 Market trends	149
10.5.3 Market growth rate	149
10.5.4 Market profitability	150
10.5.5 Potential market size	151
10.5.5.1 Target group	151
10.5.5.2 Target Analysis	151
10.5.6 Industry cost structure	152
10.5.7 Production cost of module	153
10.5.8 Conclusions on Cost	154
10.5.9 Distribution channels	154
10.5.10 Critical success factors	155
10.5.10.1 In the market	155
10.5.10.2 In the product	155
10.5.11 Competitor analysis	155
10.5.12 SWOT analysis	156
10.5.12.1 Strengths	156
10.5.12.2 Weaknesses	156
10.5.12.3 Opportunities	157
10.5.12.4 Threats	157
10.6 Operations plan	157
10.6.1 Activity plan	159
10.7 Marketing plan	159
10.7.1 Product	159
10.7.2 Business location and Marketing budget	160
10.7.3 Pricing and Price differentiation	160
10.8 Concluding chapter	162

Bibliography	163
---------------------	------------

List of Figures

1.1	Global Frequency Plan	3
1.2	MM-wave attenuation	4
1.3	Bandwidth and power limitations in different parts of the world [Huang and Wang, 2011].	5
1.4	Organization of the project group	7
2.1	Gantt chart of USRP-group	10
2.2	Gantt chart of project leader	11
2.3	Gantt chart of Theory-group	11
3.1	Constellation diagram of BPSK and QPSK [Proakis, 2007].	16
3.2	PAM signal with 4 different levels of amplitude [Proakis, 2007].	17
3.3	Constellation of 16QAM [Islam et al., 2011].	18
3.4	Example of a convolutional encoder [Proakis, 2007]	23
4.1	USRP N210 (Same interface as N200).	25
4.2	USRP N210 detailed specifications.	26
5.1	Block Diagram of the Transmitter	32
5.2	Analysis of transmitted sequence.	32
5.3	Block Diagram of the Receiver.	34
5.4	Local oscillator simulation leakage	35
5.5	Received sequence. The top image gives the real part of the sequence and the bottom image gives the spectrum.	38
5.6	Results from C++. Top left: real part of received signal. Middle left: real part of the matched filter. Bottom left: crosscorrelation of training symbols and signal. Top right: downsampled signal before phase correction. Middle right: downsampled signal after phase correction. Bottom right: BER (constant zero)	39
5.7	BER as a function of transmission gain.	40
5.8	Measurement and spectrum of the channel without filtering the local oscillator leakage	43
5.9	Measurement and spectrum of the channel without filtering the local oscillator leakage	43
6.1	Transmitter part of the OFDM system	46
6.2	Source coding block scheme	47
6.3	Block diagram of the encoder	48
6.4	A rate-1/2 binary convolutional encoder used in the system	49

6.5	Turbo code decoder [Johnson, 2009]	50
6.6	The performance of the turbo code used in the simulation	52
6.7	64-BEES, an energy efficient mapping.	53
6.8	Pilot and information subcarriers	54
6.9	ACF of Golay training sequence	56
6.10	The structure of the OFDM frame	56
6.11	Spectrum of the transmitted OFDM signal	56
6.12	Packet detection using the Schmidl-Cox algorithm, with an SNR of 6 dB	58
6.13	Packet detection using the Schmidl-Cox algorithm, with an SNR of 0 dB and a LO offset of 0.01	59
6.14	Block scheme of the receiver end for OFDM system	60
6.15	Phase noise measured for this project	68
6.16	Phase noise simulated for 60 GHz system	69
6.17	Channel model	70
6.18	Eye diagram for AWGN channel	70
6.19	Eye diagram for AWGN channel in presence of the phase noise	70
6.20	Power spectrum of the both transmitted and received sequence for 5 GHz	71
6.21	Channel impulse response for 5 GHz	72
6.22	32-QAM constellation for different values of prefix and postfix	73
6.23	Eye diagram with chosen CP=12 and CF=4	73
6.24	Eye diagram with chosen CP no CS	74
6.25	Eye diagram with CP=3 and CS=4	74
6.26	Distorted signal	75
6.27	Power spectrum of the transmitted exponential signal, $\nu = 0.2$	76
6.28	Power spectrum of the received exponential signal, $\nu + \Delta\nu_{off}$	76
6.29	Power spectrum of the received noise at 60 GHz	77
6.30	Crosscorrelation of the received data with the training sequence at 60 GHz, $\nu = 0.2$	78
6.31	Power spectrum of the transmitted OFDM system at 60 GHz	78
6.32	Power spectrum of the received OFDM system at 60 GHz	79
6.33	Phase variation	79
6.34	Phase variation in frequency	80
6.35	Constellations at 60 GHz	81
6.36	BER vs distance for 60GHz	82
6.37	BER in function of the gain for 60 GHz transmission	82
6.38	Received constellations for different gains	83
6.39	Channel impulse and frequency responses at 60 GHz	84
6.40	BER as a function of the frequency	84
6.42	Comparison of diffent equalization techniques	87
6.43	Comparission of equalization techniques, with 16-QAM	88
6.44	BER for AWGN	89
6.46	BER for 16-QAM and BEES16	90
6.47	BER for 16-QAM	91
6.48	BER in function of the receiver gain for different constellations, 32 carriers	92
6.49	BER in function of the receiver gain for different constellations, 64 carriers	93
6.50	BER in function of the receiver gain for different constellations, 32 carri- ers, RMS=2200	94

6.51	BER in function of the receiver gain for different constellations, 64 carriers, RMS=2200	95
6.52	BER in function of the receiver gain for 16-QAM	96
7.1	Example of transmitted Sequence at 60 GHz. Real part and Spectrum of data.	101
7.2	Example of received Sequence at 60 GHz. Real part and Spectrum of received data.	103
7.3	Example of Phase Estimation.	107
7.4	Channel estimation at 5 GHz for USRP 2 (Gain and Phase).	109
7.5	Channel estimation at 60 GHz (Gain and Phase)	110
7.6	Received 4-QAM constellation at 5 GHz.	110
7.7	Received 16-QAM constellation at 5 GHz.	111
7.8	Received 4-QAM constellation at 60 GHz.	111
7.9	Received 16-QAM constellation at 60 GHz.	112
8.1	Block diagram of DFT-precoded OFDM system	116
8.2	Block scheme for channel estimation	116
8.3	The structure of OFDM transmitted signal	117
8.4	DFT-precoded with $M = 30$ and clip rate 0.3	117
8.5	DFT-precoded with $M = 60$ and clip rate 0.9	118
8.6	PHN Estimation	122
8.7	Phase Noise estimation (perfect CFO estimation)	123
8.8	Phase Noise estimation (non-perfect CFO estimation)	123
8.9	MSE of the channel	124
8.10	Phase of a pilot over the OFDM symbols.	125
8.11	Phase of the pilot after correction	127
8.12	Comparison of the BER between LLSE tracking and Kalman filtering for different CFO	128
8.13	Simulated received constellation	128
8.14	BER in Kalman tracking in presence of not of PHN, with or without DFT precoding	129
8.15	Performance analysis	130
8.16	Performance analysis	131
8.17	Performance analysis	132
8.18	BER evolution without CFO correction ($SNR = 30dB$)	132
8.19	Received constellations for different receiver configurations	133
8.20	Estimated channel	134
8.21	Pilots after channel correction	135
8.22	Phase of the pilot after correction	135
8.23	Position of the errors on the I component	136
10.1	Smartphone sales trend worldwide 2009-2013 [Singh, 2013].	150
10.2	Smartphone sales worldwide 2013 [Pettey and van der Meulen, 2013].	150
10.3	Production cost of iPhone 5 components [Fiegerman, 2012].	152

*Dedicated to all the wonderful people of the world and especially
the exceptional Claude Shannon*

Chapter 1

Background

The main goal of this project was to design and implement a real-time communications system, operating at 60GHz frequency. This type of communication is expected to be one of the possible solutions of future wireless communications and is therefore a relevant problem which attracts attention of both communication providers and researchers.

In this project we focused on designing a line of sight communication link using two Universal-Software-Radio-Peripherals, USRPs, as transmitter and receiver. The core target was to reach the requirements of the project and to learn how the theoretical knowledge acquired from previous courses can be applied on a real life problem. Some algorithms used in the project were developed by the group and were mostly based on previous experience and past courses taken, other solutions were based on IEEE research papers. Both were successfully simulated, tested and compared where it was possible. The optimal solutions in terms of computational complexity, available resources and time constraints were implemented in C++.

This project required various skills from each member: MATLAB and C++ programming, practical algorithm development based on knowledge from such courses as Signal Processing, Advanced Digital Communications, and Adaptive Signal Processing as well as new information found in research papers and other sources. Furthermore, the project gave the members of the group valuable experience within work organization as well as management of a relatively large group of 8 people. The project can be considered successful. We met the advanced requirement for the effective bit rate in offline regime, reaching a staggering 45 Mbits/s with 0 BER. Some requirements were not completely met, but the reasons and proposed way of achieving them are described in the report.

1.1 Historical overview

With the invention of the radio-transmitter in the late 19th century, the inception of the mobile communication technologies in the mid 1940's and the rise of wireless internet during the 21st century, the available frequencies for new applications to use have become fewer and the available bandwidth scarce [Sarkar et al., 2006].

Until now, the free bandwidth that has been considered the most interesting and versatile for scientists and corporations have been the 2.4 and 5GHz bands. These bands are relatively easy to work with, they are unlicensed and they are practical when implementing medium-distance applications such as WIFI. However, the offered free bandwidth at these frequencies is relatively slim. In Sweden, the available continuous bandwidth amounts to less than 0.1 GHz in the 2.4 GHz frequency band and less than 0.3 GHz in the 5 GHz frequency band [Post och Telestyrelsen, 2014].

Prices of internet data rates are being pushed down. The difference between 100 Mbps and Gigabit-speeds are closing, which creates a need for developing techniques of transmitting at gigabit speeds wirelessly while at the same enabling systems to transmit at even higher rates [Huang and Wang, 2011].

However, the small available bandwidths of the 2.4 and the 5 GHz frequency bands limits the amount of data that can theoretically be sent, which has made researchers turn to the millimeter-wave-range.

1.2 60 GHz Communication

In the range of frequencies between 30 GHz and 300 GHz, the millimeter-wave spectrum is found. The name derives from the fact that, at these frequencies, the wavelength of signals are between 10-1 mm. This frequency band is sometimes also denoted Extremely High Frequency (EHF). The spectrum is largely unused and/or unlicensed and has therefore, together with a decreasing amount of free bandwidth available within lower frequencies, begun attracting the attention of researchers and corporations [Yong et al., 2011].

There are a couple of unlicensed (free-to-use) or easy to license bandwidths available in the EHF, one of these frequency bands is situated around 60 GHz [Yong et al., 2011]. This is the band that is considered interesting and that will be used in this study.

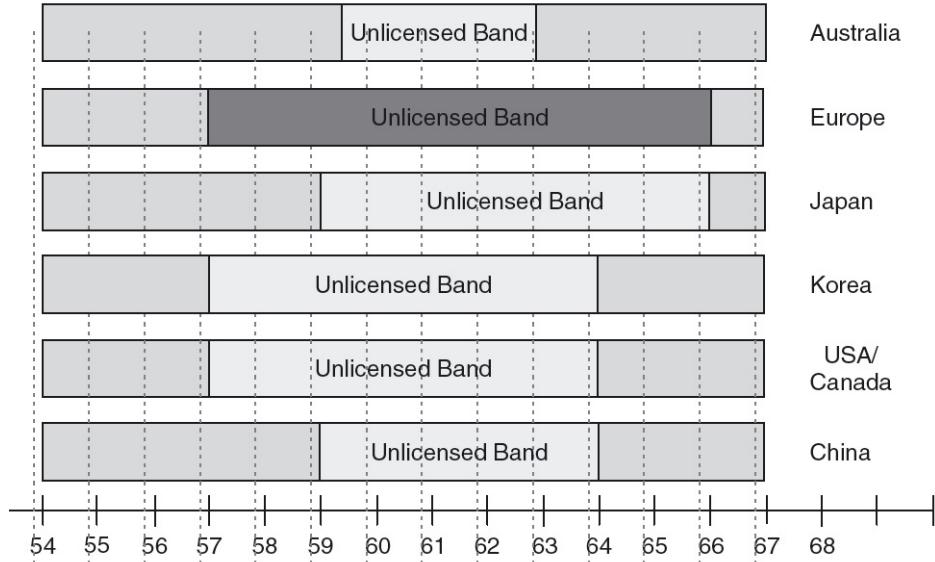


FIGURE 1.1: Frequency plan globally for 60 GHz) [Yong et al., 2011].

It can be seen in figure 1.1 that in most of the world it is possible to use unlicensed band around 60 GHz with a channel bandwidth of 7 GHz which motivates further studies of these bands.

As can be seen in figure 1.2, oxygen attenuation is at its peak at 60 GHz. 5 mm wave attenuation is also very severe, rendering the combined path loss extremely high. This makes the frequency band most suitable for short range, 1-10 m, Line-of-Sight (LOS) communication ensuring high security and low interference.

1.2.1 Applications

As the quality of movies and internet-media increases, there is a continuously rising demand for streaming these services over a short distance, for example from an ATSC tuner or computer to a television set. In these indoor-environments where LOS is often available 60 GHz communication is a very viable option where the necessary data rates (gigabit per second) can be achieved [Emami, 2013].

Wireless backhaul solutions are also considered to be valid applications for 60 GHz traffic. Instead of communication via cable, a LOS wireless communication system could be designed that, due to the characteristics of 60 GHz communication, would be able to transfer at very high data rates while at the same time being extremely secure [Huang and Wang, 2011].

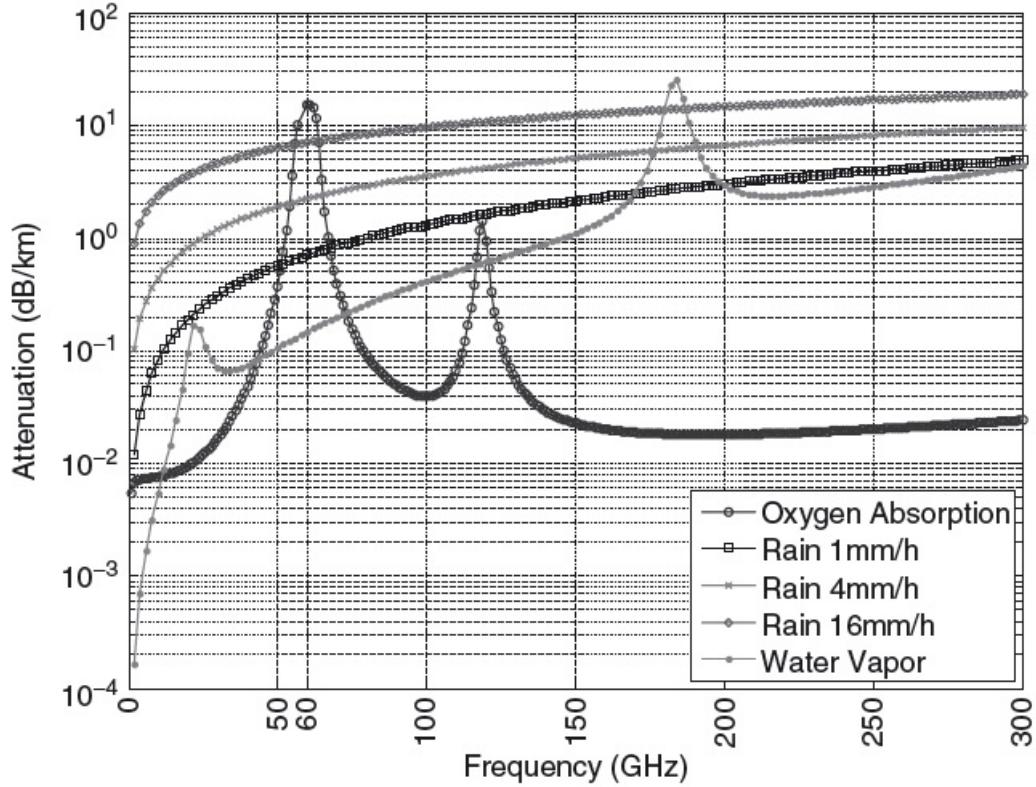


FIGURE 1.2: Attenuation from oxygen absorption, rain and water vapor) [Yong et al., 2011].

1.2.2 Engineering Constraints

As can be seen in figure 1.2 above the path loss at 60 GHz is large. This is primarily what limits the communication to short ranges and it is primarily due to the increase frequency and the attenuation. Apart from path loss, there are a number of other limitations attached to this task. Although scattering of the signals is almost non-existent, it makes the multipath propagation even worse [Huang and Wang, 2011].

Regulation is another problem. Apart from the generous limit in bandwidth, the power that is allowed to be transmitted is also limited, in some areas. As can be seen in figure 1.3, apart from bandwidth, the allowed signal power differs and can be restricted to as little as 10 mW with an antenna gain as low as 37 dBi.

1.2.3 Previous Work

During the last 10 years a lot of studies have been made concerning 60 GHz wireless communications systems. Different methods for modulation have been tried, OFDM is always one of the solutions but not always the best for every application due to

Region	Unlicensed Bandwidth (GHz)	Max. Tx Power	Max. Antenna Gain
Europe	9 GHz (57–66) min 500 MHz	20 mW	37 dBi
Japan	7 GHz (59–66) max 2.5 GHz	10 mW	47 dBi
Korea	7 GHz (57–64)	10 mW	To be decided
Germany	1 GHz (57.1–57.8) (58.6–58.9)	50 mW	Not specified
USA	7 GHz (57–64)	500 mW	Not specified

FIGURE 1.3: Bandwidth and power limitations in different parts of the world [Huang and Wang, 2011].

its sensitivity to phase and frequency offsets. However, it is a very suitable option on account of its robustness and way of using a wide spectrum efficiently [Yong et al., 2011].

There are already a number of different standards such as WirelessHD and 802.11ad, developed for both single- and multi-carrier transmission. Common denominators include short range, under 20 m, and high data rate, usually above 1 Gbps. Large delay spread and regulatory restriction of the power are considered the biggest limitations of the systems [Huang and Wang, 2011].

1.3 Goal

The main goal of this project was to build a communications system that would satisfy the following requirements:

- i real-time transmission;
- ii effective bit rate of at least 100 kbit/s;
- iii capability to transmit files of a size between 10 and 100 kbytes;
- iv Frame-Error-Rate of at most 10^{-2} with channel coding or
- v BER of at most 10^{-6} without channel coding;

In addition to the listed primary goals, we tried to push the data-rate up to 1 Mbit/s in real-time and 20 Mbit/s in non real-time. In order to fulfill the mid-term requirements, the system was first implemented in offline mode.

1.4 Organization and Resources

1.4.1 Human Resources

The project team consisted of 8 members.

- Project manager: Johan Lovén
- João Lemos
- Mathieu Navaux
- Peter Abel
- Frédéric De Poret
- Adrià Casamitjana Díaz
- Daniel Anfelt
- Natalia Dementieva

1.4.2 Bluewave Microsystems

A circuit board-design was supplied by Bluewave Microsystems. It would not have been possible to complete this project without their help.

Bluewave Microsystems can be found on <http://www.bluewave.se/>.

1.4.3 Course responsible and Assistance

- **Per Zetterberg** (Teacher)
- **Senay Amanuel Negusse** (Project Assistant)

1.4.4 Hardware

- 4 PCs with Ubuntu 12.04 (two UI: gnome or unity);
2 extra PCs were added during the last 4 weeks;
- 3 USRP N210 and 1 USRP 2 by *Ettus Research*TM;
- 2 60 GHz boards for the USRPs added during the last 4 weeks;
- 4 Ethernet Cables;

1.4.5 Software and Related Links

- GNU Emacs, Coding language C++;

- MATLAB R2013a with KTH USRP skeleton code ;

- Compiling source code:

g++ (<http://gcc.gnu.org/>);

make (<https://www.gnu.org/software/make/>);

- Debugging:

printf, cout, save/load from/to file;

- Signal processing and communications library:

IT++ (<http://itpp.sourceforge.net/>);

- Code repository and versioning :

GitHub (<http://toroid.org/ams/git-central-repo-howto>)

1.4.6 Organization

The organization of the project team is shown in figure 1.4. The organization was slightly modified during the later stages of the project but this structure was kept for the majority of the time.

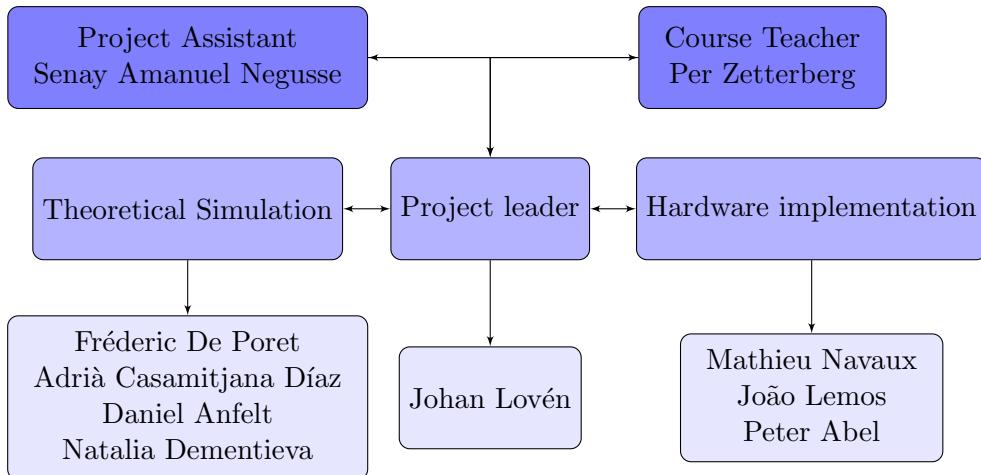


FIGURE 1.4: Organization of the project group

Chapter 2

Methodology

This chapter first presents the general organization of the team. Following that, some considerations about time management and communication are discussed.

2.1 The different groups

First a literature study was conducted, where the members of the group gathered information about the topic. This was done in order to acquire enough general information on the problem and also to know within what areas they wanted to work. The literature study and the project description suggested a division of the workforce into three sub-groups:

- the Theory Group, in charge of designing the system and the algorithms needed to realize this system
- the Implementation Group, working with C++ implementation of the algorithms designed by the theory group, and also testing these implementations in the hardware
- the Project Manager, writing the report, coordinating the meetings and supporting the team in general

The division of the group can be seen in figure 1.4.

2.1.1 Theory Group

The main objective of the Theory Group was to determine what system design would be the most suitable to solve the problem at hand. This was done by searching the

literature, by simulating in Matlab and finally testing on real data. Some of the foremost problems that had to be considered and dealt with were frequency offset estimation, i.e. identification of the frequency offset between the two local oscillators, channel estimation and phase noise. After defining a suitable system, which could handle all the anticipated difficulties of the 60 GHz link, it was described in detail as to facilitate implementation in C++.

2.1.2 Implementation/USRP-Group

For the Implementation Group, the most important task was to implement the functional system designed by the Theory Group. Problems arose both in how the general Matlab-designs were to be implemented in C++ in an efficient way, and also in how the written software interacted with the hardware. When the system had been implemented successfully, a test suite was created in order to enable everyone to test the system.

For the Implementation Group, the initial phase of the project consisted of a combination of literature studies, both in C++, IT++ and USRP-hardware, and all their respective libraries. On top of this, a set of assignments had to be completed and demonstrated at a mid-term presentation.

2.1.3 Project Manager

The main role of the Project Manager was to complete the report. This was done by being involved in the entire project, being able to describe the entire system and its separate parts. A big part in doing this was having extensive meetings where everyone was able to explain the work that was currently being done, in a comprehensible way. On top of this, there were elements of monitoring and encouraging the members of the subgroups, maintaining focus on the goals and keeping the time-plan.

2.2 Meetings

Meetings were conducted on a weekly basis and followed a recurring schedule. The agenda followed a number of simple questions which, when answered, gave a good view of what had been done during the last week and what work needed to be done the following week. The questions were:

- What has been done?

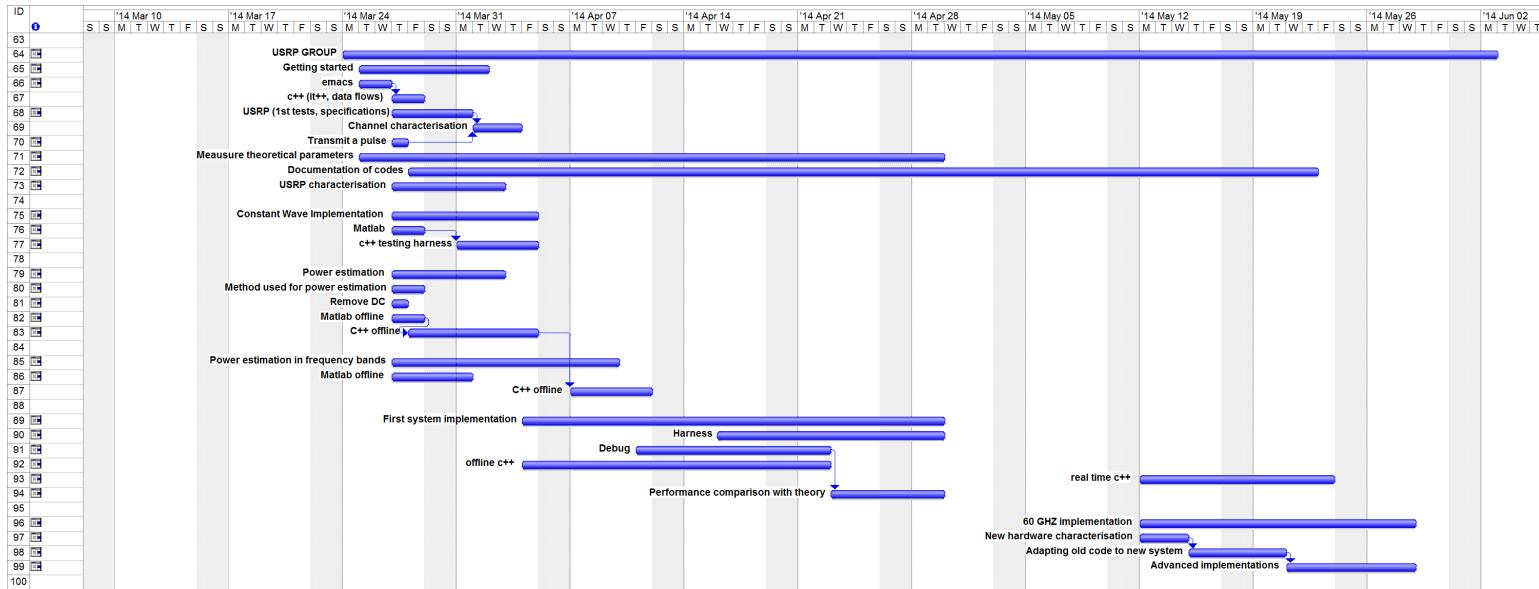


FIGURE 2.1: Gantt chart of USRP-group

- Who has done it?
- What activities are left from last week?
- What activities are to be done for this week?
- How many hours have been put in by each member?

The minutes of the meeting were compiled into a progress report, depicting the answers for the questions above. The reports were sent to the supervisor and the assistant in order for them to be able to follow the progress of the group as well as to enable them to track the workload of each member.

2.3 Gantt chart

A Gantt chart was constructed during the planning phase of the project. The members of the subgroups decomposed the goals into activities. The members also approximated the time to completion of these tasks and the team leader put it all together into a Gantt chart. The chart gave a good visualization of the work that had to be completed as well as an approximate time limit. The separate chart for each one of the groups can be seen in figures 2.1, 2.2, and 2.3

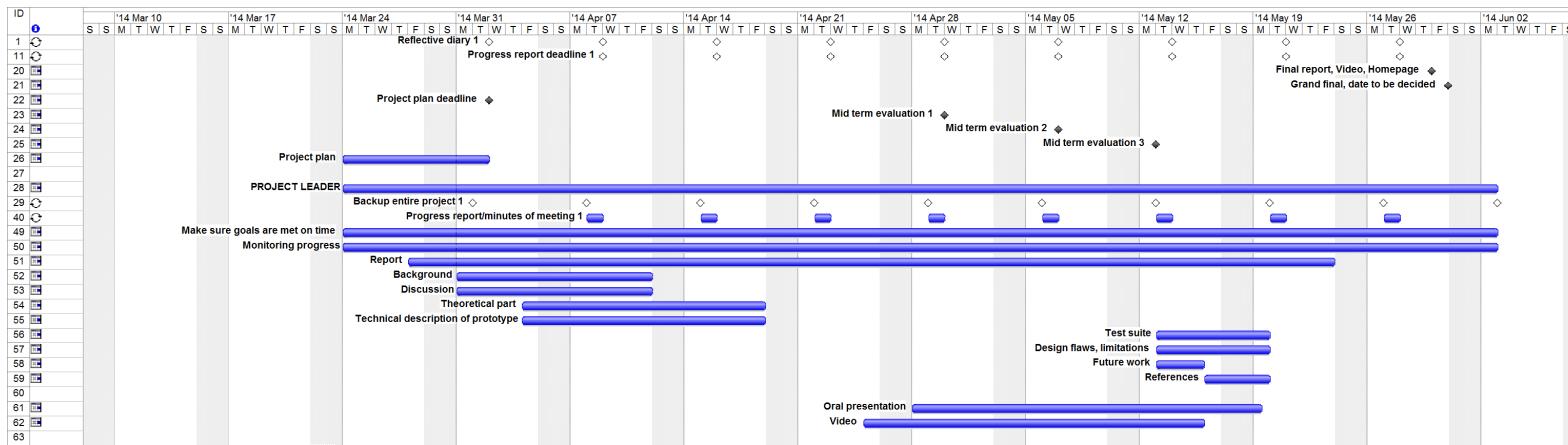


FIGURE 2.2: Gantt chart of project leader

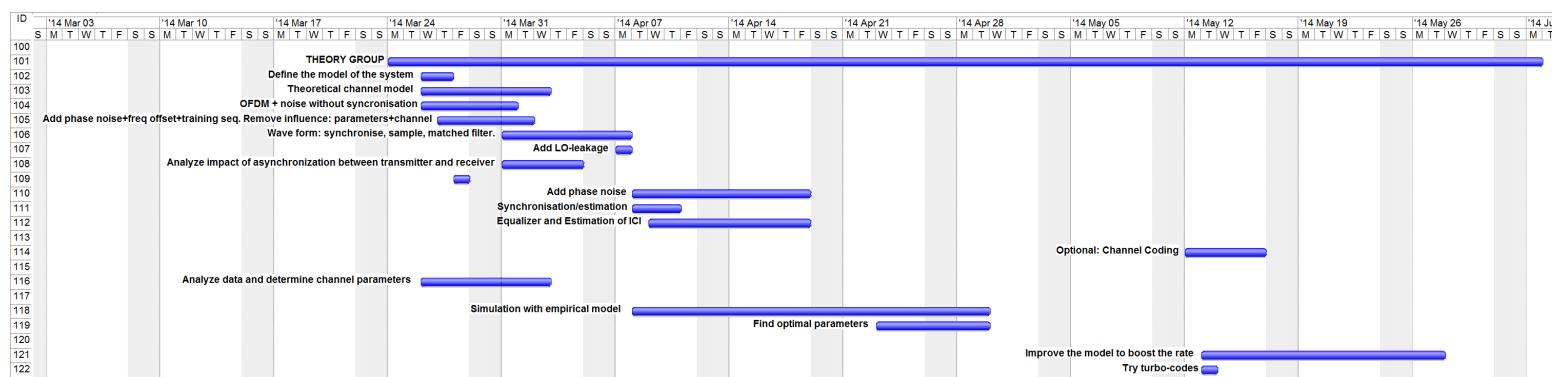


FIGURE 2.3: Gantt chart of Theory-group

Chapter 3

Theory

This chapter is intended to explain all the theoretical fundamentals involved in the project. The chapter aims to give the reader a good foundation for understanding the theory behind the implementation of the system and acts as a lead-in before describing the actual implementation.

The chapter starts with a description of the channel, and the characteristics prominent for this system. This is followed by a description of the modulation, the multiplexing and how they will be conducted. Finally, a couple of subsections follow, explaining important theoretical parts of the system as well as the concepts of channel coding.

3.1 Channel characteristics

There are several traits affecting the signal when it propagates through the channel. In short, the effects that are of interest for the system are: Additive noise, phase noise, frequency offset and signal fading. Phase noise can be observed as rapid, random fluctuations in phase on the waveform. Frequency offset is the absolute difference in clock frequency on the local oscillators involved in the transmission. Signal fading is a change in attenuation, as the signal propagates through the medium, from the transmitter to the receiver [Emami, 2013].

3.1.1 Noise

In any communications system, noise appears and distorts the signal. Noise may arise from a variety of different sources. One such source are the components of the receiver and the transmitter, which introduce thermal noise [Proakis, 2007]. This is often the

most powerful source of error. Another important noise contributor can be the signals from other systems, emitting on the same frequencies, i.e. interference. Whether this problem is likely to occur or not is very situation specific. For a 60 GHz system, it should be unlikely to see interference from other systems.

For the OFDM system, one may also find an interference-phenomenon within the system, as information in one carrier can leak and cause interference in another carrier. This leakage is also added to the noise problem. Several ingenious methods can be used in order to cancel the noise and to correct the errors that it introduces. A selection of these methods will be covered later on in this report.

The theoretically achievable data rate for an additive white gaussian noise channel is determined by:

$$C = B \log(1 + SNR(B)) \quad (3.1)$$

where C is the channel capacity in bits per second, B is the bandwidth in hertz and SNR is the signal-to-noise-ratio of the average received power and the average noise power in the receiver. The expression in 3.1 implies that by decreasing the influence of noise, it is possible to raise the capacity of the channel without increasing the received power or allocating more bandwidth [Proakis, 2007].

3.1.2 Signal fading

If one were to send out a short pulse over a multipath channel, one could, in theory, see that the received signal was a pulse train. Copies of the pulse arrive with a delay, after taking a longer route to the final destination. The received pulses differ in amplitude and phase, because of the different paths taken by the incoming versions of the original pulse. All of the different paths contribute with different characteristic features and manipulate their respective version of the pulse in some unique way. The multipath channel is also time varying, meaning that the received train has constantly changing characteristics, caused by changing propagation delay and attenuation factor. The reasons for this may be movements of the involved devices, in relationship to each other or to the surrounding environment. Additionally, there is a frequency selective behaviour, where the multipath distortion causes some frequencies to be more severely affected than others [Proakis, 2007]. This has interesting implications in an OFDM system, as we may transmit in parallel in a narrowband channel, where we can consider the

channel constant and thus, there is no distortion. Furthermore if channel information is available at the transmitter: one can transmit more complex information over the subcarriers with low attenuation, and less complex information (if any information at all), over more severely attenuated subcarriers.

When transmitting actual data, the multipath problem may lead to intersymbol-interference (ISI), meaning that the beginning of a symbol is contaminated by delayed versions of the last transmitted symbol, creating an overlap in the time domain. In an indoor-environment, multipath distortion is always present and changes dynamically. One way of handling the distortion is to increase the amount of power sent with a fade margin, usually around 20-30dB. This can reduce the distortion as much as 30dB of the signal power [Huang and Wang, 2011].

Since the wavelength is small, the problem of diffraction is rare. This is because there are only few objects that are of the same size as the wavelength, and that are able to cause diffraction. However, this makes the multipath propagation more brutal, since waves will bounce off instead of diffract [Emami, 2013].

3.1.3 Path loss and Shadow loss

Looking at 60 GHz transmission, it can be observed that the path loss is much more severe than for systems both with a lower and a higher carrier frequency. This effect is caused by oxygen attenuation [Yong et al., 2011]. Millimeter-waves are affected by oxygen and the frequencies affected the most are the 60GHz-waves. Mm-waves are also critically attenuated by rain. The magnitude of the attenuation is related to the intensity of the rain and a decrease of the signal power in the magnitude of a couple of decibels can be expected [Yong et al., 2011]. However, in an indoor environment, the relatively minuscule delay spread will result in delayed components actually contributing in a constructive manner, to the signal quality. This means that one can observe path loss exponents of 2 or less, for a normal indoor setup [Moraitis and Constantinou, 2002]. Apart from this, the system is still susceptible to shadowing and moving bodies.

Another fact that needs to be recognized is the loss of power due to the high frequency. Friis transmission equation 3.2 gives the path loss between two isotropic antennas in free space. d is the distance between transmitter and receiver.

$$L = 20 \log_{10}\left(\frac{4\pi d}{\lambda}\right) \text{ (dB)} \quad (3.2)$$

Looking at the equation, the difference between transmitting at 5 GHz and at 60 GHz presents an additional loss of about 22 dB [Huang and Wang, 2011].

3.1.4 Local Oscillators

In a communications system, a local oscillator is used to up-convert the baseband-signal frequency to the carrier frequency. Upon reception, the signal needs to be down-converted to baseband frequency and this is done by another local oscillator, on the receiver side. These procedures raise several issues, since the oscillators are never completely synchronized in time. We call this the local oscillator offset (LO) and it is often conceptualized as $LO_{offset} = f_{LO_{receiver}} - f_{LO_{transmitter}}$, the difference between the local oscillator frequency of the transmitter and the receiver. In OFDM, it is especially important to correct the phase and frequency offset, in order not to have the different subcarriers interfering with each other [Schmidt et al., 2004].

3.2 Modulation

To be able to send data reliably and efficiently, over a channel, the data needs to be modulated in a proper way, to ensure that the channel has as little effect as possible on the signal. This chapter will bring up modulation schemes relevant to this project.

The way the data bits of the signal are represented, after modulation, is usually visualized by plotting the points in a constellation diagram. An increased distance between the points means that the modulation has a higher resilience to noise. An increased distance between a point and the origin implies more power in that point. The goal of an efficient constellation is thus to maximize the distance between points, while minimizing the mean distance of the points, to the origin [Proakis, 2007].

3.2.1 Phase Shift Keying

Phase Shift Keying (PSK) is a modulation method that, as the name implies, shifts the phase of the carrier wave. For the simplest version, Binary-PSK, the phase is shifted 180 degrees, generating the constellation diagram shown in figure 3.1a.

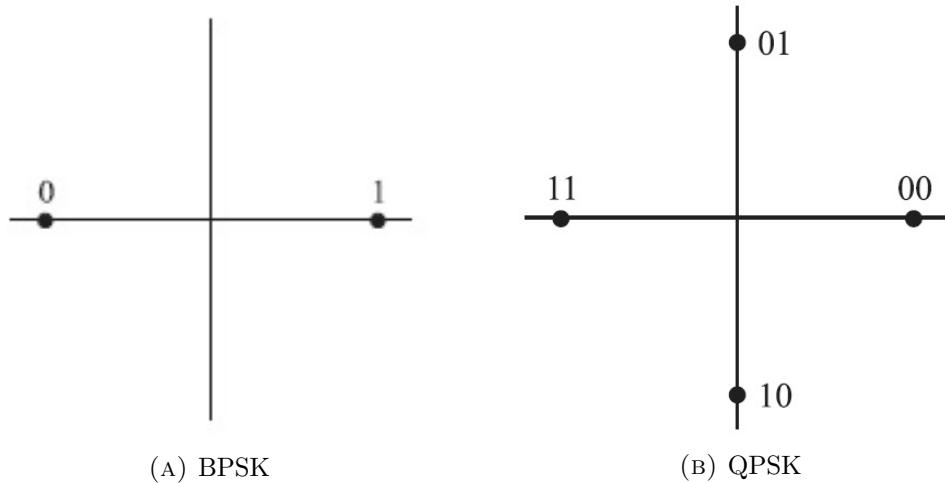


FIGURE 3.1: Constellation diagram of BPSK and QPSK [Proakis, 2007].

It can be seen that BPSK ensures the maximum distance between the points, making the transmission highly robust to noise. An inherent advantage of this scheme is that one would only need to estimate the phase, to be able to decode the transmission successfully. The cost of this robustness is that the carrier is only able to modulate 1 bit per symbol, which limits the possible data rate. Adding one more phase shift to the carrier, or putting two BPSK-carriers together gives Quadrature-PSK (QPSK). A QPSK constellation diagram can be seen in figure 3.1b and can transmit 2 bits per symbol at the cost of a smaller distance between the modulated points, implying increased noise sensitivity [Proakis, 2007].

3.2.2 Amplitude Modulation

Instead of modulating bits onto symbols by varying the phase as in PSK one can do modulation by varying the amplitude. In Pulse Amplitude Modulation, PAM, modulation of the signal is done by alternating the amplitude in this way. Several levels of amplitude change can be allowed, as can be seen in figure 3.2, thus increasing the bits per symbol, but simultaneously making the system more susceptible to noise [Proakis, 2007].

To make use of amplitude modulation, one would have to estimate the gain of the received signal and if using algebraic amplitude values, one would also need to estimate the phase.

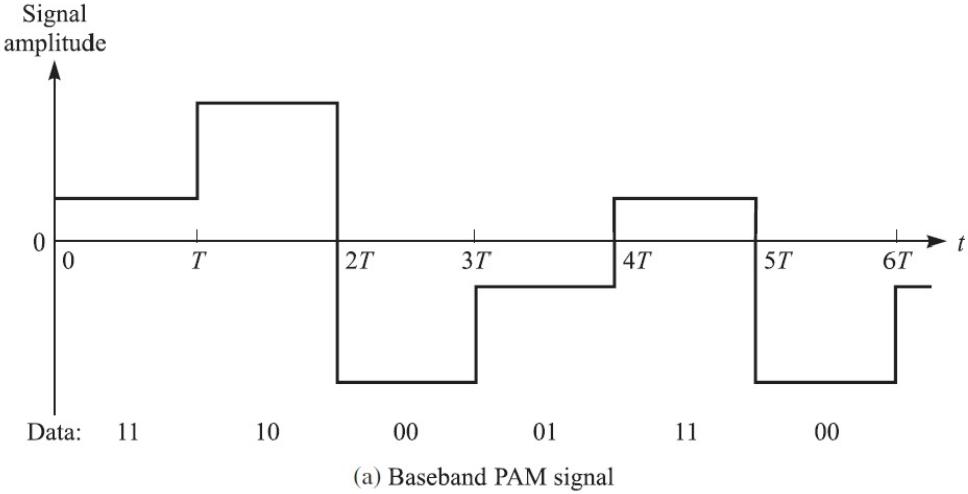


FIGURE 3.2: PAM signal with 4 different levels of amplitude [Proakis, 2007].

3.2.3 Quadrature Amplitude Modulation

A version of combining both phase and amplitude modulation is Quadrature Amplitude Modulation (QAM), meaning that we work with both amplitude and phase when modulating data. The waveform is illustrated in equation 3.3.

$$s_m(t) = A_{mi}g(t)\cos(2\pi f_c t) - A_{mq}g(t)\sin(2\pi f_c t), m = 1, 2, \dots, M \quad (3.3)$$

It can be seen that by changing A_{mi} A_{mq} , it is possible to modulate the amplitude and the phase of the signal. By using more levels for both parameters, we can modulate more bits into each symbol. The most frequently used options include 16QAM, 64QAM and 256QAM [Proakis, 2007], but for an arbitrarily high SNR it is theoretically possible to go towards infinitely many constellation points, putting more and more bits into every symbol.

Gray coding minimizes the bit error rate of the constellation and 16QAM modulates the signal on 16 symbols, thereby enabling 4 bits per symbols. Gray coded 16QAM is visualized in figure 3.3. Gray coding is a way of ensuring that adjacent symbols only differ by one bit [Proakis, 2007].

Moreover, it is possible to design more complex constellations using any thinkable pattern. One such example is the honeycomb constellation used in an implementation of System 2 (6). The objective of such an alternative design can be increasing the power efficiency, however these designs are often marred with problems such as difficulties in finding a gray code or problems related to turbo coding.

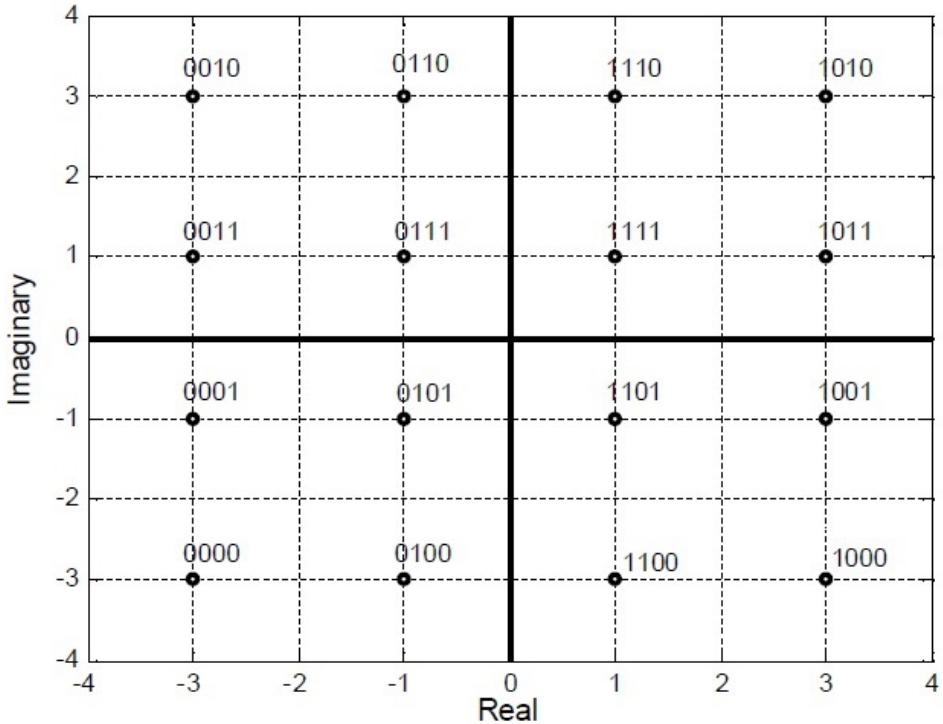


FIGURE 3.3: Constellation of 16QAM [Islam et al., 2011].

3.3 Orthogonal Frequency Division Multiplexing

Recently, Orthogonal-Frequency-Division-Multiplex (OFDM) has become to be the multiplexer of choice for high data rate wireless systems. It has been adopted as the most favoured technology in several IEEE-standards such as 802.11a, Wi-Max as well the LTE-system [Yong et al., 2011].

OFDM is a method of dividing the available bandwidth into a set of subcarriers. Each subcarrier can be individually modulated with, for example, QPSK. An advantage of this setup is that, for a wide channel, it is possible to put more data on the more reliable subcarriers, while putting less on the subcarriers with high attenuation.

A part of the available bandwidth is allocated to each subcarrier. However, subcarrier bandwidth will partly overlap with neighbouring subcarriers. This can be allowed since the carriers are mainly orthogonal to each other[Proakis, 2007].

An important part of the any OFDM system is the cyclic prefix. The cyclic prefix is basically a copy of the last section of the OFDM symbol, appended in the beginning

as a prefix (as the name suggests). By doing so, it is possible to replace the linear convolution:

$$y[m] = \sum_{l=0}^{L-1} h[l]x[m-l] \quad (3.4)$$

where L is the channel length, $x[m]$ is the transmitted signal in discrete time and $y[m]$ is the received signal in discrete time, with the circular convolution:

$$\hat{y}[m] = \sum_{l=0}^{N-1} h[l]x[(m-l)modN] \quad (3.5)$$

where N is the length of the OFDM symbol. This corresponds to doing the filtering in the frequency domain instead of in the time domain, which saves effort and computational load for the channel equalization.

OFDM can be chosen as the suitable scheme for a 60 GHz communication system when it is important to eliminate multipath distortion and when the power efficiency of the transmitter is not of significance [Yong et al., 2011].

3.3.1 Interleaver

Errors in communication systems often appear in bursts. This can be a problem when using forward error correcting codes, such as turbo codes or LDPC codes, since the error correcting capability is challenged by series of incorrect readings. To avoid this, the data is interleaved, mixing the bits from different places in the code, according to a predefined pattern, and then putting them back in their original place at the receiver, inverting the original operation. As a result, burst errors are spread throughout the signal in a more uniform manner. Since sporadic errors are easier to handle with error detection or error correction codes, this increases the performance of the system [Proakis, 2007].

3.3.2 Implementation tricks

When implementing OFDM, several issues need to be solved in order to design a system that is robust to the channel characteristics of 60 GHz communication, as well as efficient and reliable.

To resolve the problem of ISI, which in our case is probably a result of resonance in the circuits at the transmitter and the receiver rather than multipath propagation, a cyclic prefix is added to each OFDM symbol. The length of the cyclic prefix should be chosen to be longer (in time, so to speak) than the number of taps (minus 1) used to model the channel. In this manner, the receiver can make out what part of the received signal is tainted with ISI, and also what the signal should be interpreted as. If the channel is causal, a prefix is enough. If the channel has non-causal tendencies, as was observed in our case, it will also be useful to add a cyclic suffix, meaning that the OFDM symbol is tailed by a copy of the first section of that same OFDM symbol.

Another advantage to using a cyclic prefix is that the repetition of information can be used to estimate some of the channel related parameters, such as LO-offset.

A way of handling the synchronization between transmitter and receiver is to use pilots. Pilots work like training sequences, consisting of symbols known by the receiver. These sequences are used in OFDM and help the system stay in sync as well as estimate changes in the phase and frequency offset caused by the channel [Shen and Martinez, 2006]. In a scenario where the channel is slowly varying, it can be sufficient to estimate the LO offset, phase and channel only in the beginning of a packet, whereas a channel that varies faster may require tracking of these parameters.

The traits that make OFDM an appropriate choice for the project is its relative robustness to the distortion introduced by the channel and the easy encoding. The weaknesses that come along with OFDM and that have been the biggest issues for the project are the phase and frequency offsets, as well as the Peak-to-Average-Power-Ratio, PAPR. These problems are solved by good synchronization, through the use of pilots.

3.3.3 Packet detection

One way of handling the packet detection is by using the Schmidl-Cox [Schmidl and Cox, 1997] scheme for packet detection and frequency offset estimation. This method uses a lightweight algorithm where every packet is preceded by one or two systematically designed OFDM symbols, depending on the application. If the only objective is to identify an incoming packet, there is no need for a second symbol. Also, it is possible to detect a small LO offset using only one symbol, but for arbitrarily large offsets one has to use two symbols. For this project, only the first symbol was implemented and tested. A feature that was never thoroughly investigated, but which is possible, is coarse SNR

estimation using the first symbol. In a duplex system, this information could be used to guide the choice of constellation.

3.3.4 DFT precoding

Another drawback of OFDM modulation is the problem of high peak-to-power-ratio (PAPR). When transmitting, we ideally wish to have a signal with as low PAPR as possible. This quality reduces the problem of amplifier saturation, which leads to nonlinearities in the signal. To mitigate the problem, we can resolve to a number of methods. Reducing overall signal power in a linear fashion, clipping of the signal, clipping and filtering (to avoid the problem of spectral leakage, caused by clipping), constellation adaption, nonlinear companding transforms, and many more. Clipping is a computationally cheap method, while many of the others require extensive computation. A popular method used in conjunction with clipping is that of DFT precoding, presented and tested in [Aboul-Dahab et al., 2013], which means that the constellation is transformed into the frequency domain before the IFFT of the OFDM. Depending on the number of used carriers for the OFDM and the order of the DFT, the total operations of DFT and IFFT can more or less cancel out. The advantage of this technique is a linear transform that always reduces the BER for a certain normalized clipping ratio (CR), which is defined as predefined clip level by the maximum of the signal.

The implementation of the DFT in the system was done by using a predefined DFT transform:

$$P = \begin{bmatrix} p_{00} & p_{01} & \dots & p_{0(N-1)} \\ p_{10} & p_{11} & \dots & p_{1(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{(N-1)0} & p_{(N-1)1} & \dots & p_{(N-1)(N-1)} \end{bmatrix}$$

where $p_{mn} = e^{-j2\pi mn/N}$, m and n are integers from 0 to $N - 1$ and N is the order of the DFT used.

The signal before the OFDM IFFT, X_n , is multiplied window by window with the matrix P , meaning that the time domain signal will be generated by $x_n = IFFT(P * X)$ and restoration of the data is done through $\hat{X} = P^{-1} * FFT(\hat{x})$, where \hat{x} is the received signal after equalization on synchronization, P^{-1} is the inverse of P and \hat{X} is the restored

constellation. The number of symbols that are to be DFT precoded should be an integer multiple of the DFT order.

3.4 Fast Fourier Transform (FFT)

Using the Discrete Fourier Transform (DFT) transfers a time-discrete periodic signal into the frequency domain by producing a finite signal in the frequency domain, that consists of discrete values and that is periodic. This can be seen as a way of fragmenting a set of discrete periodic values into its corresponding periodic frequency parts [Reay, 2012].

The FFT is widely used within computing all over the world. It is used to efficiently compute the DFT and it's inverse, by using ingenious mathematical methods. What the FFT does is to break down the DFT into smaller parts, computing the DFT of the smaller parts recursively. By doing this the computation is divided into smaller and smaller segments, decomposing the original single DFT into a large number of smaller DFTs. What it effectively does is reducing the order of computation, since computing the DFT of an N-point signal gives rise to computational complexity in the magnitude of $O(N^2)$ operations while an FFT lowers that magnitude to $O(N \log(N))$ [Diniz et al., 2010].

3.5 Kalman filter

In a noisy channel with all of the aforementioned problems of varying LO offset and phase, a Kalman filter is used to estimate the parameters of the channel. The filter uses a recursive algorithm producing an optimal estimation of the channel. It has two states, prediction and correction; the prediction step uses the incoming data to predict the parameters of the channel and the correction takes the new incoming bits, weighs them according to their certainty and then corrects the prediction [Kalman, 1960].

3.6 Channel coding

To mitigate noise and distortions of the channel, redundant bits are introduced into the signal to facilitate detection and correction of erroneous bits. Within this project, focus is laid upon working with a cyclic prefix (described in the OFDM chapter 3.3) and a couple of different Forward Error Correction codes.

3.6.1 Convolutional codes

When convolutional codes were invented, they provided a major breakthrough in communications theory, suddenly enabling systems to get significantly closer to the Shannon rate. Convolutional coding works with the help of a finite state machine, producing codewords for the incoming bits that depend on the current bits as well as previous bits. A figure of a convolutional encoder is shown in 3.4.

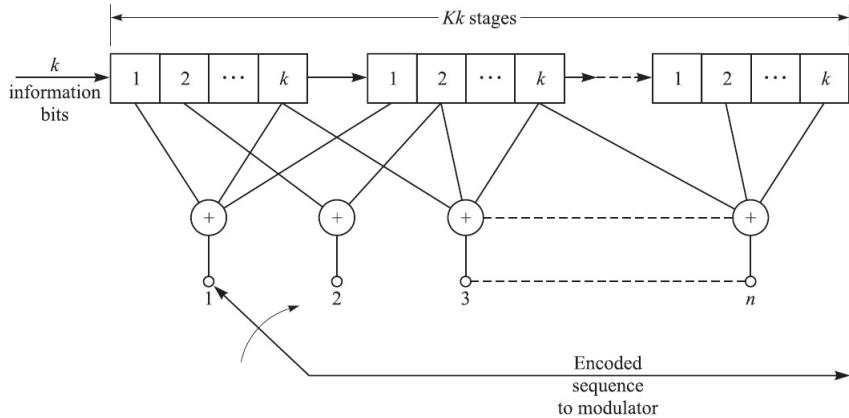


FIGURE 3.4: Example of a convolutional encoder [Proakis, 2007]

3.6.2 Turbo codes

In order to solve the problem of handling too long codes when trying to reach the Shannon channel capacity limit, turbo codes use two convolutional encoders on the transmitter side and two decoders on the receiver side. In between the encoders is an interleaver, introducing more randomness in the code, pushing the data points further away from each other and decreasing the BER. Instead of using a decoder that decodes the symbols by simply determining which points they are closest to and changing it into that point (hard decoding) the decoders used in turbo coding uses the probability of how likely it is that a received point corresponds to a point in the decoding scheme. Via log-likelihood ratios the decoder outputs the decoded points and the probability that they are correct (soft decoding). With two decoders looking at different bits, a joint estimation is done. This further increases the chance of correct detection. The process is an iterative one that constantly increases the probability using the joint detection until the detection is perfected [Guizzo, 2004].

3.7 Antenna

Antenna technology is an area that is considered to be outside of this project and the only studies that will be made are how the distance between the transmitter and receiver antenna affects the received signal power. Further studies should look at how directionality of the antennas affect the performance of the system, since the type of antenna profoundly affects the performance of the system.

Chapter 4

Technical specification

In this project the Universal Software Radio Peripheral, USRP, hardware was used, more specifically the USRP N210 which is developed and manufactured by Ettus Research Products. This chapter serves as a presentation of the hardware, and it's corresponding software, used to interact with the 5 GHz and 60 GHz bands. It's purpose is to give the reader a conceptual understanding of the hardware, software and their respective use in this project. The entire chapter has been written with the help of the specification of Ettus Research Products found on their webpage www.ettus.com and with the help of the USRP lecture held by Per Zetterberg at KTH on the 24th of March 2014.

4.1 The USRP

The USRP is a family of boards designed for RF applications from DC to 6 GHz, including multiple antenna (MIMO) systems. For the current purpose, the N210 (networking series) has been analyzed. In the 60 GHz system, an extended set of components were added to gain access to the 60 GHz spectrum, but the N210 was still used.



FIGURE 4.1: USRP N210 (Same interface as N200).

4.1.1 N210 technical description

The N210 has the best performance within the Ettus USRP family and it is suitable for complex applications requiring wide bandwidths and high performance, such as 60 GHz wireless transmission. The product consists of a Xilinx Spartan 3A-DSP 3400 FPGA, 100 MS/s dual ADC, 400 MS/s dual DAC and Gigabit Ethernet connectivity. The device has the ability to stream up to 50 million samples per second. However, to limit round off errors, we will use 16 bits numbers and so be limited to 25 million samples per second.

4.1.1.1 Features

- Use with GNU Radio;
- Modular Architecture: DC-6 GHz;
- Dual 100 MS/s, 14-bit ADC;
- Dual 400 MS/s, 16-bit DAC;
- Up to 50 MS/s Gigabit Ethernet Streaming;
- Spartan 3A-DSP 3400 FPGA;
- 1 MB High-Speed SRAM.

In figure 4.2 it is possible to see the detailed specification of the N210 board.

Spec	Typ.	Unit	Spec	Typ.	Unit
POWER					
DC Input	6	V	SSB/LO Suppression	35/50	dBc
Current Consumption	1.3	A	Phase Noise (1.8 Ghz)		
w/ WBX Daughterboard	2.3	A	10 kHz	-80	dBc/Hz
CONVERSION PERFORMANCE AND CLOCKS					
ADC Sample Rate	100	MS/s	100 kHz	-100	dBc/Hz
ADC Resolution	14	bits	1 MHz	-137	dBc/Hz
ADC Wideband SFDR	88	dBc	Power Output	15	dBm
DAC Sample Rate	400	MS/s	IIP3	0	dBm
DAC Resolution	16	bits	Receive Noise Figure	5	dB
DAC Wideband SFDR	80	dBc	PHYSICAL		
Host Sample Rate (8b/16b)	50/25	MS/s	Operating Temperature	0 to 55°	C
Frequency Accuracy	2.5	ppm	Dimensions (l x w x h)	22x16x5	cm
w/ GPSDO Reference	0.01	ppm	Weight	1.2	kg

FIGURE 4.2: USRP N210 detailed specifications.

4.1.2 The millimeterwave transmitter and receiver

In order to transmit and receive at 60GHz, additional boards are required. These are the Hittite HMC6000LP711E and HMC6001LP711E for the transmitter and receiver respectively.¹

These are integrated complete mmWave IC and low profile antenna. They provide 23.5 dBm of EIRP operating over 57 - 64 GHz with 1.8 GHz of modulation bandwidth. The interface consists of a universal analog baseband IQ. The receiver includes an LNA, image reject filter, RF to IF downconverter, IF filter, I/Q downconverter, and frequency synthesizer.

The data sheets are available on www.hittite.com and the main features are presented in the following sections.

4.1.2.1 Transmitter's features

- Support for IEEE Channel Plan;
- EIRP: 23.5 dBm;
- Output Power: 16 dBm;
- Antenna Gain: 7.5 dBi;
- Max Gain: 38 dB;
- Gain Control Range: 17 dB;
- Integrated Frequency Synthesizer;
- Integrated Image Reject Filter;
- Programmable IF Gain Block;
- Universal Analog I/Q Baseband Interface;
- Three-Wire Serial Digital Interface;
- 7x11mm QFN Package: 77 mm².

¹<https://www.hittite.com/products/index.html/category/395>

4.1.2.2 Receiver's features

- Support for IEEE channel Plan;
- Receiver Gain: 2 - 67 dB;
- Noise Figure: 7.0 dB;
- Integrated Low Profile antenna: 7.5 dBi;
- Integrated image Reject Filter;
- Integrated Frequency Synthesizer;
- Programmable IF and Baseband Gain Blocks;
- Universal analog I/Q Baseband interface;
- Integrated AM and FM Demodulator;
- Three-Wire Serial Digital;
- Interface 60 Lead 7x11 mm SMT Package: 77 mm^2 .

4.2 Description of software and functions

The Ettus USRP is delivered with the USRP Hardware Driver (UHD). UHD provides the user with a host driver as well as an API² for all of Ettus Research products. UHD can work alone or with third-party applications. For this project the software was developed in C++.

To be able to understand and to use the product, as well as to be able to build a robust and efficient system, a basic understanding of the classes, objects and functions supplied with the USRP is crucial. The aim of this section is to provide guidelines, to make it clear what functions are important and what they do. To get a more thorough explanation of the USRP software, we refer to the manual.

4.2.1 USRP Class

The USRP class is the one which allows the program to create an interface with the USRP hardware and that manages the data exchanged with it through the Ethernet cable. One of its subclasses is the `multi_usrp`, which was used in this project. It is a very useful class since it handles both Tx, Rx and multiple USRP's.

²http://files.ettus.com/uhd_docs/doxygen/html/index.html

4.2.2 Multi-USRP Class

This class facilitates ease-of-use for most scenarios. It provides convenient functions to tune the device, the daughterboard, set the board gains, antennas, filters, and other properties. This class can be used to interact with a single USRP with one or more channels (in this project it was limited to one channel), or multiple USRP's in a homogeneous setup. All variables take optional parameters for board number or channel number.

4.2.3 Device Pointer

A device pointer can be used to refer to an object of type `uhd::usrp::multi_usrp::sptr`. It creates a pointer to a device object which contains the fields and functions managing the data-flow between the computer and the USRP board.

4.2.3.1 USRP Address

The USRP interface is managed through the use of an Ethernet cable. The address of the USRP is fixed and set to `192.168.10.2` (local IP). In order to change the address and initialize, the object uses `make(address)` in the scope of the class. This will return the pointer to the created object.

4.2.4 USRP Clock

The USRP is equipped with a clock. The values of the clock can be changed with the functions supplied in `multi_usrp`, for example `set_clock_source`. Some parameters can also be altered with the help of the structure `clock_config_t`, although it is more favorable to use the aforementioned set time/clock source calls.

4.2.5 USRP Parameters

The USRP parameters are easily set. It is done by calling the functions, using the device pointer. By doing so, we can choose the different parameters necessary for the transmission. As an example, the rate is set by calling `set_xx_rate`. Another example is setting of the Rx/Tx (xx) antenna value, which one would do by calling `set_xx_antenna`.

4.2.6 USRP RF Setting

In the UHD library, the LO frequency is set with structures called `tune_result_t` and `tune_request_t`. They allow the host to tune the RF chain and to check the value that was set by the USRP device, since it may not be able to use the value at the requested frequency.

To get the serial name, the function `get_usrp_xx_info` can be used. This returns strings with the name and serial number of the USRP.

4.2.7 USRP tx_stream/rx_stream

The Tx/Rx streamer is the host interface which transmits samples. It represents the layer between the samples on the host and the samples inside the device's DSP (Spartan FPGA in N210) and it is used to transmit and receive data from the USRP.

The TX and RX stream pointers (pointing to single, or to a collection of send/receive buffers) have to be initialized with `get_tx_stream()` and with proper structure `stream_args_t` as argument. The structures contain information about the CPU format, format over-the-wire and the channels (for multi-channel).

After initializing the stream, it is possible to ask the USRP to send or receive data over the (xx) streamer pointer. The functions that do this are `send` and `recv`. These functions receive a pointer to a buffer as argument, the size of a buffer, a `xx_metadata_t` structure and a time out.

4.2.7.1 metadata_t

Structure that describes the transmitted or received package distribution. It includes time specification, fragmentation flags, burst flags, and error codes.

4.2.7.2 stream_cmd_t

In order to set the stream, it is also necessary to provide the device with a stream command. This is a command to configure and control a stream. For example, one possible command is `STREAM_MODE_START_CONTINUOUS`, which tells the device to stream samples indefinitely.

Chapter 5

Single-Carrier system

This chapter aims to present the results of the first implementation done in C++. The first implementation is a basic single-carrier communication system that was implemented using the USRP hardware in a real-time communication environment. The objective of implementing this system was to acquire hands-on experience, by writing software for a real-time system, with USRP's as RF interfaces between two ends.

This system implementation transmitted at 5 GHz, rather than the 60 GHz that was the actual target. The reason was the delays in acquisition of 60 GHz hardware. However, the system was developed to be able to cope with low SNR levels, be able to work in presence of phase noise and with a large and slowly varying frequency offset, which was expected to be the three most important factors that would differ between 5 GHz and 60 GHz. It gave us the necessary training to tackle more complex systems.

Firstly, the global theoretical ideas are covered, followed by some considerations about the real implementation. Finally, the performance of the system is analyzed, some results are presented and the main lessons learned are summarized.

5.1 Transmitter

The transmitted sequence used in the first implementation of the real-time transmission consists of 10 samples of zeros, so called guard bits, followed by 100 complex samples of training sequence and then 9380 complex samples of data. In this implementation, 4-QAM modulation is used, therefore, each transmission contains 18760 information bits. This sequence is then upsampled with a rectangular pulse of length 4, and thus giving a

total of 37960 complex samples sent. The block scheme of the transmitter can be seen in figure 5.1.

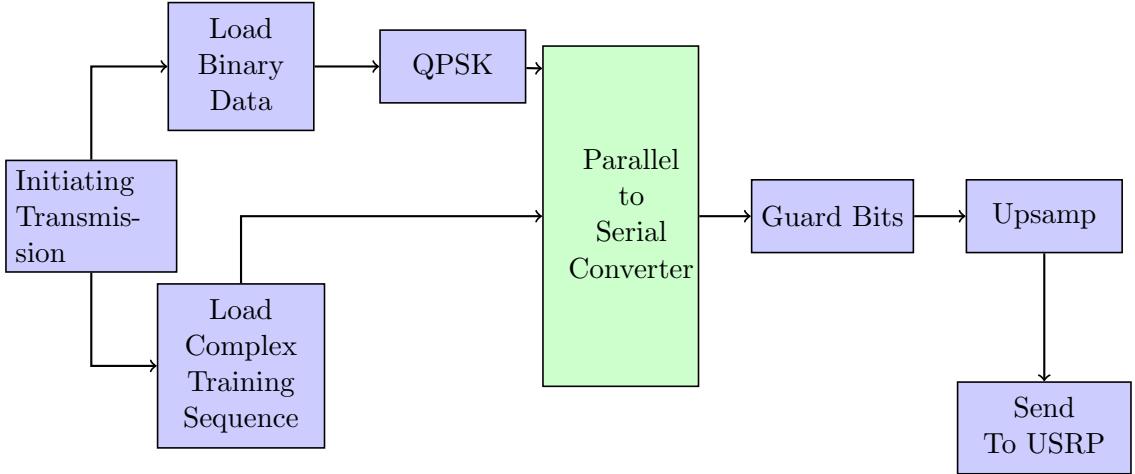
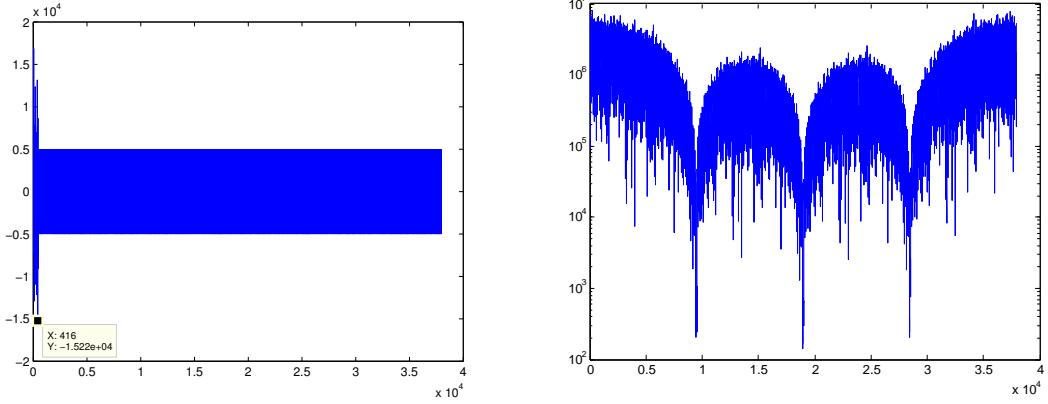


FIGURE 5.1: Block Diagram of the Transmitter

After generating the sequence to be transmitted, we pass it to the USRP, which takes care of sending it through the wireless medium. Both real part and spectrum of the sequence sent in one experiment can be seen in figure 5.2. This is a very simple, single-carrier transmission system.



(A) Real part of transmitted sequence.

(B) Spectrum of transmitted sequence.

FIGURE 5.2: Analysis of transmitted sequence.

5.2 Receiver

The block scheme is depicted in figure 5.3. A detailed description and other consideration follow.

After receiving the sequence we need to find the best sampling time. This is done using the crosscorrelation between the received sequence and the training sequence. Here,

problems with the LO-offset may arise. The frequency offset influences the crosscorrelation by destroying the peak at the optimal sampling time, if the rotation in phase is bigger than $\pi/2$. To avoid this problem, the frequency offset is removed before the correlation using the method described in section 5.2.1. It mainly consists in taking the advantage of the LO leakage in the received signal by taking the difference between the two peaks in the spectrum. Usually the peak with the largest amplitude corresponds to the receiver and the smaller one corresponds to the transmitter. However, this is not necessarily the case. In order to avoid sign problems it is favorable to increase the frequency of one of them, to make known in advance which one has the higher frequency. This is done in order to have no doubt about the sign of the frequency offset. Once it has been removed, there is no risk of getting the wrong sampling time, when correlating with the training sequence.

The next steps are the matched filtering of the received sequence and the downsampling. As the channel impulse response is not a Dirac pulse, it is possible to, instead of doing a simple downsampling, make an equalization. However, as this system has been designed to be simple, this relatively complicated method is not used. Moreover, as the upsampling coefficient is quite high (4), the inter-symbol interference is not a problem. After downsampling, it is possible to do hard detection.

Even after the frequency offset has been removed, the phase is still varying almost from $\pi/4$ to $\pi/2$ during the whole transmission. This is a problem for QPSK and it creates some errors at the end of the transmission. In order to counter this, the phase offset needs to be tracked during the transmission. This was done using a Kalman filter.

The Kalman filter uses a random walk model to estimate the phase. The model is initialized using the phase from the crosscorrelation estimation. Then the phase between the received training sequence and the actual training sequence is used to get a better estimate of the phase at the last sampling time of the training sequence. After this, the Kalman filter continues working using the assumption of a QPSK mapping. The phase is measured by considering that the decision about the current data, using the previous phase estimate, is correct. By doing this, it is possible to get a new estimate of the phase. Since the variation of the phase and the noise level are low, these methods can be used successfully.

To avoid problems in the tracking, the covariance of the measurement noise should be quite high, reducing the impact of a faulty decision. A different approach is the

use of $\hat{\phi}_{n+1} = E(\phi_{n+1}|\hat{\phi}_n)$ which can be calculated using an additive white Gaussian noise model and calculating the variance of the noise before the training sequence. This approach was not used, however, due to its complexity.

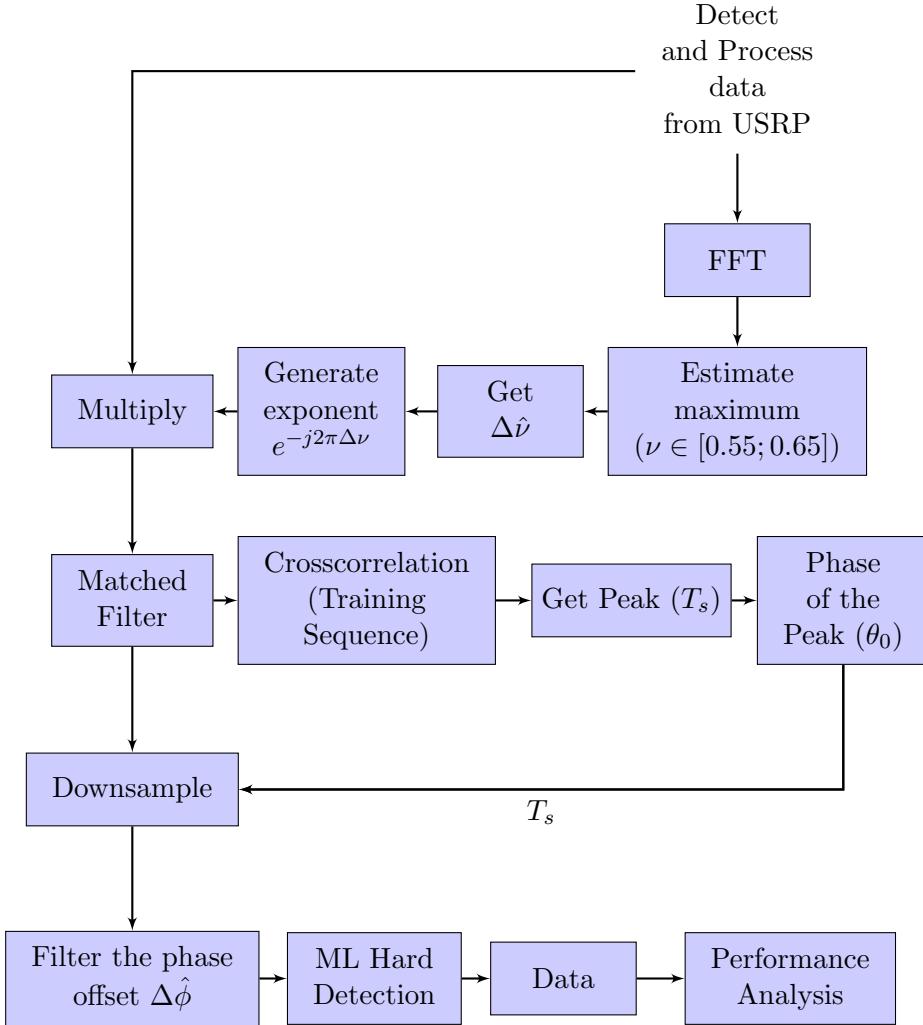


FIGURE 5.3: Block Diagram of the Receiver.

5.2.1 Local Oscillators revisited

Within the implementations, local oscillators are used to upconvert the modulated signal to the frequency upon which it is to be sent, in our case 5 GHz and 60 GHz. The implications of this, and the problems related to it, were covered in 3.1.4.

Figure 5.4 depicts the LO leakages found in a received sequence. In this early implementation, the leakages were assumed to represent the absolute local oscillator frequency and not being time varying. This means that it was possible to identify the offset just by considering the difference between the two identified peaks. In the ideal case, these peaks should overlap. The normalized difference is the offset that we want to estimate.

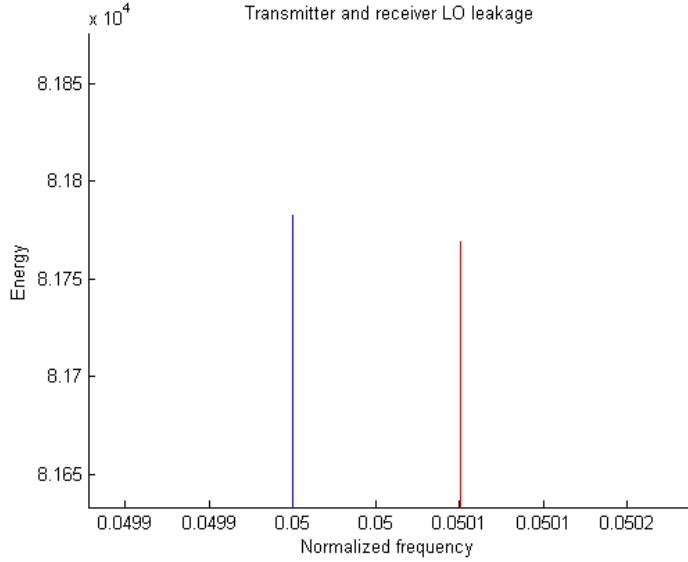


FIGURE 5.4: Local oscillator simulation leakage

In this case, the simulation was run with an LO offset of $2 * 10^{-4}$, meaning that the transmitter's LO ran at a frequency 0.2 % faster than the LO at the receiver. The described method identified an LO offset of $1.9996 * 10^{-4}$. This method for identifying the LO offset is suitable only under conditions that there exists an observable leakage and that there are no real-time constraints to take into consideration (since it requires some computation and preferably as much data as possible).

5.3 Real-time Implementation

There is no difficulty in simulation of the system described above in Matlab. However, when as real-time implementation is required, several problems with hardware, software, interface, and handling appear. The system was implemented based on the C++ API and a provided skeleton code to communicate with the USRP's used (model 2 and N210). This section describes the used designs and shows the results obtained in a C++ to C++ only transmission. Note that all the Matlab plots which are shown here are using the data from binary files written during the real-time processing in C++.

5.3.1 Transmission

In order to recreate the transmitted sequence in C++, binary data (of a predefined length) was loaded from a file, modulated and put together with the complex training sequence. The sequence was sent to the USRP, which was done only once and in a single function call. Due to the large size of the sequence, sporadic underflow errors

occurred. In this system, for the underflow problem, we re-transmit again when the problem appears. In order to completely avoid the problem, we send smaller blocks of the sequence to the USRP. We conjugated the sequence right before it was sent in order to obtain the correct results, since the same operation is performed in the receiver.

5.3.2 Reception

To detect the transmission, the C++ implementation always listens to the channel and detects a transmission based on the power received in each buffer. After waiting for the transient regime of the USRP to vanish, which is achieved just by leaving a number of initial symbols unused, the receiver program detects and processes the signal as described above. The implemented thread system allows us to read the channel and process several transmissions in parallel, which is critical for our real-time system since the system will require some computations to be done while data is still being received, to be able to meet the real-time requirements.

In more detail, one thread is in charge of collecting the data from the USRP (usrpT). Another one is used, to analyze the content of the received buffer and, if power is detected, build the storage buffer and launch an additional thread to decode the data.

5.3.2.1 The USRP thread (usrpT)

This thread is the most critical one in terms of priority. All unnecessary serial operations have been eliminated in order to assure the highest responsiveness possible from the computer, with regard to the USRP. The pseudocode is the following:

```

loop
    Ask data to the USRP
    Push the newly received buffer into a queue (usrpQ)
    Operation V on the semaphore
5: end loop
```

5.3.2.2 The detection thread (detectionT)

The detection thread tests the contents of the buffer and launches a processing thread if necessary.

```

loop
    if Power has been detected then
```

```

if Has not yet received all the data then
    Storage buffer ← + the current buffer
5:    usrpQ.pop()
        Operation P on the semaphor
        Current buffer ← usrpQ.front()
else
    No more detected power
10:   Launch a processing thread with the storage buffer
end if

else
    Operation P on the semaphore
    Current buffer ← usrpQ.front()
15:   Test the power in the current buffer
if Power is detected then
    Power has been detected
else
    Delete the current element
20:   end if
end if
end loop

```

5.3.2.3 The processing threads

The processing threads are launched by detectionT. There can be an arbitrarily large number of them in parallel and all the decoding operations are performed on the storage buffer, according to the description given in the previous sections.

The program was given a real-time priority, a minimum nice value and the scheduling priority of the USRP was set to the maximum. The non thread-safe operations are protected by a mutual exclusion object, mutex.

5.3.3 Received Sequence

In the received sequence (obtained with the procedure detailed above) we can observe the channel effect on the signal (see figure 5.5). The channel can be considered approximately flat fading and with AWGN, for this system (even when using the USRP 2 as was done in this transmission). One can also observe the LO in the spectrum of the received signal (at a normalized frequency of 0.6, as defined in the transmission). The sequence is

processed in a parallel function called from the main program. This function was tested with the harness methodology to ensure that all processing components were working correctly.

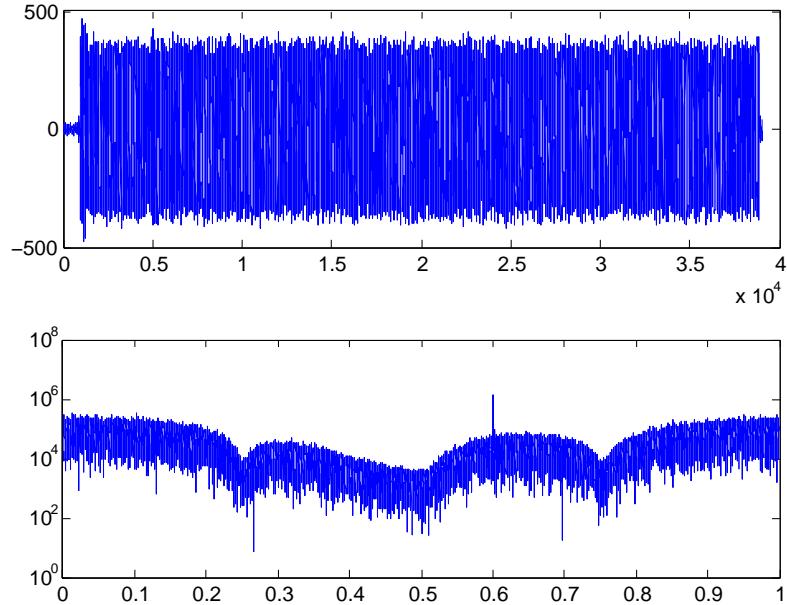


FIGURE 5.5: Received sequence. The top image gives the real part of the sequence and the bottom image gives the spectrum.

5.3.4 Processed Sequence

The signal processing on the received signal is done in several steps, as described in the previous section. In this subsection we will evaluate the results obtained during these processing steps. Figure 5.6 plots the sequences and their corresponding constellations at different stages during the processing of the received signal.

As can be seen, the top left subfigure in 5.6 shows the real part of the received sequence corrected with the frequency shift. We were expecting it to be flatter. It still shows a sinusoidal modulation which means that the frequency shift correction has not been perfectly accomplished. This will affect the following stages.

The middle left figure in 5.6 depicts the real part of the matched filtered sequence. We use rectangular pulse shaping which corresponds to lowpass filtering. Nevertheless, no great difference can be observed, since the received sequence does not contain any significant high frequency components.

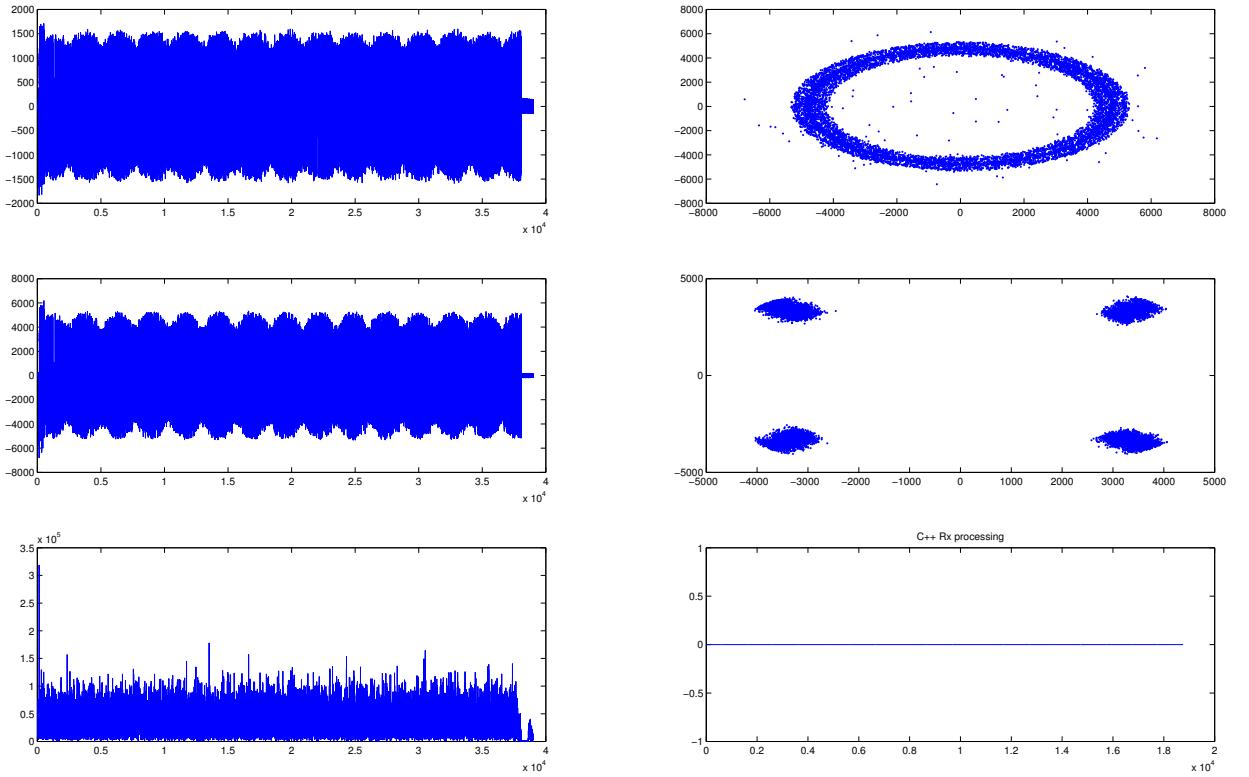


FIGURE 5.6: Results from C++. Top left: real part of received signal. Middle left: real part of the matched filter. Bottom left: crosscorrelation of training symbols and signal. Top right: downsampled signal before phase correction. Middle right: downsampled signal after phase correction. Bottom right: BER (constant zero)

In bottom left figure in 5.6, the crosscorrelation between the known training sequence and the matched filtered sequence can be observed. One can clearly see the peak that has been used to track the beginning of the training sequence and to synchronize the data.

The two first figures on the right show the downsampled constellation before and after the phase correction. The poor frequency correction can be observed, since we have a "doughnut" shape in the constellation. One can also observe the training sequence in the figure, appearing as scattered points in the top right plot. After the phase correction the constellation can be perfectly recovered, as can be seen in the middle right plot.

Finally, the bit error is plotted (for assessment reasons, we already know the transmitted data at the receiver), where one can see the 0 BER.

5.4 System Performance

The test performed in the previous section was done with a high transmission gain and at short distance (approximately 1.5 m). In order to understand the detection and decoding limits of our implementation, we performed some tests with larger distance and different transmission gains. For small gains it was impossible to decode the received signal. The transmission was always detected, but the low SNR made it impossible to track the crosscorrelation peak, and thus correct decoding of the signal could not be done. Figure 5.7 shows the gain threshold for 4.5 m distance with LOS.

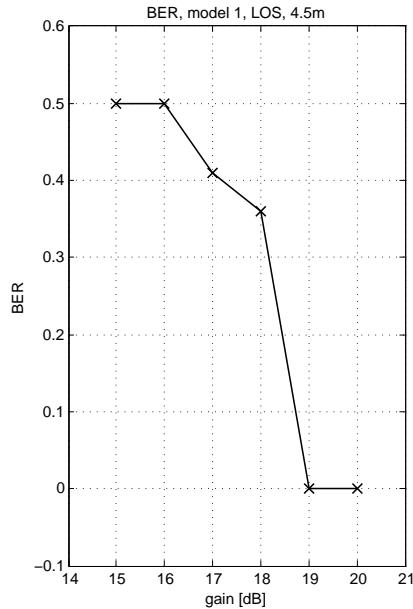


FIGURE 5.7: BER as a function of transmission gain.

Evaluating the power of the noise, we can say, approximately, that for an SNR higher than 6 dB we are able to perfectly decode the received sequence.

5.5 Results and conclusions

As one can see, with this simple system we reach a 0 BER. The processing of the signals is very basic. The receiving thread implementation is more complex and it is crucial to make it work properly. The used rate equals approximately 12.4 Mbit/s . We use 4-QAM and upsample it by 4. We transmit 9280 complex data in a 9490 not upsampled complex sequence:

$$\text{Rate, } R = 25 \text{ Msamples/s} \times \frac{9280}{9490} \times \frac{2}{4} = 12.4 \text{ Mbits/s} \quad (5.1)$$

As an advanced requirement of this project, we needed to implement a system in real-time. If we compute the time required to decode the received sequence in real-time, it gives 8.33 ms ($0.5F/C$), since we transmit a short sequence of 9280 elements. The limiting factor considering time is the computation of the crosscorrelation. Since this is used to track the beginning of the transmission, we can reduce it to the first few points, which ensures catching the training sequence. If we do so, the received sequence can be decoded in 5.56 ms .

5.6 Lessons learned from system 1

The implementation of the first system was important for us to train our skills in C++ and to familiarize ourselves with the hardware as well. Key principles as the Harness Methodology or using Git for developing the C++ code in a team, were fundamental to succeed in the implementation part of this project. Many issues, easily identified thanks to the simplicity of the system, that occurred during the first month helped us later when implementing the second system. Those issues and the ways they were solved, are described in detail in the following sections.

5.6.1 C++ Coding, Git, ITPP and Harness Methodology

At the beginning of the project, Git and ITPP was rather new to every member of the implementation group. Also, the experience in C++ coding was limited. We had some experience in C, Java or C++, but none of us were very skilled. Therefore, the first system was important to gather some experience in C++ coding, using ITPP. This build up resulted in a smaller amount of time going into implementing the second system.

After a few weeks, we had written a bunch of basic functions, i.e., calculating the cross-correlations, the power of signals or filtering signals. Since ITPP is known for slowing down the entire program, we decided to avoid ITPP as much as possible. This sometimes lead to more work having to be done, but it gave us the option of increasing the speed of our program later, by going into the subparts to optimize them.

Also, a crucial point was to define the types of the interface variables between each function. By doing so, we were able to cut down the work in smaller subparts. Each part was dedicated to one person, in charge of implementing and testing it. When all subparts were finalized and tested, they were integrated to form a transmitter and a

receiver. We shared the entire code through Git.

For testing larger functional subparts, the Harness Methodology was often used. This basically runs the C++ code and the Matlab script, saves the results of both and then compares the results. With this method, it could be guaranteed that the Matlab and the C++ code had similar functionality.

5.6.2 Over- and Underflows

During the initial work, hardware issues appeared and we learned how to avoid or solve these problems. At the very beginning, so-called under- and overflows often happened. Overflows typically happened at the receiver side and underflows typically happened on the transmitter side. If the USRP was not getting enough data from the connected computer to fill the output buffer, underflows of the internal buffer happened. This lead to a 'U' being displayed on the screen and the rest of the output buffer being filled with zeros. On the receiver side, it could have happened that the USRP had a full internal buffer and that the connected computer was not yet ready to catch this buffer. A so-called overflow happened and a 'D' was displayed on the screen. Both, over- and underflows can spoil the transmission. Choosing the buffer size wisely helps to avoid under- and overflows.

In order to avoid these problems we launched several separate threads for processing the data. Threads that were detecting transmissions via power and threads that were gathering the data from the USRP, all running in parallel. There was also a maximum number of total samples which could not have been transmitted with the `send()`-function. This maximum length influences the maximum packet size. The maximum number was estimated to around 63000 samples. Having larger packet sizes would result in a crash of the program. Therefore, when larger messages are to be transmitted, the data is cut into several packets which can be transmitted in series.

5.6.3 Transient Problem

In order to find out the transient of the the USRP's, we investigated what we received when we were not transmitting anything. In figure 5.8 (top) the received noise can be seen. There is a slope in this picture. The spectrum in figure 5.8 (bottom) shows that the slope is due to the local oscillator leakage. The local oscillator leakage can be seen

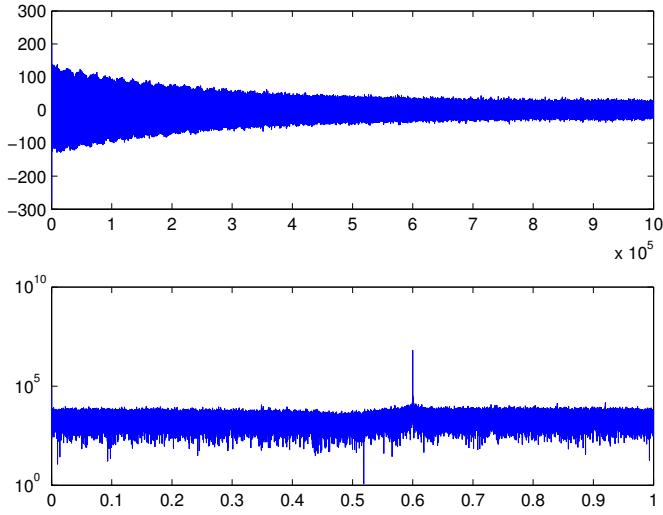


FIGURE 5.8: Measurement and spectrum of the channel without filtering the local oscillator leakage

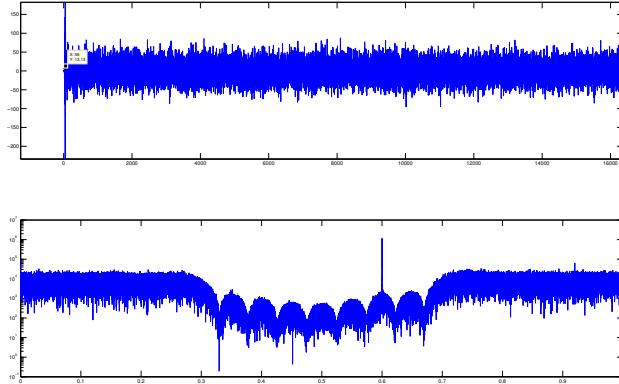


FIGURE 5.9: Measurement and spectrum of the channel without filtering the local oscillator leakage

at 0.6 (in normalized frequency) in the spectrum.

By filtering out the local oscillator leakage, we see the transient at around 58 samples in figure 5.9 (top). The spectrum of the filtered received noise in figure 5.9 (bottom) now gives an idea of the channel.

5.6.4 Root Mean Square (RMS) of the Signal

The RMS of the signal which was transmitted turned out to be important as well. On the one hand, if the RMS is too high, the output amplifier reaches saturation and the

signal is distorted. On the other hand, if the RMS is below a certain threshold, the USRP will only send zeros. By testing empirically, we found an optimal RMS to use. A RMS around 5000 gave superior performance in all respects.

5.6.5 Conjugate Problem

Considering transmitting and receiving, there was also a difference between reading the message created by Matlab from file and creating the message in C++ directly. When the message was created in C++ it had to be conjugated in the transmitter right before sending it and conjugated again in the receiver after receiving it. Without conjugating it twice, it is not possible to decode the data.

Chapter 6

OFDM system

While system 1 was based on a single-carrier transmission scheme, system 2 is an OFDM system where all components have been chosen specifically to reach the final objectives of the project in terms of data rate, processing time and file size. It will also achieve far better results than 12Mbs system 1 that primarily designed in order to fulfill the basic requirements for the project and form the background for system 2.

To boost the data rate as far as possible and knowing that one of the most significant advantages of Extremely High Frequency (EHF) communications is available wide channels, it is important to get full advantage of the available spectrum in order to get maximum data rate. A very common and already conventional solution of increasing the spectral efficiency is using an OFDM system with a high order constellation. In this chapter, we will present the system in general and its Matlab implementation. Matlab functions were used both for pure simulations and for transmissions over real channels using USRPs. Chapter 7 will focus on the C++ implementation of this system.

This chapter is organized in the following way. First the general description of the developed system and its features is provided. Next, channel characteristics for 5 GHz and for 60 GHz are evaluated, together with the challenges that were found during moving from 5 GHz to 60 GHz frequency. Finally, the performance of the system and of particular algorithms is investigated. We analyse the performance of the system in offline mode both with simulations in Matlab and with real data for different parameters, that can be adjusted in the system.

6.1 Transmitter design and specifications

The transmitter part is shown in figure 6.1. An exhaustive description of each block is presented in the following sections. We will start by a word about source coding and channel coding. Some considerations about the used constellations follow. Then, the general sent sequence is presented.

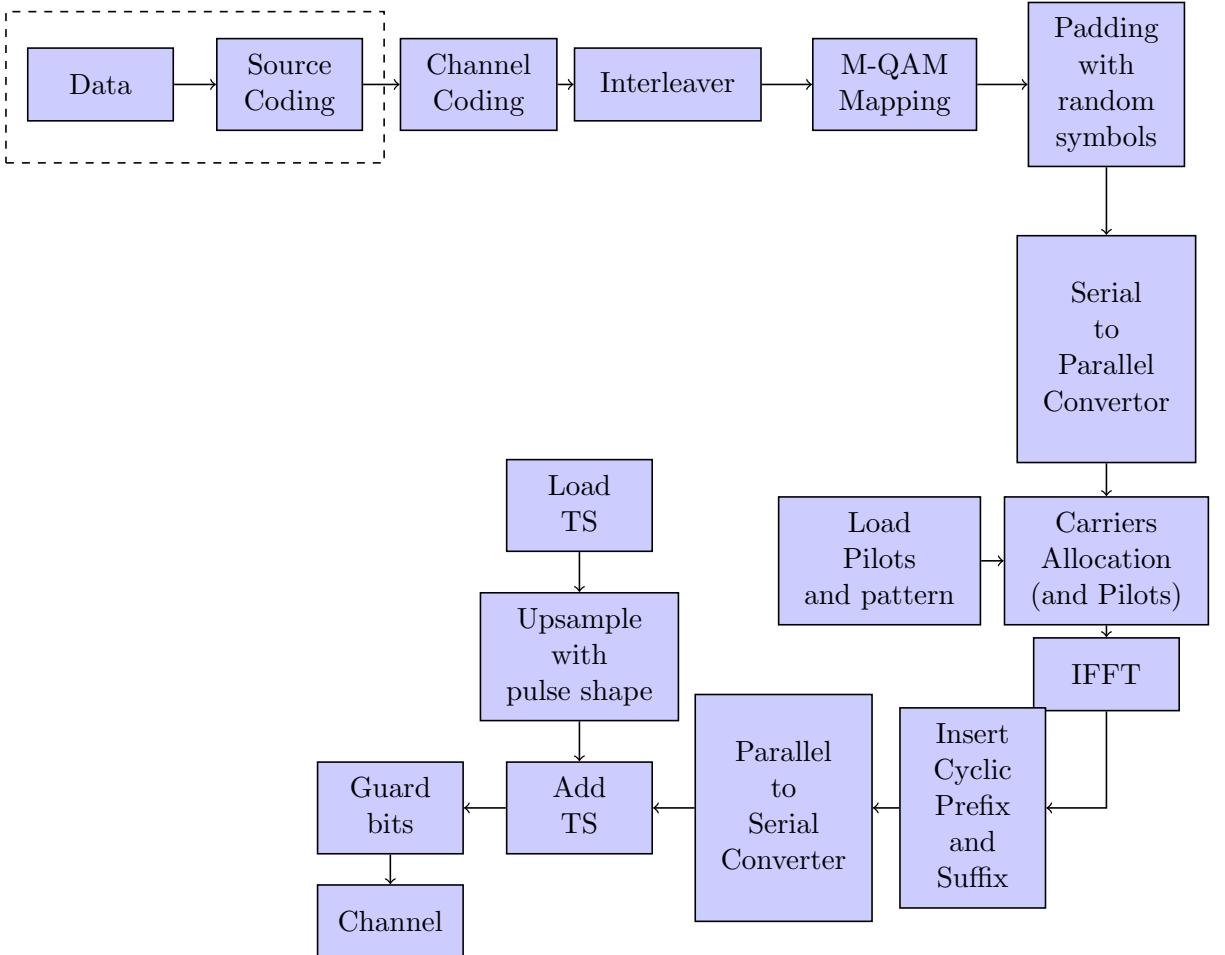


FIGURE 6.1: Transmitter part of the OFDM system

6.1.1 Data and Source Coding

Usually, the binary sequence extracted from a file is badly conditioned and not optimized for the transmission. As an example, images usually have a run of equal bits. This results in high PAPR, problematic in OFDM system. Source coding is then necessary. Additionally, it reduces the size of the file.

The compression algorithm is based on the entropy of transmitted sequence and uses vector Huffman coding. This approach consists of splitting the original sequence into

blocks of m bits (for example of 4) and coding possible block into a sequence of bits with different length. The assignment follow certain rules. The shortest codewords are assigned to the most probable symbols and the longest codewords to the least probable symbols. However the coding depends on data statistics, consequently a decoding key is send at the beginning of the codeword. This coding becomes optimal when the block length m goes to infinity ($m \rightarrow \infty$). In the end, we have a compression rate of:

$$R = \frac{m}{E(S)} \quad (6.1)$$

$$E(S) = - \sum_{x \in A^m} P(x) \log_2(P(x))$$

where $E(S)$ is the entropy of the signal and is expressed in bits. $P(x)$ is the probability of each block of m symbols, and A^m is the set of all possible blocks. As the channel coding algorithm accepts only a certain fixed length codeword (100000 bits), the source coding pads additional random bits such that the total length becomes a multiple of this fixed codeword length and divides the code into several blocks.

The block scheme of the frame is shown in figure 6.2.

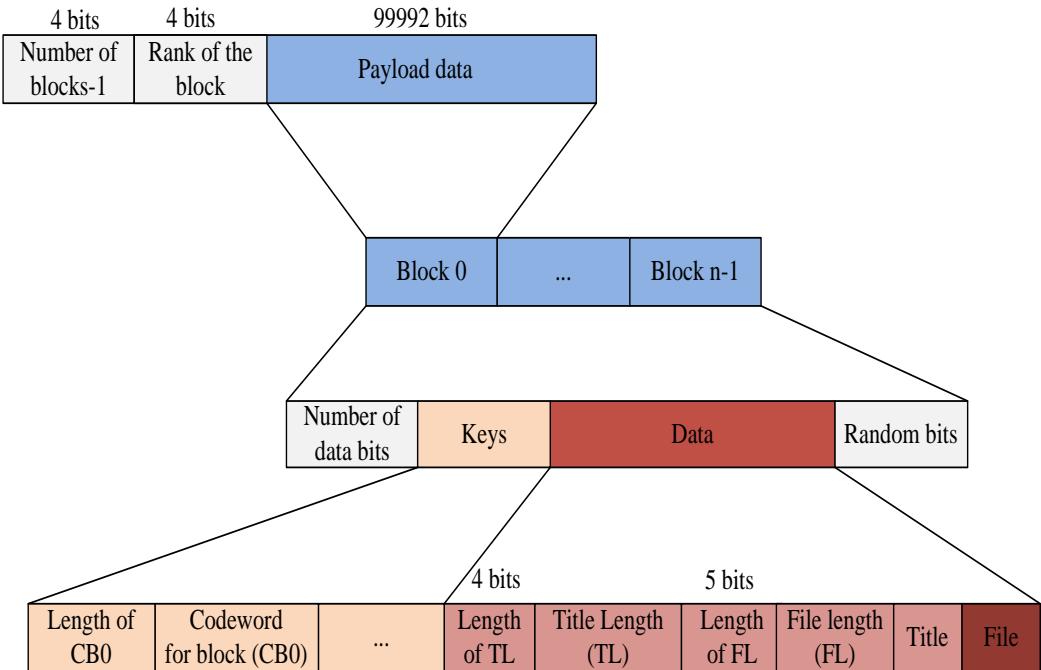


FIGURE 6.2: Source coding block scheme

6.1.2 Channel coding

Once the binary sequence correctly conditioned, one would like to protect himself against error and in the same time transmit at higher rate. Channel coding introduce wisely chosen redundancy to achieve this goal.

The choice of coding algorithm, however, is not obvious since there are different solutions depending on the computational resources and desired bit error rate. Forward Error Correction (FEC) codes are considered particularly interesting for being proficient in coping with both burst and single errors in the channel. Turbo Code seemed a fair option. They offer effective error protection even in the presence of phase noise and for operation in the low SNR regime. The rate of the code was set to $R = 0.8$. A more thorough description follow.

6.1.2.1 Encoder

The block scheme of the turbo encoder is shown in figure 6.3.

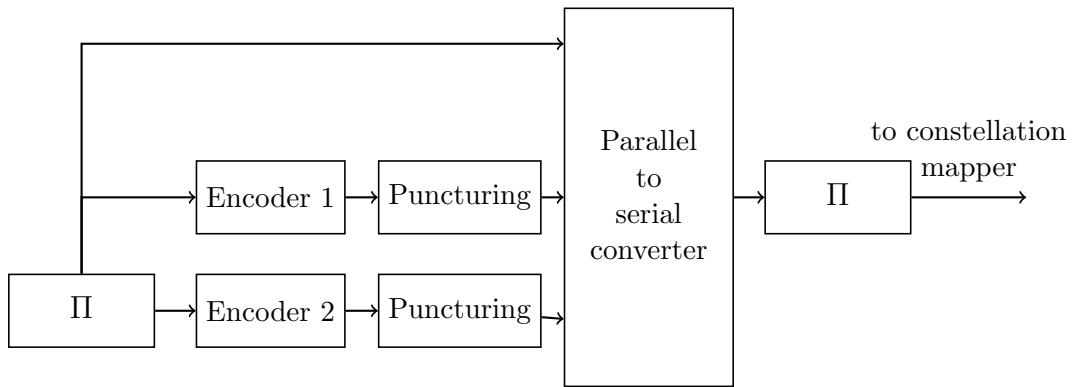


FIGURE 6.3: Block diagram of the encoder

The encoder consists of two convolutional encoders that are concatenated together. They operate in parallel and are identical with the rate $R_{conv} = 1/2$ and 4 memory cells (16 states). The generator polynomial is presented in equation 6.2.

$$G(D) = \frac{1 + D^2 + D^3}{1 + D^3 + D^4} \quad (6.2)$$

The interleavers are used twice: first, before the second encoder and then, the second time, after the parallel-to-serial converter for the whole sequence of bits. This is done in order to minimize the effect of burst errors that are likely to occur in the channel. The

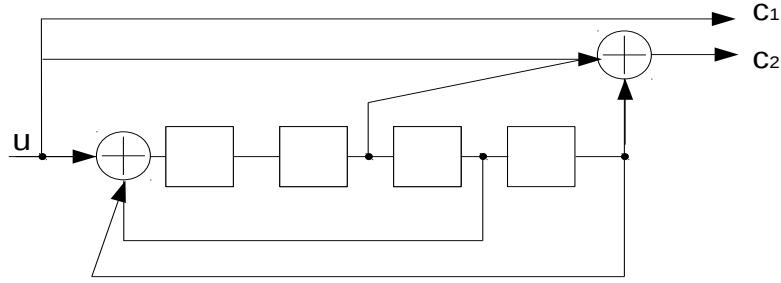


FIGURE 6.4: A rate-1/2 binary convolutional encoder used in the system

interleaving guarantees that the parity bits from the encoder 1 and encoder 2 will differ completely. The length of the first and second interleavers were 100 000 and 125 000 bits respectively. Both interleaving patterns were generated randomly. The codewords are not terminated which means that there is no zero padding in the end of the transmitted data. The output sequence of both encoders is then punctured in order to obtain higher data rate.

The puncturing masks, however, can be easily changed as well as the rate of the code. The puncturing patterns for rate $R = 0.8$ that were used in the simulation are given below.

$$P1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$P2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

6.1.2.2 Decoder

For the decoder, each convolutional decoder is decoded using log-BCJR algorithm (maximum a posteriori decoding) with the approximation.

$$\log(p_1 + p_2) = \max(p_1, p_2) \quad (6.3)$$

Finally, the detection is made by using soft decision. For that, one needs to compute the log-likelihood ratio for each bit as given in 6.4.

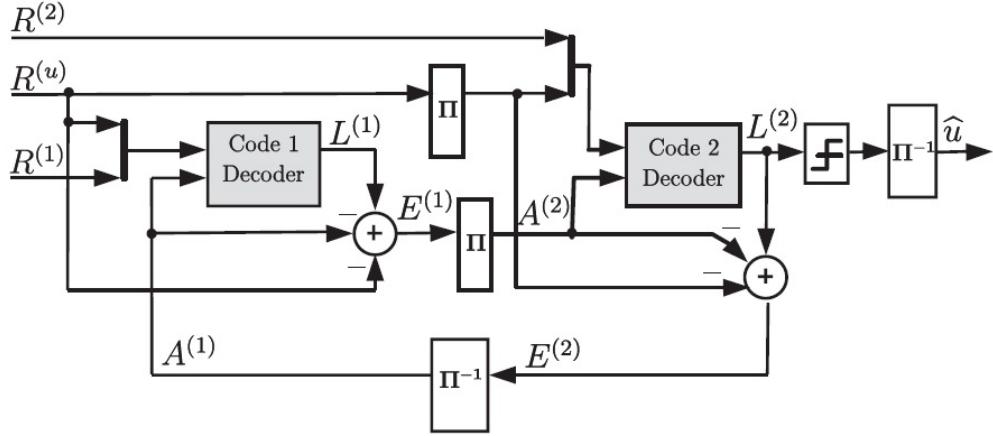


FIGURE 6.5: Turbo code decoder [Johnson, 2009]

$$\begin{aligned}
 LLR(b_i) &= \log \left(\frac{\sum_{S_j|b_i=1} P(R_k|S_j)}{\sum_{S_j|b_i=0} P(R_k|S_j)} \right) \\
 &= \log \left(\frac{\sum_{S_j|b_i=1} e^{\frac{|S_j - R_k|^2 G^2}{2\sigma_n^2}}}{\sum_{S_j|b_i=0} e^{\frac{|S_j - R_k|^2 G^2}{2\sigma_n^2}}} \right)
 \end{aligned} \tag{6.4}$$

where R_k is the received value to which the bit b_i is mapped; $(S_j)_{j \in ([1:2^n])}$ are the different possible mappings; σ_n^2 is the variance of the noise; and G is the gain at the considered frequency and at the considered time. Of course, the computation uses the estimates of the two last values. The first one is computed with the training sequence and the second one is computed with the pilot and the Kalman filter. Here we assume all symbols equiprobable and the noise have a constant power spectrum.

Once the log-likelihood ratio is computed for all the received bits, the objective is to get the log-likelihood ratios for the information bits. To compute them, a simplified version of the $\log - BCJR$ decoding is used. This algorithm consists in the calculation of the logarithm of the probability of doing a certain transition k , at a certain time t_0 : $m(t_0, k)$. Each transition make the system pass from a certain state: $S_{initial}$; to a new state: S_{final} . The probability $m(t_0, k)$ is the probability to be at the corresponding initial state ($\alpha(t_0-, S_{initial})$) knowing only the past of t_0 , times the probability to be at the corresponding final state ($\beta(t_0+, S_{final})$), knowing only the future of t_0 ; times the probability that the bits, corresponding to the transitions are sent (which use the log-likelihood ratios previously computed). This last probability is noted $\gamma(t_0, k)$. In fact, $\alpha(t_0-, S_{initial})$ can be seen has the probability of doing the considered transition knowing the past; $\beta(t_0+, S_{initial})$ can be seen as the probability of doing the considered

transition knowing the future, and $\gamma(t_0, k)$ is the one only considering the present. All these probabilities are independent and we can estimate $m(t_0, k)$:

$$\begin{aligned} m(t_0, k) &= P(k_{t_0} | t < t_0) \times P(k_{t_0} | t = t_0) \times P(k_{t_0} | t > t_0) \\ &= \alpha(t_{0-}, S_{initial}) \times \gamma(t_0, k) \times \beta(t_{0+}, S_{final}) \end{aligned} \quad (6.5)$$

However, $\alpha(t_{0-}, S_{initial})$ and $\beta(t_{0+}, S_{final})$ should be computed. These probabilities can be easily computed using the recursions:

$$\begin{aligned} \alpha(t_{n+1-}, S_o) &= \sum_{S_p} \gamma(t_n, S_p - > S_o) \times \alpha(t_{n-}, S_p) \\ \beta(t_{n-1+}, S_o) &= \sum_{S_p} \gamma(t_n, S_o - > S_p) \times \beta(t_{n+}, S_p) \end{aligned} \quad (6.6)$$

In both convolutional codes of used turbo code the codeword is not terminated and the initializations are:

$$\begin{aligned} \alpha(t_{0-}) &= \begin{cases} 1, & \text{if } S_o = S_i \\ 0, & \text{otherwise} \end{cases} \\ \beta(t_{N+}) &= \frac{1}{\text{number of states}} \end{aligned} \quad (6.7)$$

where N is the last transition. Then it is possible to compute the probability $m(t_0, k)$.

Finally the log-likelihood ratios of the input bits are easily calculated:

$$LLR(t_0) = \log \left(\sum_{k/b_{input}(t_0)=1} m(t_0, k) \right) - \log \left(\sum_{k/b_{input}(t_0)=0} m(t_0, k) \right) \quad (6.8)$$

As we only work with the log values of the probabilities, all the products become summations. In order to avoid heavy computation, the summations in the log domain are approximated to the search of the maximum:

$$LLR(t_0) = \underset{k/b_{input}(t_0)=1}{Max} (\log(m(t_0, k))) - \underset{k/b_{input}(t_0)=0}{Max} (\log(m(t_0, k))) \quad (6.9)$$

The log likelihood ratio is computed using pilot signals. For punctured bits, the log likelihood ratio is set to zero. The number of iterations for the decoding is set to 20.

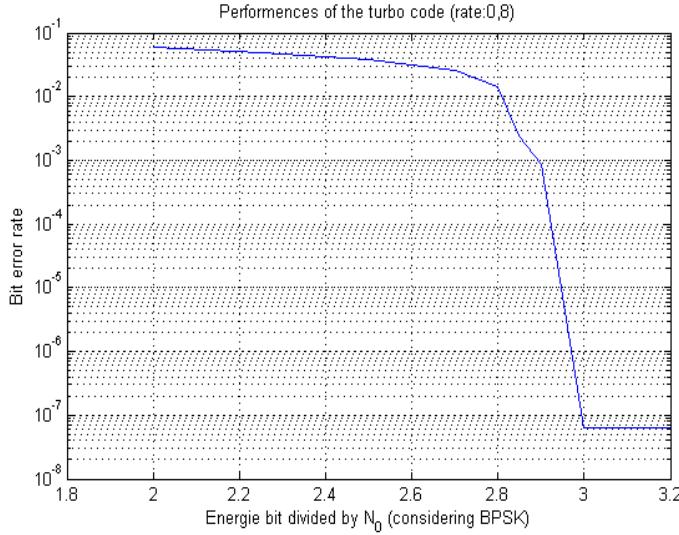


FIGURE 6.6: The performance of the turbo code used in the simulation

However, in order to reduce the decoding time, the process is forced to stop if the codeword is the same after three consecutive decoding procedures and is instead considered to be decoded successfully.

One can adjust the log likelihood depending on the desired performance, channel characteristics, and OFDM parameters, such as number of carriers and number of points in constellation. Figure 6.6 shows the BER rate in relation to signal to noise ratio (SNR). For low SNR the implementation of the FEC coding significantly improved the performance of the system, it made it much more robust to negative channel effects and allowed an increased order of constellation mapping for each of the subchannels.

In 5 GHz channel, one can use 128 subcarriers and QPSK modulation scheme without coding and get zero BER. We will use channel coding in order to boost the rate using higher order constellations, such as 32 or 64 QAM, where the number of errors is rather high if coding is not used. However, Turbo code results in higher delay in the decoding and greater dependence on channel parameters.

6.1.3 Constellation mapping

Increasing the constellation order increases the rate. Our aim was to find an optimal constellation in terms of effective data rate, bit error rate and decoding time. The limiting factor in our system is the phase noise. When using high-order constellations (Number of bits/symbol > 6), even with perfect synchronization, frequency and phase offset estimation, decoding was impossible as the number of errors was 50% even with

FEC. Classical 64-QAM with Gray mapping did not work as expected, resulting in high error rate, therefore, a more advanced mapping was chosen. The so-called 64-BEES is described below.

This is an energy efficient constellation, shown in Figure 6.7, which allowed us to reach up to 72 Mbps thanks to a reduction in the PAPR. We built a system with this mapping, which gave us good simulation results. However, with the 5 GHz boards, the Turbo decoder did not converge for several test sets. Thus, in order to guarantee stability and robustness of the transmission, we decreased the constellation order down to 16-QAM.

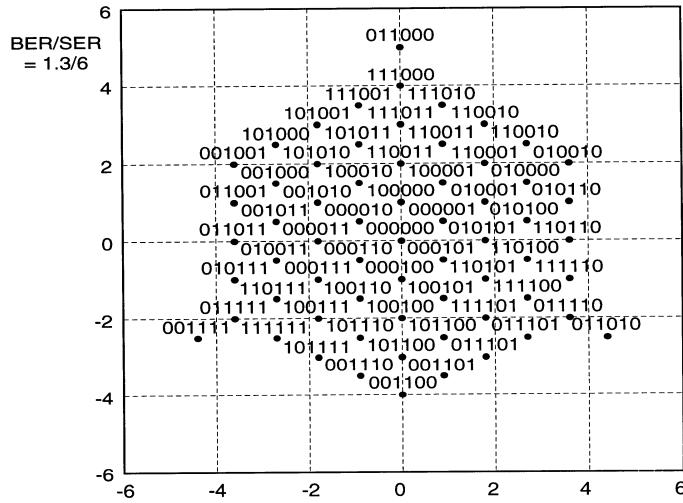


FIGURE 6.7: 64-BEES, an energy efficient mapping.

6.1.4 Padding with random symbols

After applying the modulation scheme, one may not get complete OFDM symbols, i.e. not being able to fill each subcarrier with data in the last OFDM symbol. Then, we should pad our signal in order to fill every subcarrier with data. For that there are two choices:

- Zero-padding
- Padding with random symbols

The first choice will lead to poor peak to average ratio, whereas the second choice will maintain the same PAPR. Obviously, we will pad the signal with random symbols.

6.1.5 Carrier and pilot allocation

Estimation of the channel at the receiver is primordial in order to reduce interference. At this effect, we add pilots. They are complex sequences with random phase and constant modulus that is equal to one. Once generated, we store them in memory in both transmitter and receiver. They will be inserted on wisely chosen and fixed frequency positions, also known from the receiver.

In our system we follow the pattern stated in figure 6.8. Note, that the numbers in the figure are taken from the MATLAB code. The normalized frequency $\nu = 0$ corresponds to the index 1.

Note that subcarriers situated between bins 54 and 79 are saved free, since our channel has high attenuation at those subbands. These bins correspond to high-frequencies and thus, the poor frequency response mainly come from the ADC low pass filter.

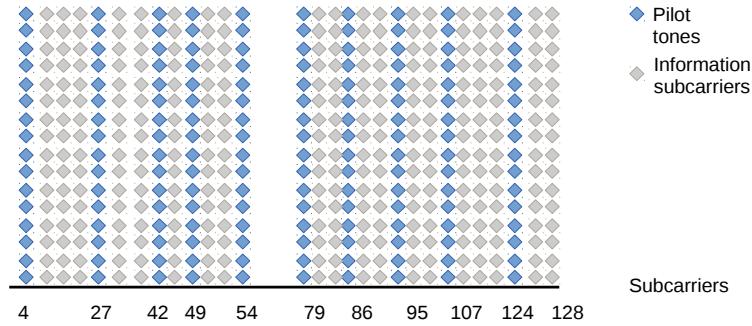


FIGURE 6.8: Pilot and information subcarriers

After insertion of the pilots, the information is ready for going to time domain (IFFT) which has a size equal to the total number of subcarriers. The reciprocal operation (FFT) will be performed at the receiver. This means we assume cyclic convolution with the channel in time domain. Cyclic prefix and suffix are then necessary.

6.1.6 Cyclic prefix and suffix insertion

In order to cope with Inter Symbol Interference (ISI) coming from both future and past samples, one need to keep a guard samples in time domain between OFDM symbols. The length of these guard samples should have at least the same length minus 1, as the equivalent channel impulse response for past taps. The ISI can also introduce interference between carriers, since a delayed version of complex exponentials of different

frequencies are not anymore orthogonal.

That can be solved by adding to our signal the so-called cyclic prefix and suffix at the beginning and at the end of our OFDM symbol. With L , that equals to the number of causal taps on the channel and M defined as the number of non causal taps, this basically consists in taking the last $L-1$ samples of our symbol as the prefix and the first M samples of our symbol as the suffix.

In 5 GHz environment, we use $L - 1 = 18$ and $M = 2$ and at 60 GHz $L - 1 = 10$ and $M - 1 = 4$. The cyclic prefix is also used for estimation of the channel parameters.

6.1.7 Training Sequence

To allow the receiver to get the right sampling time, we use a time domain training sequence. For our model we used a Golay complex training sequence (TS), generated from two complementary sequences as described in [Parker et al., 2002]. It has good autocorrelation properties and suits good for the synchronization algorithm. The auto-correlation of the training sequence is shown in Figure 6.9 used both for 5 GHz and 60 GHz transmission. In order to raise the reliability, the TS was upsampled with a rectangular pulse,, necessary because we cannot transmit the TS in all the channel, otherwise it results in distortions. The upsampling rate equals to $Q = 4$ for 5 GHz.

Training symbols are used primarily for synchronization but also for obtaining coarse estimates of the channel's phase offset and channel gain.

As the training sequence reduce the effective data rate, we strived to reduce the number of transmitted symbols as much as possible. A compromise needs to be made between having enough training sequence bits, allowing to decode robustly and having as few as possible in order to get the highest possible effective data rate and avoid problems due to the frequency offset.

Lastly, guard bits were added in front of each block to get the OFDM frame ready for the transmission. The structure of the OFDM frame is shown in figure 6.10.

The power spectral density of the OFDM signal for $N = 128$ subcarriers is shown in figure 6.11.

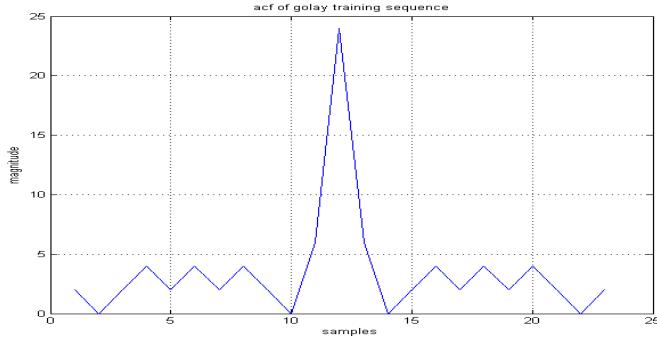


FIGURE 6.9: ACF of Golay training sequence

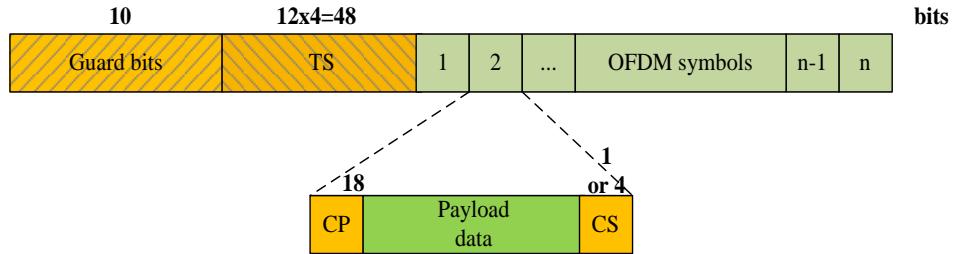


FIGURE 6.10: The structure of the OFDM frame

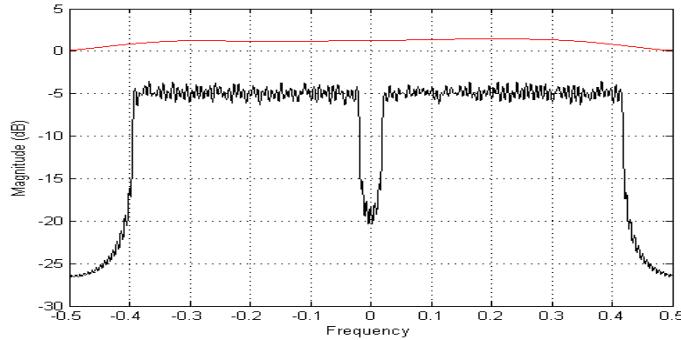


FIGURE 6.11: Spectrum of the transmitted OFDM signal

6.1.8 Packet detection

Instead of using time domain synchronization, one could use frequency domain method. For this packet detection within this system Schimdl-Cox method was proposed. The method adds two systematically designed OFDM symbols in front of each packet. For packet detection, only the first symbol is needed.

The second half of the first symbol is an exact copy of the first half, when they are observed in the time domain. The way this is achieved is by generating a pseudo noise, for instance using an M-sequence, modulating the noise with a symmetric 4 point constellation of choice, for instance QPSK or 4 points from a 64-QAM, and then assigning

one symbol to each even number subcarrier in the first OFDM symbol, X_1 , for example as in 6.10.

$$X_1 = [7 + 7j, 0, -7 + 7j, 0, 7 - 7j, 0, -7 + 7j, \dots] \quad (6.10)$$

Once the IFFT is performed on the signal, the first OFDM symbol will have the intended, repetitive pattern. This will be unique for the transmission, since all OFDM symbols containing actual data will have information on all/most subcarriers. The problem of identifying an incoming transmission is reduced to that of finding a symbol with this repetitive pattern, which only requires knowledge of the predefined number of channels for the system, and a specified bandwidth to scan. Identification of the repetitive pattern is done in a sliding window fashion:

$$P(d) = \sum_{m=0}^{L-1} r^*(d+m)r(d+m+L) \quad (6.11)$$

where L is half the number of subcarriers and r is the received signal in the time domain. This can be done iteratively, adding the next sample of the window while removing the last one, for each sample:

$$P(d+1) = P(d) + \text{conj}(r(d+L)) * r(d+2L) - \text{conj}(r(d)) * r(d+L) \quad (6.12)$$

Another metric is defined, consisting of the energy in the latter half of the observed window:

$$R(d) = \sum_{m=0}^{L-1} |r(d+m+L)|^2 \quad (6.13)$$

which can be iterated in the same way as the P. Finally, a detection metric, M, is computed by normalizing the pattern energy by the squared energy of the latter half of the window:

$$M = \frac{|P|^2}{R^2} \quad (6.14)$$

The result can be observed in figure 6.12 where the simulation was done with a system using 64 subcarriers and a cyclic prefix length of 5, adding only white noise to the

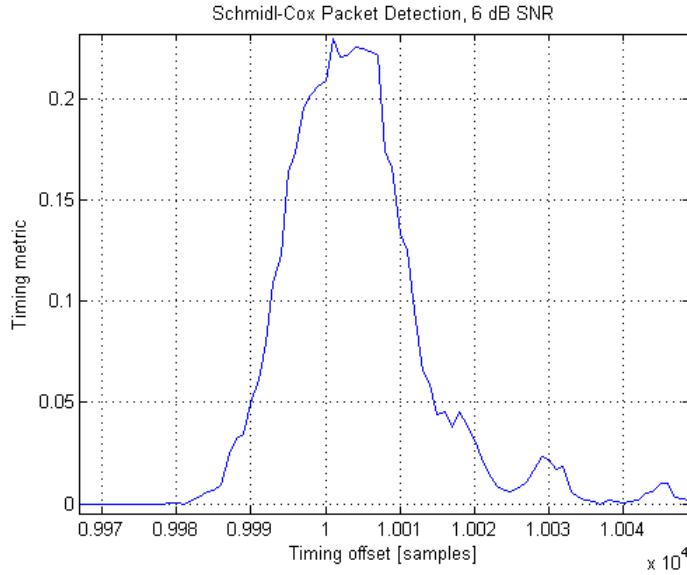


FIGURE 6.12: Packet detection using the Schmidl-Cox algorithm, with an SNR of 6 dB

received signal. The signal was preceded with a series of 10,000 zeros, to simulate the silence, or absence of data, before the beginning of a transmission. The received signal had an SNR of 6 and as we can see in the figure, the metric reaches a much higher magnitude than the surrounding elements. In 6.13 we see the detection metric for a similar situation, but with a lower SNR of 0 dB and also with a simulated LO offset of 0.01. As can be seen in the figure, the metric is still much more powerful at the start of the packet than in the surroundings.

The plateau is of the same length as the cyclic prefix. In a high-SNR-environment, it is easy to identify the first sample of the signal, but as the SNR is reduced one may have to resort to cyclic prefix-correlating methods in order to determine what sample is truly the first actual sample and which ones that ought to be discarded, on behalf of being cyclic prefix samples.

Once a packet has been detected and the first sample has been identified, the angle of $P(d_{optimal})$ can be used to get a good frequency offset estimation. Provided that the angle is known to be contained in the interval $[-\pi, \pi]$ at all times, the frequency offset can be computed as $\hat{f}_{off} = \phi/(\pi T)$, where ϕ is the angle of $P(d_{optimal})$ and T is the number of carriers in the OFDM system.

This method, simulated and tested in MATLAB simulation was, however, not implemented in C++ because of time constraints of the project.

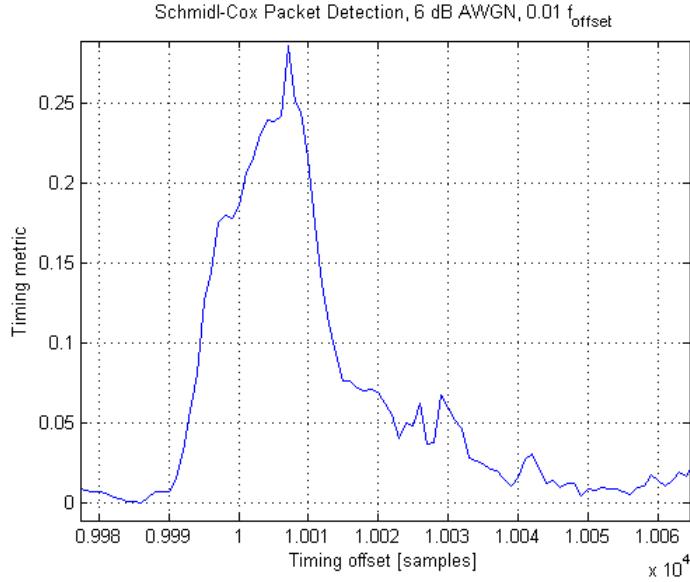


FIGURE 6.13: Packet detection using the Schmidl-Cox algorithm, with an SNR of 0 dB and a LO offset of 0.01

6.2 Receiver Part

The received data is distorted by several channel effects that include:

- noise, that is modeled as AWGN
- loss of the synchronization
- frequency offset
- phase noise

To recover the original data from received distorted signal, the following parameters should be estimated before demodulating the data:

- sampling time t_{samp}
- channel impulse response \hat{h}
- frequency offset $\Delta\hat{\phi}$
- phase noise $\Delta\hat{\theta}$

In this system we do not estimate $\Delta\hat{\phi}$ and $\Delta\hat{\theta}$, just the common phase error.

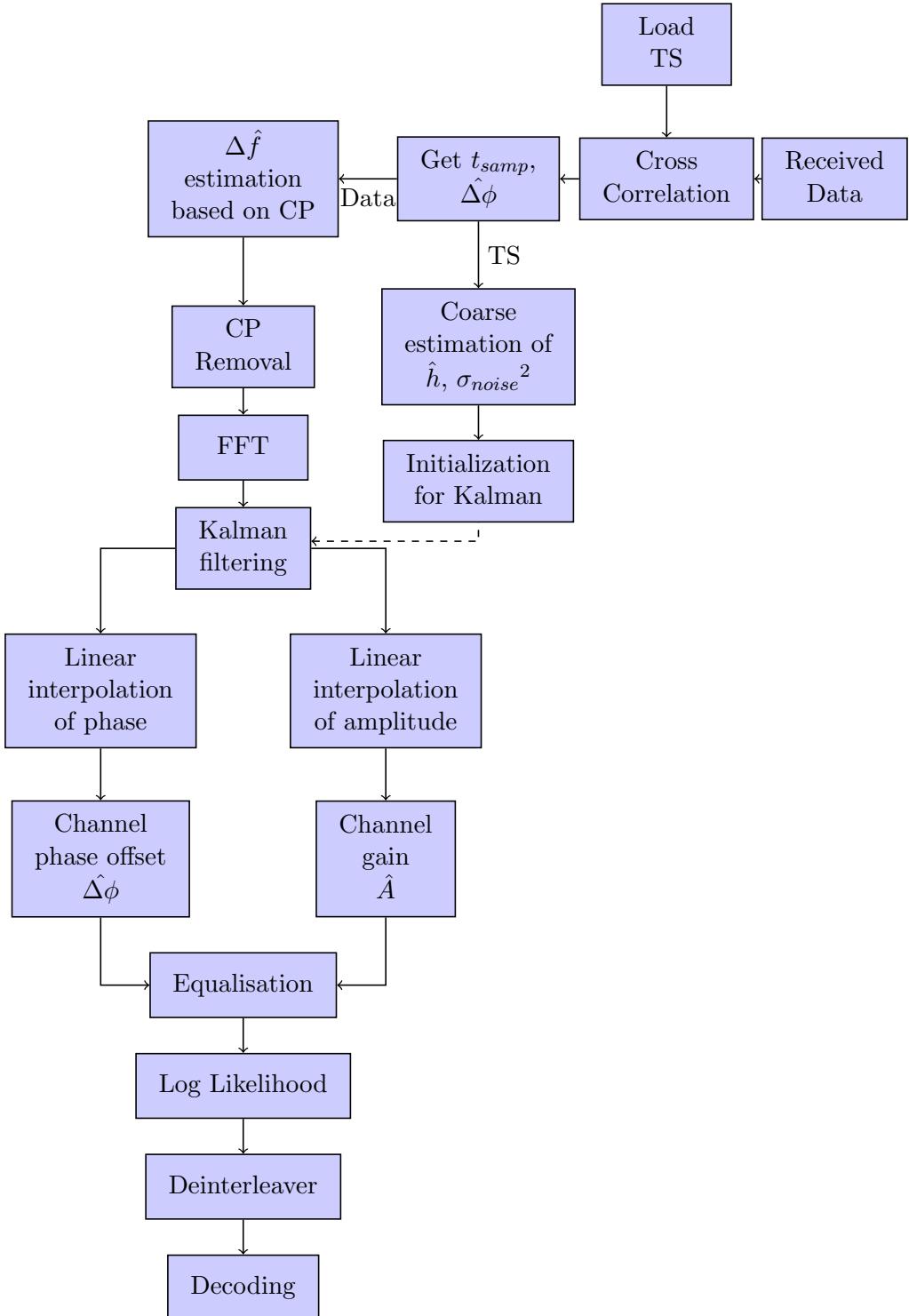


FIGURE 6.14: Block scheme of the receiver end for OFDM system

Finding an optimal algorithm to estimate all these parameters is a challenging task, first, because of the number of parameters that have to be estimated and, second, because some of them are changing and depend on frequency and time, whereas the others can be assumed to be constant for the whole transmission time. Our solution is presented in

the following sections. The block scheme of the receiver is shown in figure 6.14 where we estimate.

6.2.1 Sampling time for the synchronization

The first problem that arises at the receiver, once the data is received, is the problem of time-synchronization. As soon as the system is synchronized, it is able to further process the data and correct the other effects. In order to determine the correct sampling time instant, we use a traditional method based on the training sequence sent at the beginning of each frame. At the receiver, we computed the crosscorrelation of the received data with a known training sequence, upsampled by factor 4. The peak value of the crosscorrelation corresponds to the correct sampling point as shown in 6.15.

$$t_{\text{samp}} = \arg \max_{t_s \in [\text{data}]} \sum_{n=0}^{L-1} r(n + t_s) \cdot t(n) \quad (6.15)$$

The training sequence, however, can also be used for estimation of the phase offset and the channel impulse response. For that reason, we sent a complex training sequence. We used it to estimate the channel impulse response and noise variance. These estimates becomes quite coarse, however, they give an idea of the channel and can be used later as initialization of the parameters of the Kalman filter used to make more accurate parameter estimation.

To estimate the channel impulse response \hat{h} , as a FIR Wiener filter

$$\hat{h} = \Sigma_{tt'} \Sigma_{tt}^{-1} \quad (6.16)$$

where $\Sigma_{tt'}$ is the Toeplitz matrix with the autocorrelation function of the training sequence t for different time lags.

$$\Sigma_{tt} = \begin{bmatrix} R_{tt}(0) & \cdots & R_{tt}(L+M-1) \\ \vdots & \ddots & \vdots \\ R_{tt}(L+M-1) & \cdots & R_{tt}(0) \end{bmatrix} \quad (6.17)$$

where M is the number of channel filter taps.

$$\Sigma_{tt'} = [R_{tt'}(-M) \cdots R_{tt'}(L)] \quad (6.18)$$

Using the obtained channel filter impulse response, we can compute the estimate of the received training sequence \hat{t}_{rx} . Taking the difference between the estimated and the real training sequences, we can estimate the variance of the noise.

Important to notice that described above method should be used when the frequency offset equals 0. If the frequency offset is neglectable comparing to the length of the training sequence (the phase difference between the first and the last sample of the training sequence is small) it is still possible to do it. This is not a big problem for transmission at 5 GHz. However, for communication over 60GHz, this becomes an important issue. One way of combat it is to reduce the length of the training sequence as much as possible. After upsampling, there are only 24 samples to avoid problems with frequency offset, but it is still a large number of samples for this frequency:

$$\begin{aligned} \nu &\approx 0.01 \\ \Delta\phi &= \nu \times n_{samples} 2\pi rad \\ &\approx 0.48\pi rad \end{aligned} \tag{6.19}$$

Consequently, as follows from the Equation 6.19, we are close to the critical length.

Once the starting point of the data was known, it was separated from the training sequence. At the next step we used the cyclic prefix symbols to estimate the frequency offset $\Delta\hat{f}$ according to the method described by [Sandell and Edfors, 1996]. However, this method gives the frequency offset modulo $(1/N)$, with N the number of carriers. Consequently, ν_{offset} should be lower than $(1/2N)$. If it is not true, the data will be shifted in the frequency domain. After the frequency offset was obtained, the cyclic prefix and suffix were removed from the data and the data was converted to the frequency domain for further processing.

To get a detailed estimation of the channel parameters, several approaches are available. The choice depends, to a large extent, on the available computational resource, complexity, and the amount of additional information that has to be transmitted as it reduces the effective data rate.

Pilot tones embedded in the data sequence provide sufficient information for obtaining reliable estimation. As we had a relatively wide channel bandwidth, we used 10 pilots tones for 5 GHz and 5 pilots for 60 GHz in order to get sufficient information about

all frequencies. In our model we used a state space model and implemented a Kalman filter to filter the channel measurements. The detailed description of the fine parameter estimation and signal equalization is given below.

6.2.2 Filtering of the pilot's channel information

We now want to estimate the channel frequency response at each OFDM symbol, equalizing both the frequency offset and the common phase error. However, the measurements on the pilots are noisy. Filtering is then necessary. We filter the phase and gain independently thanks to two Kalman filters presented hereby. This method also calculates the frequency offset at the OFDM symbol level.

Phase filtering In order to track the changes in the phase (which encloses phase variations and frequency variations) and to track the amplitude of the pilot subcarriers where we allocate pilot signals, we used a Kalman filter. The noisy observations used:

$$\text{ChannelPilot} = \frac{Rx_{\text{pilot signals}}}{Tx_{\text{pilot signals}}}$$

The result will be a matrix of (number of pilot frequencies, number of OFDM symbols)

This is iterated to get a new estimation of each OFDM symbol (n stands for the number of the current OFDM symbol).

$$\begin{pmatrix} \phi_{n+1} \\ f_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 2\pi \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \phi_n \\ f_n \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} w(n)$$

$$\phi_{obs_n} = \phi_n + v(n) \quad (6.20)$$

and the output is:

- $\hat{\phi}$: phase estimate in function of the time for each pilot inside an OFDM symbol.

The the initialization values for the phase are:

$$\begin{pmatrix} \hat{\phi}_0 \\ f_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{aligned}
 Q(0) = Q_N &= \begin{pmatrix} 10 & 0\pi \\ 0 & 1 \end{pmatrix} \\
 R_w &= \begin{pmatrix} 0.1 & 0\pi \\ 0 & 0.000001 \end{pmatrix} \\
 R_v &= 0.1
 \end{aligned} \tag{6.21}$$

$$\phi_{obs_n} = \text{angle}(\text{ChannelPilot}_n)$$

Gain filtering To track the gain, a random walk model is used.

$$A_{obs_n} = |\text{ChannelPilot}_n|$$

$$A_{n+1} = A_n + w(n)$$

$$A_{obs_n} = A_n + v(n) \tag{6.22}$$

The output is:

- A : The gain for each pilot subcarrier and for each OFDM symbol

The initial values for the gain tracking are:

$$\begin{aligned}
 \hat{A}_0 &= Gain_0 = DFT(\hat{h}) \\
 Q(0) &= Q_N \\
 R_w &= 0.1 \\
 R_v &= 0.1
 \end{aligned}$$

6.2.3 Channel interpolation

Once the pilot filtered, one can do linear interpolation between them, independently for the phase and the gain. The used method is presented hereby.

For the high and low frequencies outside of the range of the pilots we used constant interpolation (hold of the value of the last pilot). Between the frequencies, that were allocated to the pilots, we used linear interpolation. For the constant interpolation, gain and phase are equal for each subcarrier within these areas. This technique gave very bad results, and therefore, the pilots are chosen in order to have pilot on the side of the channel.

6.2.3.1 Linear interpolation

In the center of the channel, both boundaries are pilot carriers. The used formulas follow. Note the care taken with the phase to chose the shortest path.

Phase subindex i stands for the pilot index (Recall that $i=0$ is the pilot on the 4th area boundary). Subindex n stands for the data subcarrier index.

$$\phi_n = \frac{f_n - f_{p_i}}{f_{p_{i+1}} - f_{p_i}} \cdot (\phi_{p_{i+1}} - \phi_{p_i}) + \phi_{p_i}$$

where $(\phi_{p_{i+1}} - \phi_{p_i})$ is calculated in order to be in the range $[-\pi; \pi]$

Gain

$$A_n = \frac{f_{p_{i+1}} - f_n}{f_{p_{i+1}} - f_{p_i}} \cdot A_i + \frac{f_n - f_{p_i}}{f_{p_{i+1}} - f_{p_i}} \cdot A_{i+1}$$

It is very important that the frequencies should be in the range of $] -1/2; 1/2[$ and not in range $[0; 1[$.

6.3 Channel description

Channel estimation is crucial for successful system design. It is the primary goal before the system can be designed. Some tests for channel characterisation were made prior the OFDM system was implemented, but more precise results were obtained after the first simulation was done in MATLAB. Therefore, this section is given after the OFDM system has been introduced. It discusses the channel distortions that affect transmitted data. The channel introduces the following distortions:

- **AWGN noise**

The AWGN noise is modelled in the simulation with different values of the noise

variance and is used as one of the parameters for investigation of the system performance.

- **Frequency offset**

The frequency offset is the tiny difference in frequency between transmitter and receiver oscillators. This offset is low in the 5 GHz channel whereas is much higher when one goes up to 60 GHz. Due to oscillators unestabilities, it is expected to be time varying. However, these variations are small compared to the mean value of the offset.

- **LO leakage**

Appears to be significant in 5 GHz equipment and should be taken into account and transmission close to DC should be avoided.

- **Impulse response**

The impulse response of the channel describes how many signal components will arrive at the receiver. For OFDM implementation the number of taps is important for choosing the correct value for cyclic prefix.

- **Channel gain in function of frequency**

Knowledge of the frequency response of the channel allows to choose effectively the carriers of the OFDM sub bands that will be used for channel estimation and information transmission. The goal is to use as much of the available bandwidth, but channel variations in frequency domain should be estimated and corrected at the receiver. In case of fast fading channel, this becomes a difficult task.

- **Phase noise**

An important aspect that should be carefully considered for EHF communications is phase noise. Phase noise arises due to the local oscillator imperfections at the receiver and transmitter. It can be seen as phase modulation of the local oscillator signal that supposes to have a constant amplitude and phase [Armada, 2001]. In frequency domain local oscillator has some additional frequency components around the nominal frequency.

For OFDM systems phase noise can become the main limiting factor. Once the orthogonality between the subcarriers is destroyed, the transmitted data is corrupted. For classical OFDM and in present project, the symbol shape has a rectangular form. In frequency domain it becomes a sinc function for each sub band with the side lobes attenuated by 13 dB. The phase noise of the local oscillator can be seen

as interference from another subcarriers.

Since in this project the primary attention is paid to OFDM implementation, it is crucial for the system to be able to predict the expected influence of the phase noise and be able to correct it. The following section gives a brief theoretical description of phase noise effect and investigates its impact on the system implemented in this project.

The amount of phase noise can be reduced by improving the quality of the local oscillator, however that increases the cost of the receiver. Usually, the phase noise is described by its measured total side band power in the bandwidth of 1Hz at a given frequency offset from the carrier and with respect to carrier power. This parameter is given in the specification. For USRPs used in present project the total side band power at 1 MHz offset equals to -86 dBc/Hz [3]. We have been also provided with the data measured for provided USRPs.

6.3.1 Dealing with phase noise

Figure 6.15 shows the phase noise power in relation to frequency offset.

This measurement has been done in baseband and includes the effect of both the transmitter's and receiver's phase noise.

For an OFDM symbol $b[k] = \sum_{n=0}^{N-1} B[n]e^{j2\pi nk/N}$, where N is the number of carriers and $B[n]$ is the transmitted symbol at some subcarrier frequency, the phase noise results in two separable effects. First, there appear common phase noise. Common phase noise is a phase offset that affects all frequencies in the same amount. For OFDM system this means that there will be a rotation of the subcarriers by a certain fixed common angle for each OFDM symbol. In other words, it is equivalent to a multiplication by a complex number $e^{j\theta}$ which will rotate the constellation by:

$$r[k] = b[k]e^{j\theta[k]}$$

With the assumptions made in [Armada, 2001] that $e^{j\theta[k]} \approx 1 + j\theta[k]$ (i.e. small phase noise amplitude), the signal can be written as

$$y[k] = B[n] + \frac{j}{N} \sum_{m=0}^{N-1} B[m] \sum_{k=0}^{N-1} e^{j2\pi(m-n)k/N} \quad (6.23)$$

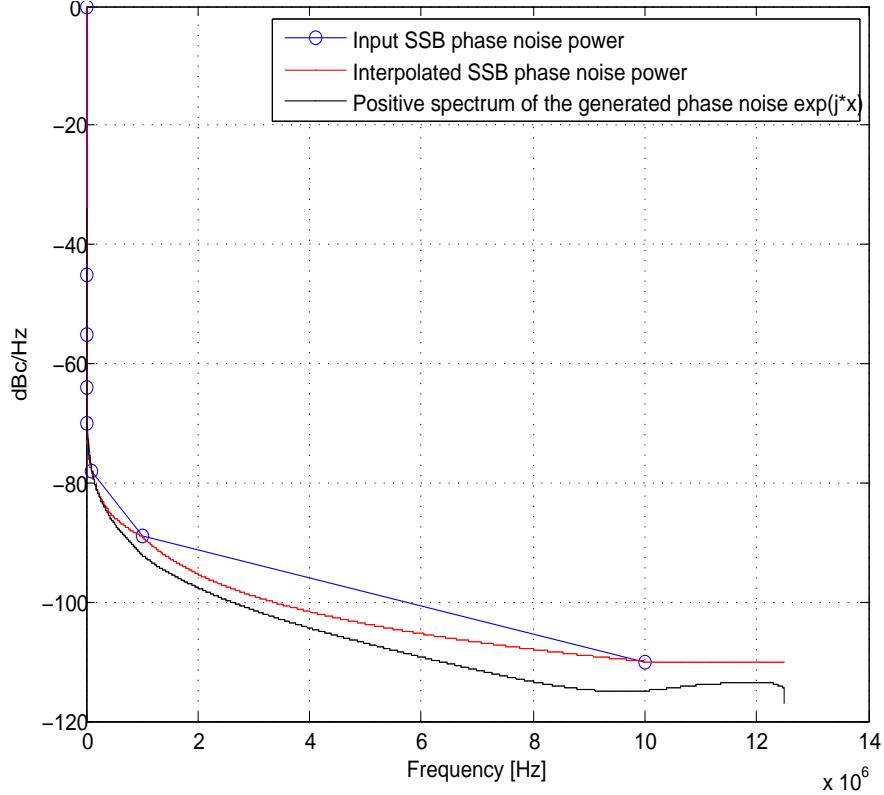


FIGURE 6.15: Phase noise measured for this project

where the case, when $m = n$ results with the common phase error:

$$\Theta = \frac{1}{N} \sum_{k=0}^{N-1} \theta[k] \quad (6.24)$$

This error can be corrected by several methods. In case of OFDM system with pilot tones in the frequency domain, common phase error can be estimated for pilot frequency and is assumed constant for all frequency subbands. Common phase error, although can be assumed to be constant or slowly varying, can have a quite large variance [Abhayawardhana and Wassell, 2002].

Another negative impact is Inter Carrier Interference (ICI) for OFDM and it is much harder to correct. It corresponds to the case when $m \neq n$. The phase noise destroys the orthogonality between subcarriers channels which create ICI. For the variance of the phase noise comparable to the spacing between subcarriers Δf it can severely distort the signal and increase the bit error rate. The degradation of the signal to noise ratio with assumption of small phase noise variance $\sigma_{phi}^2 \ll 1$

[Moeneclaey, 1997] is given by:

$$D_{SNR} = 10 \log\left(1 + \frac{E_s}{N_0} \sigma_{phi}^2\right) \quad (6.25)$$

Given at first the hardware equipment working at 5 GHz, we expected the influence of the phase noise to be not significant or at least to be not the main limiting factor for the BER. This assumption, however, depends on the number of carriers used in OFDM and the number of points in the constellation. For a carrier number greater than 32 and for a mapping with the number of points greater than 16, the common phase error should be carefully estimated and equalized.

In figure 6.16 is shown the impact of the simulated phase noise for 60 GHz transmission. All other channel parameters are set to zero, including the variance of AWGN noise. The figure shows the constellation after equalization. The encoded sequence can be decoded with zero bit error rate however, one can see quite large spreading of the symbols in I-Q domain.

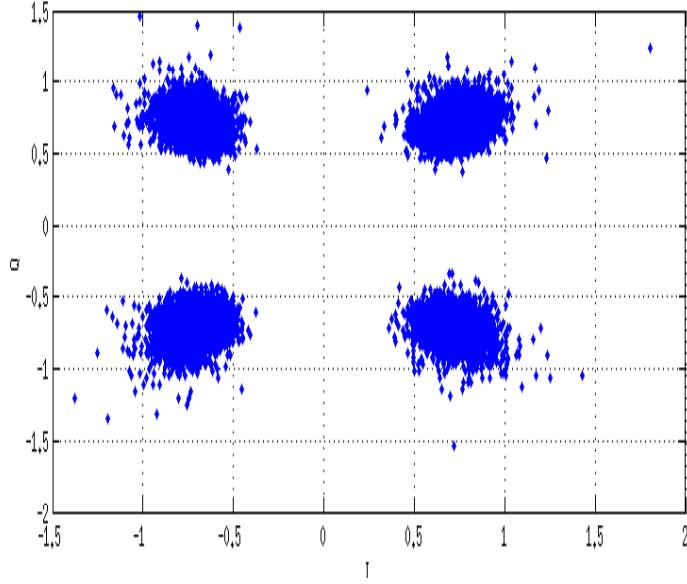


FIGURE 6.16: Phase noise simulated for 60 GHz system

The model of the channel is shown in figure 6.17, where the above mentioned distortions are included. The LO leakage depends on the hardware and particularly on multiplexors. Therefore, LO leakage is not included in the channel model.

The degradation of the signal due to the phase noise can be easily seen if one compares the eye diagrams for a simple AWGN channel and a channel with the same noise variance

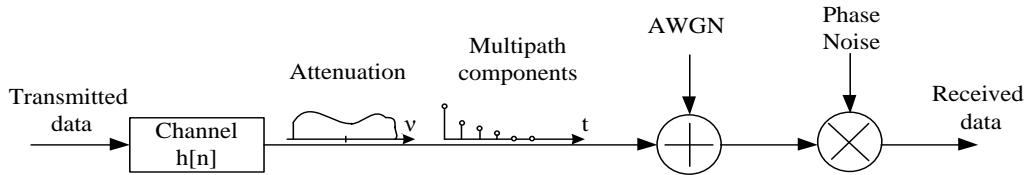


FIGURE 6.17: Channel model

but with the phase noise. These plots have been made using the channel impulse response that we estimate in the following section and the 60 GHz phase noise model.

The eye diagrams are shown in figures 6.18 and 6.19 respectively. One can observe clearly a significant eye closure that appears because of the phase noise.

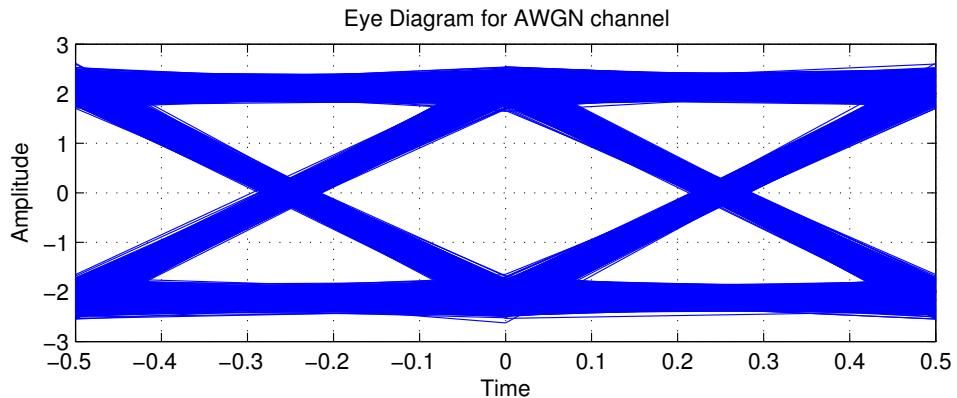


FIGURE 6.18: Eye diagram for AWGN channel

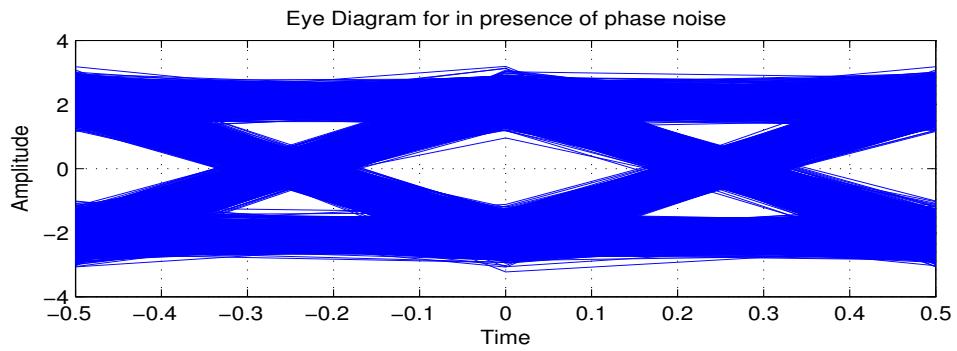


FIGURE 6.19: Eye diagram for AWGN channel in presence of the phase noise

6.3.2 Channel measurements

For 5GHz, the frequency response of the channel is measured by transmission of the pseudo noise sequence within MATLAB environment. The channel frequency response is shown in Figure 6.20. As one can see, the channel can be assumed almost flat for all frequencies within the range $|\nu| < 0.4$. The peak in the middle of the band correspond to the LO leakage. Consequently, it is not a part of the channel which is really good but a part of the channel where the noise is really powerful. It is the reason why this part of the channel is not used. This facts are used later for carrier allocation and pilot frequencies.

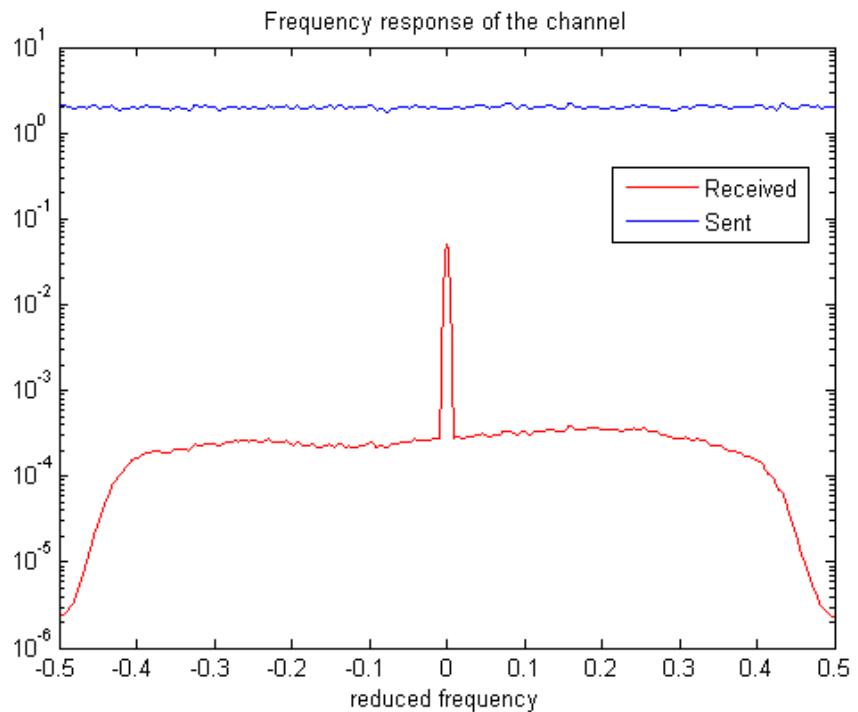


FIGURE 6.20: Power spectrum of the both transmitted and received sequence for 5 GHz

In order to estimate the channel impulse response, we transmit pseudo noise. The channel impulse response should correspond to the best filter which estimates the received signal from the transmitted signal. This best filter is the Wiener filter.

Figure 6.21 shows the channel impulse response for 5 GHz channel. Her for simplicity we use an FIR Wiener filter with 5 non-causal taps and 6 causal taps. As the synchronization is done using the maximum of the cross-correlation, $T = 0$ is defined as the most powerful tap. It happens that it is not the first one, this is the reason of the non causality of

the filter. However, there is a LO leakage, this can biased the estimation because the Wiener filter assume white noise and it is not.

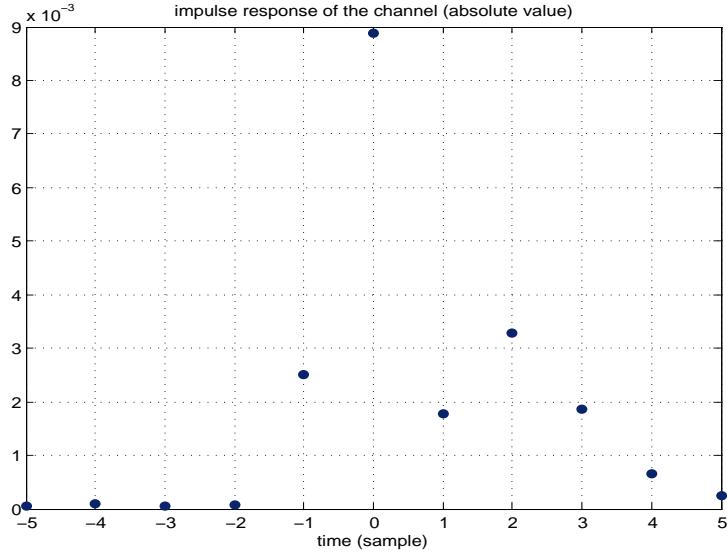


FIGURE 6.21: Channel impulse response for 5 GHz

From these graphs, one can deduce the length for the cyclic prefix and suffix. Utilization of cyclic suffix in OFDM systems is not found to be usual. However, it is a necessary parameter for this particular implementation. To show this, we simulate 5 tap-channel with two taps at time instants -1 and -2. The phase noise and AWGN are removed from the channel model. As one can see in figure 6.22, the most noisy constellation is recovered when there is no CP and CS. The insertion of CP reduces the inter symbol interference, which can be seen in closer position of the equalized received data. However, the spreading is still quite large and might result in errors after detection. The best constellation (shown in black) we got when there is cyclic prefix inserted and the improvement is significant.

To illustrate the importance of the cyclic prefix and cyclic suffix for OFDM implementation, the eye diagrams are analysed. First, the eye diagram for the case when all channel distortions are set to zero, except the AWGN and channel filtering is shown. The eye diagram shows that there is still some interference in the system. However, for this case the transmitted data can be recovered.

In the next eye-diagram in figure 6.24, one can see the negative impact of the cyclic suffix removal. The noise variance has increased which results in eye closure. Since we

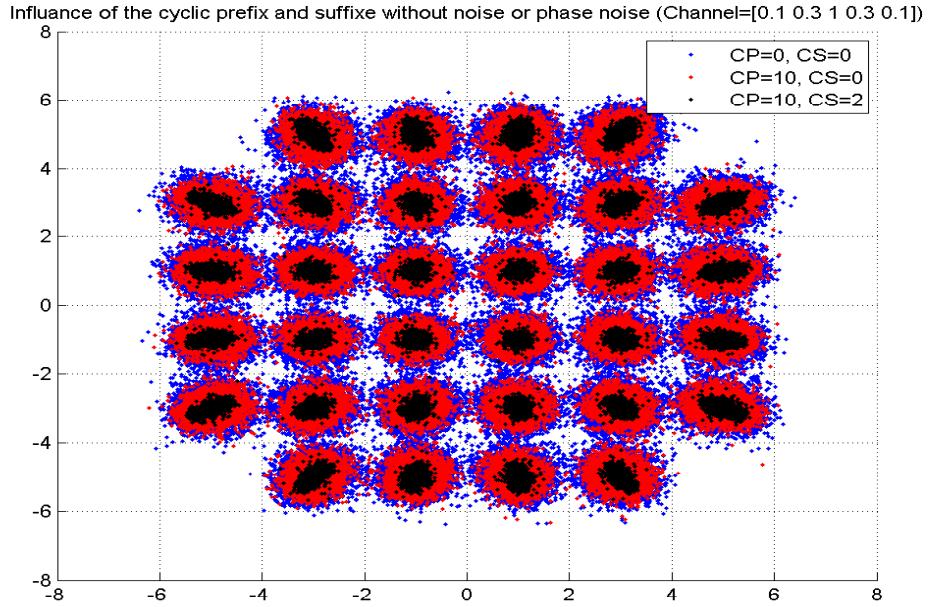


FIGURE 6.22: 32-QAM constellation for different values of prefix and postfix

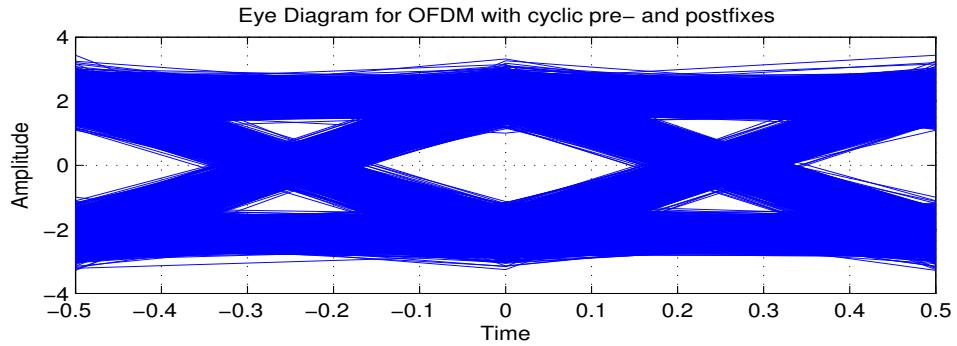


FIGURE 6.23: Eye diagram with chosen CP=12 and CF=4

have just one non-zero tap in the 'future', the degradation of the eye diagram, although quite severe, leaves still "the eye" visible. As shown in figure 6.25, wrong choice of the CP can affect the data much more severely. The plot is obtained for the case, when the CS remains untouched but the length of the CP is set to 3. Here one can clearly see that with short cyclic prefix (shorter than the number of taps in the channel), the signal is exposed to inter-symbol interference that destroys the data.

Another parameter that had to be considered, is the communication distance. The wave propagation properties for 5 GHz and 60 GHz differ significantly. For 5GHz he estimated communication range for LOS communication was estimated to be approximately equal to 8 – 10 meters. The signal is subjected to severe fading at some frequencies with the increase of the distance. For the distances greater than 10 meters, the received sequence was too noisy to be decoded even when turbo code was used. In Figure 6.26

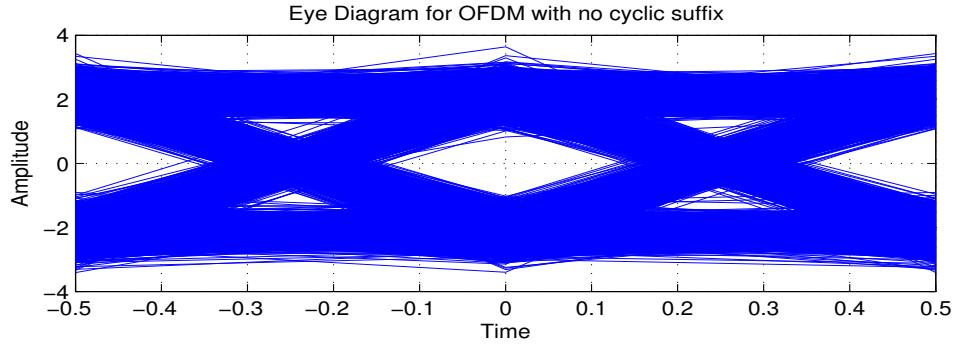


FIGURE 6.24: Eye diagram with chosen CP no CS

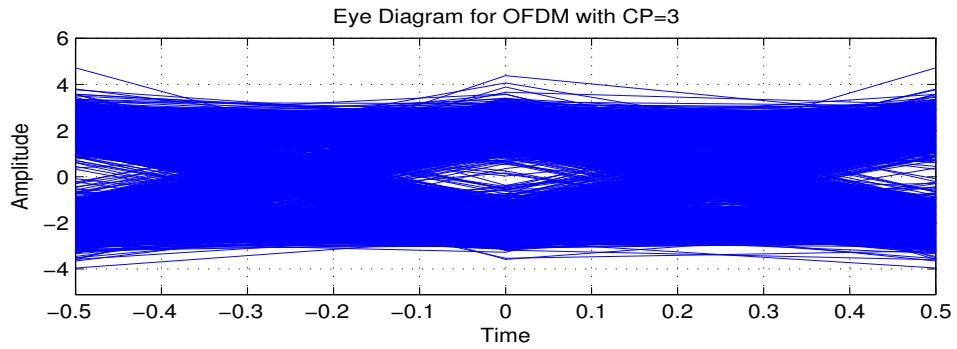


FIGURE 6.25: Eye diagram with CP=3 and CS=4

are plotted the power spectrum of the transmitted OFDM signal for 128 carriers and the estimated channel frequency response. In addition to the attenuation at high frequencies, that is present at close range transmission with the distance of about 1 m (see figure 6.20), here fading appears also at low frequencies.

6.4 From 5 to 60 GHz

Up to this point the project work, although pursuing the goal to build a 60 GHz communication system, was performed with primarily 5 GHz hardware equipment. In MATLAB simulation, only the phase noise parameters corresponded to 60 GHz frequency. Other estimations were performed for 5 GHz, including channel impulse and frequency response, frequency and phase offsets. The algorithms developed for estimation of the parameters were trying to take into account possible problems of 60 GHz communication channels, such as ISI, fading, frequency, phase offsets and phase noise. However, it was difficult to make a valid forecast for the real 60 GHz communication system and our goal when working with 5 GHz frequency was primarily to implement it in hardware and achieve as high effective bit rate as possible.

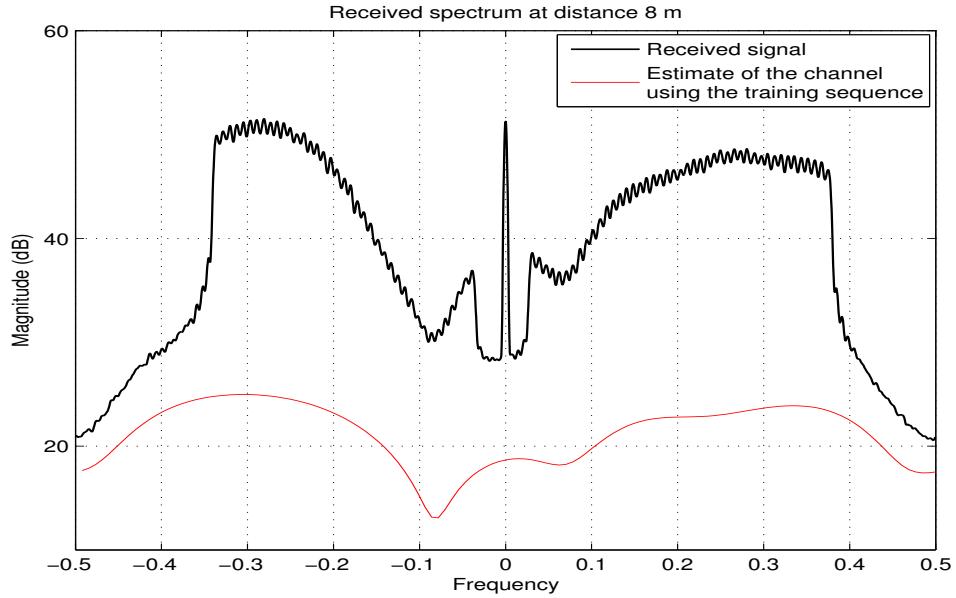


FIGURE 6.26: Distorted signal

The 60 GHz transmission system was built by using USRPs, with on-chip integrated antennas, both for the transmitter and the receiver. The basic MATLAB code for transmission and reception was provided together with the equipment. To test the system, we performed a series of transmission tests presented in the next section. The distance between transmitter and the receiver was minimized to around 15 cm. This was in some sense an optimal scenario, but since we had to estimate a lot of parameters, we started with a simplified model.

6.4.1 Frequency offset

First, we transmitted the constant Wave signal. The exponential signal was received without any problems; however, one could observe a significant frequency shift. The plots of the transmitted and received constant Wave signals are shown in figures 6.27 and 6.28 respectively.

From the figure 6.28 one can see that there appears a frequency offset $\Delta\nu$ that shifts the transmitted exponential signal from $\nu = 0.2$, to $\nu = 0.1884$.

The value of the offset varies in time, and for different transmission tests, one obtained slightly different results. The frequency offset, that for the cases, given in figures 6.27 and 6.28, equals to $\Delta\nu_{off} = 0.0116$ includes two different terms: a slowly varying carrier frequency offset (CFO) and a fast varying frequency offset, that appears due to

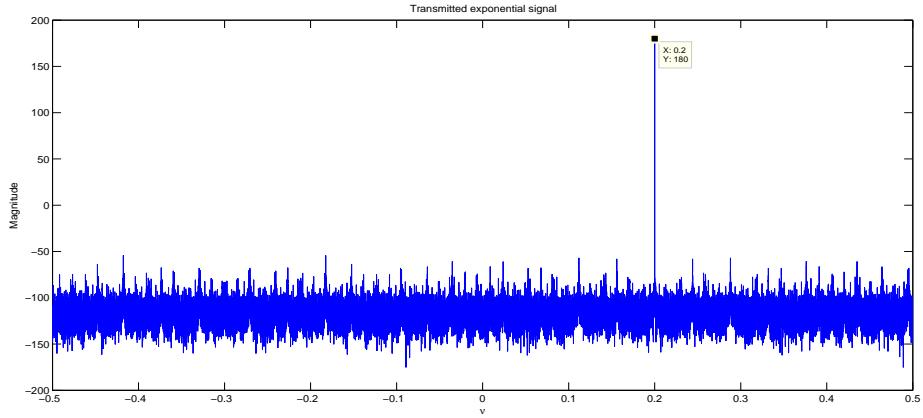


FIGURE 6.27: Power spectrum of the transmitted exponential signal, $\nu = 0.2$

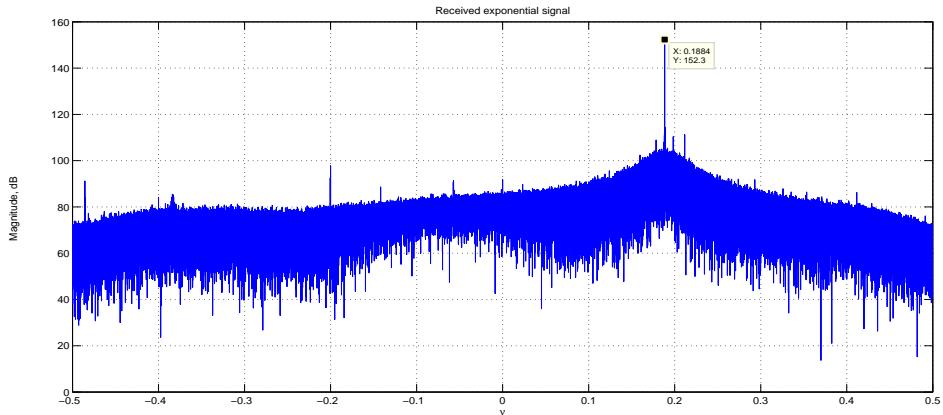


FIGURE 6.28: Power spectrum of the received exponential signal, $\nu + \Delta\nu_{off}$

the phase noise. The CFO can be assumed to be constant for the duration of several OFDM symbols and can be estimated by a relatively simple algorithm. The phase noise term, due to the high frequency used, require a model to be tracked through the symbols and is not tracked in the current implementation. As mentioned above, the phase noise results in two separate effects on the transmitted data. Common phase error and ICI demand different algorithms for their estimation and correction. It was relevant to first estimate the extent of the frequency offset $\Delta\nu(t)$ and how it was affected by each of the above terms.

6.4.2 Channel measurements

Next, in order to investigate the channel, we stop the transmitter and listen to the channel. In figure 6.29 one can see the spectrum of the received noise with several high amplitude peaks. These bursts will negatively affect the transmitted signal, introducing distortion. One also can observe severe attenuation of the high frequency components.

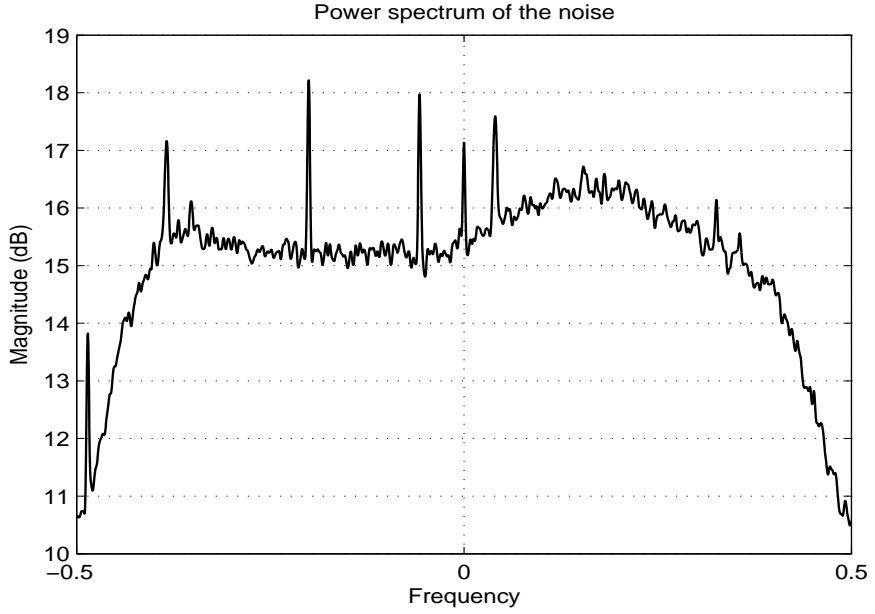


FIGURE 6.29: Power spectrum of the received noise at 60 GHz

Parameters	Value
Number of carriers	32
Number of used carriers	17
Number of pilots	6
Mapping	QPSK
Cyclic Prefix	10
Cyclic suffix	4

TABLE 6.1: Parameters 60GHz transmission

6.4.3 Modifications in the algorithms

To be able to see how the system, that was developed using 5 GHz equipment, worked with the 60 GHz equipment, we generated and transmitted an OFDM signal with the parameters given in table 6.1.

6.4.3.1 Synchronization

There were several problems with transmission over 60 GHz that needed to be corrected in order to be able to decode the data. First, the synchronization failed because of the transient process at the receiver. It resulted in a large peak that appeared in the beginning of the crosscorrelation function. In figure 6.30 the crosscorrelation of the received data with the training sequence is shown.

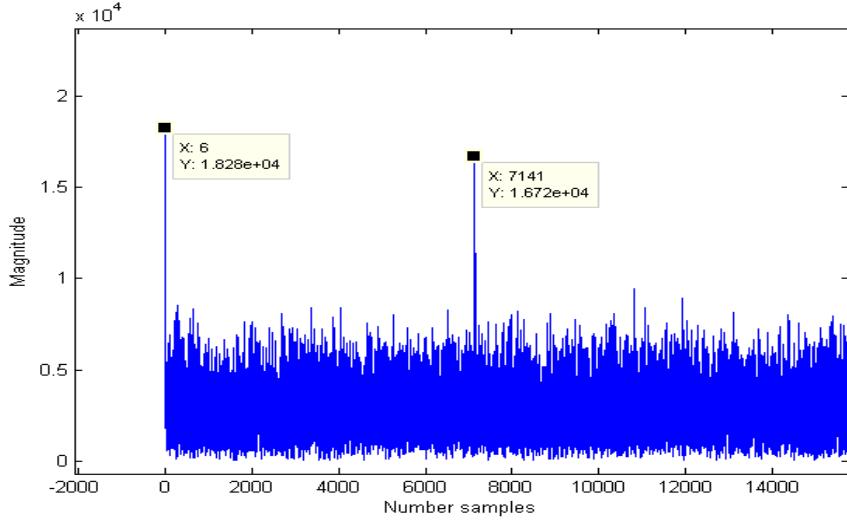


FIGURE 6.30: Crosscorrelation of the received data with the training sequence at 60 GHz, $\nu = 0.2$

The synchronization function detected an erroneous maximum and failed to synchronize the received data. To get a correct sampling time instant, which corresponded to the second peak in the figure, the first 100 samples of the crosscorrelation were set to 0.

Then, to get a correct synchronization, the length of the training sequence has been chosen to be as short as possible to avoid problems due to the frequency offset. The upsampling rate was of $Q = 2$. The spectrum of the transmitted signal is shown in figure 6.31. The spectrum of the received signal is shown in figure 6.32.

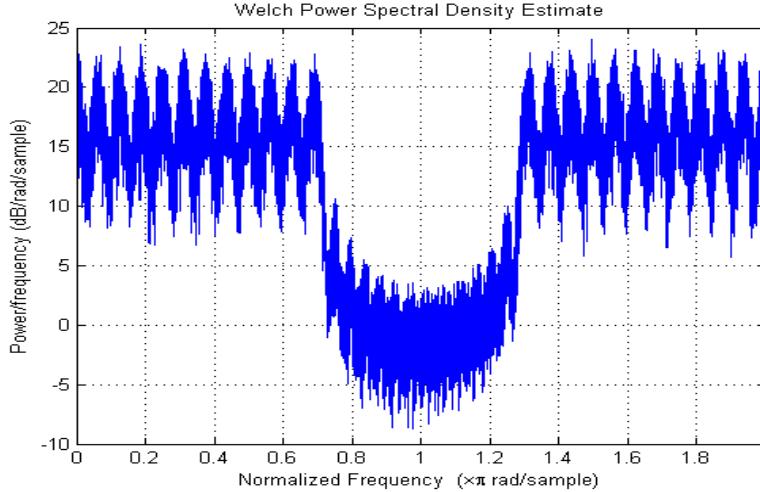


FIGURE 6.31: Power spectrum of the transmitted OFDM system at 60 GHz

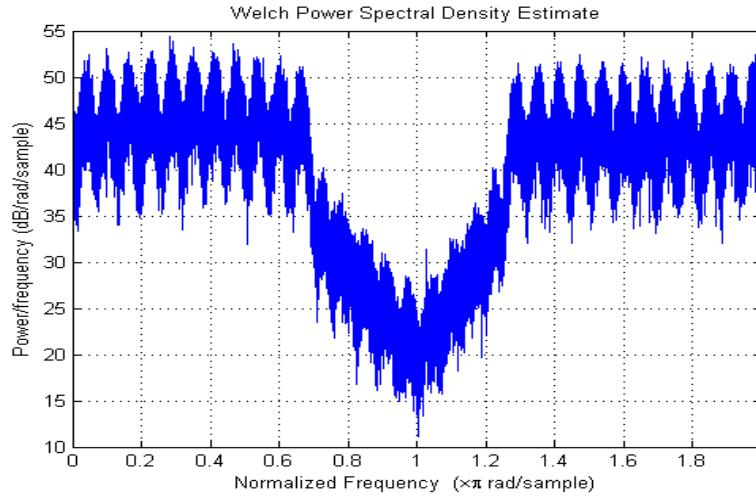


FIGURE 6.32: Power spectrum of the received OFDM system at 60 GHz

6.4.3.2 Frequency offset

As mentioned above, a frequency offset occurred, which resulted in phase variations. In figure 6.33a the phase variations for each of the used information carriers are shown, labeled in different colors. One can see, that the phase changes significantly within one OFDM symbol. These phase variations were produced by several types of distortions.

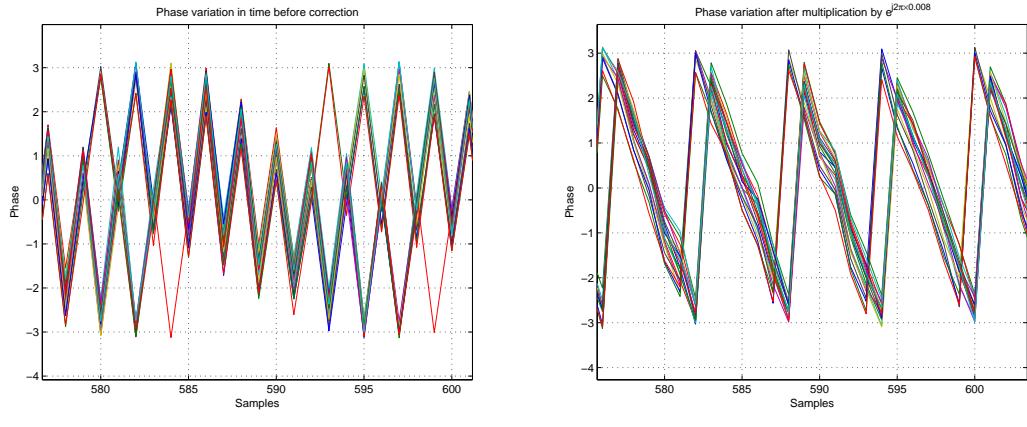
(a) Of the received signal for 60 GHz channel (b) After multiplication by $e^{j2\pi \times 0.008}$

FIGURE 6.33: Phase variation

In order to explore the severity of the impact from the residual frequency offset, we multiplied the received signal by an exponential $e^{j2\pi \times 0.008}$ which corrected the phase change. The value 0.008 was chosen empirically by trying several values, close to the one we observed when transmitting the Continuous Wave signal. Figure 6.33b demonstrates the significant improvement, compare to 6.33a, in phase variations after residual frequency offset was corrected. Obviously, this correction is not perfect, but it signifies an important fact for the system design. Let us now consider the phase variations in

the frequency domain in one OFDM symbol. Figure 6.34 shows the phase for several OFDM symbols. The variations of the frequencies that were used to carry information (without frequencies above $|\nu| = 0.4$) were not fast. The conclusion that followed, from these facts, was that the phase variations can be corrected by correcting the residual frequency offset and common phase error. This was expected to be sufficient for recovering the transmitted sequence, possibly with acceptable number of errors that can be avoided using channel coding. For 32 frequency subcarriers, the ICI impact on the system could be left uncorrected.

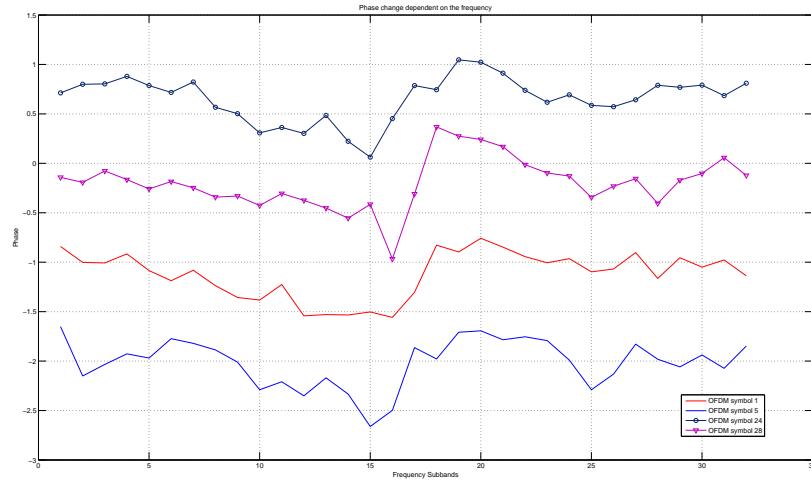


FIGURE 6.34: Phase variation in frequency

6.4.3.3 Channel Equalization

The QPSK constellation before the equalization was very noisy and is shown in figure 6.35a.

The received sequence was processed at the receiver using the same algorithm for channel estimation and equalization as one, used for 5 GHz. The equalized sequence, after correction, is shown in figure 6.35b. One can see the impact of the saturation of the amplifiers at the transmitter as well as the negative effect of the phase noise. Both of them resulted in spreading of the signal points in the IQ domain. The effect of the phase noise, however, did not degrade the received sequence significantly, compared to the 5 GHz transmission. The saturation, on the other hand, which resulted in non-linearities of the transmitter's power amplifier strongly affected the signal, producing additional noise.

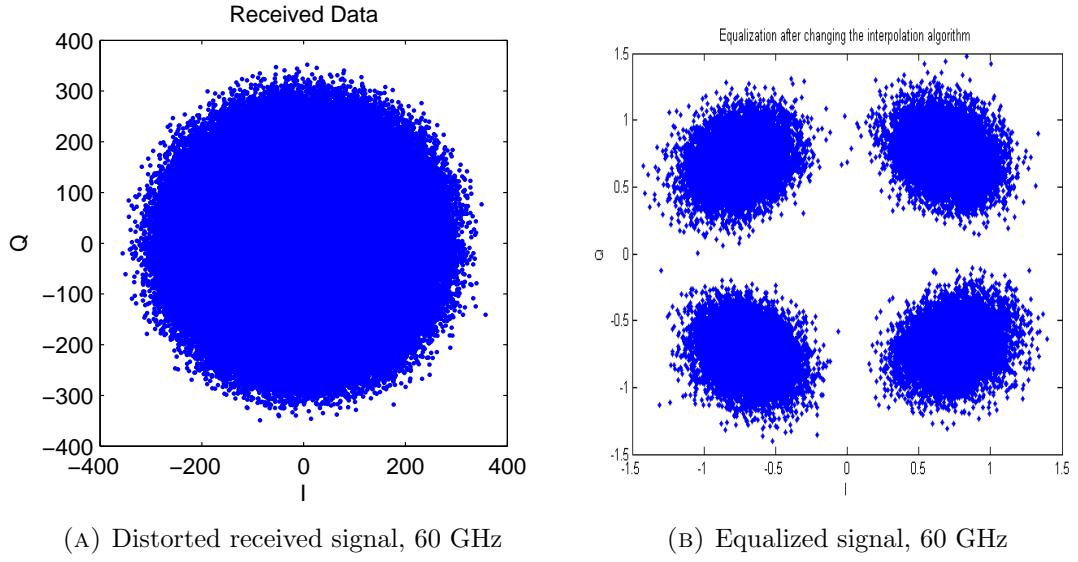


FIGURE 6.35: Constellations at 60 GHz

In the first transmission at 60 GHz in offline mode, the achieved bit error rate for the data block of length 100 000 was $2 \cdot 10^{-5}$.

6.4.4 Real environment characterization

In this section, the impact of different parameters like the distance, gain, pilot and data carriers allocation are studied.

6.4.4.1 Distance

For 60 GHz channel the dependence on the distance became more significant as well as mutual disposition of the receiver and the transmitter. The system was tested at different distances but the results are quite ambiguous even though they correspond mostly to expected ones. In figure 6.36 are shown obtained curves. In general, the BER increases with the distance due to the attenuation and absorption. But there are some abrupt jumps in BER which is hard explainable just by distance change. It may occur due to other processes, unrelated to the actual distance between two ends like a destructive multipath.

6.4.4.2 Saturation issue

Another significant effect that severely degraded the performance of the system was saturation of the amplifier at the transmitter. This resulted in non-linearities and increased the noise in the constellation. Comparing the plots obtained for different values of the

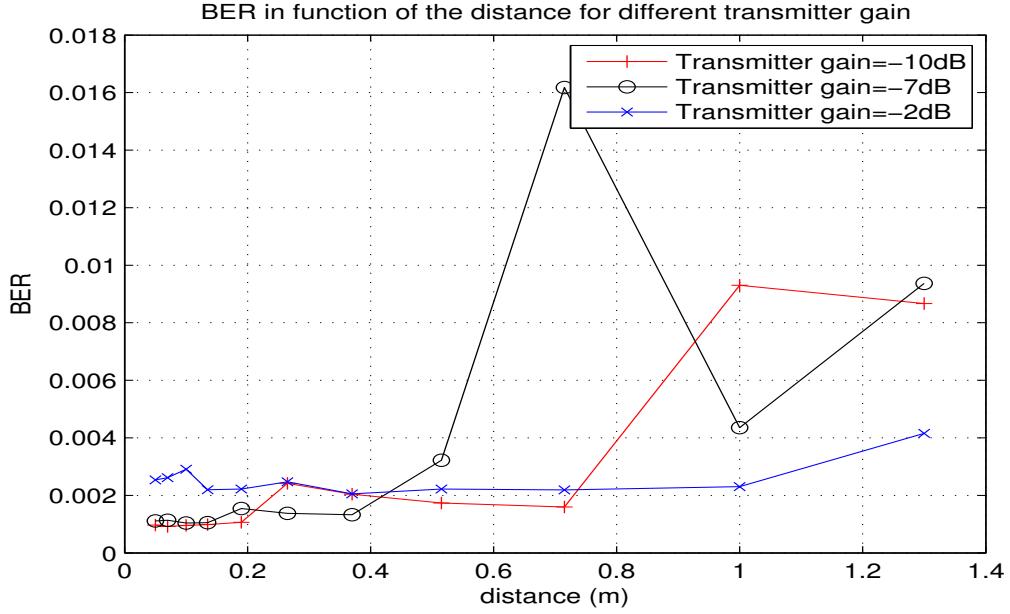


FIGURE 6.36: BER vs distance for 60GHz

transmitter gains, one can notice that there is a significant spreading between the received symbols within the IQ domain when the gain equals 10. Figure 6.38 demonstrates also the impact of the phase noise.

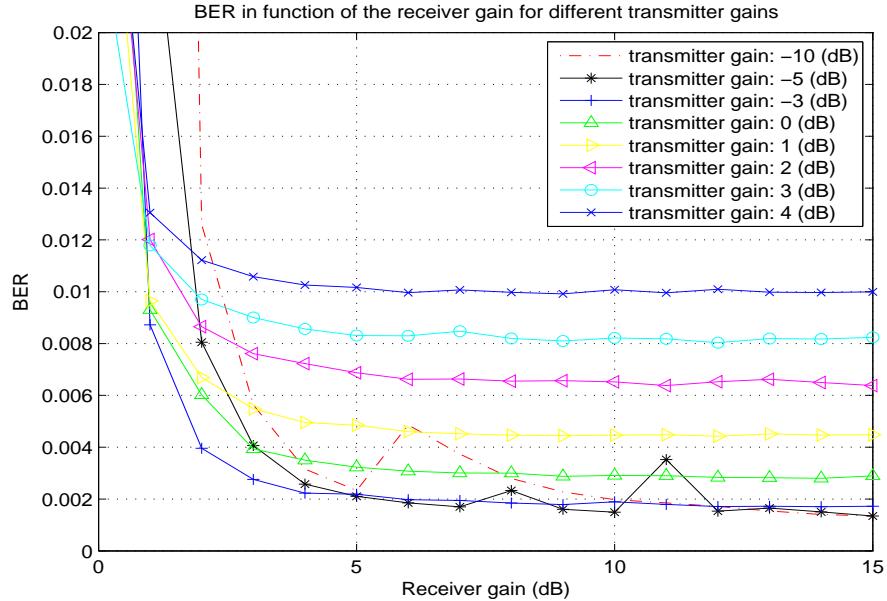


FIGURE 6.37: BER in function of the gain for 60 GHz transmission

The saturation strongly depended on the gain that was chosen for the receiver and the transmitter and the RMS value of the transmitted sequence. To investigate the impact

of the chosen gain to BER, several tests were performed with different values for transmitter and receiver gains.

Figure 6.37 shows obtained curves of this experiment¹. Notice that the BERs dependence on the receiver gain was constant after a certain value and equalled to a gain of $4 - 5 \text{ dB}$. However, a faulty choice of the receiver's gain could lead to saturation and a considerable increase in the BER. The negative effect of the saturation was more noticeable compared to the phase noise, and it was one of the more limiting factors when trying to achieve good BER. The gain of the transmitter, on the other hand, should be chosen carefully and according to the plot, 0 dB provides both low BER and is constant for a large range of values of the receiver gain.

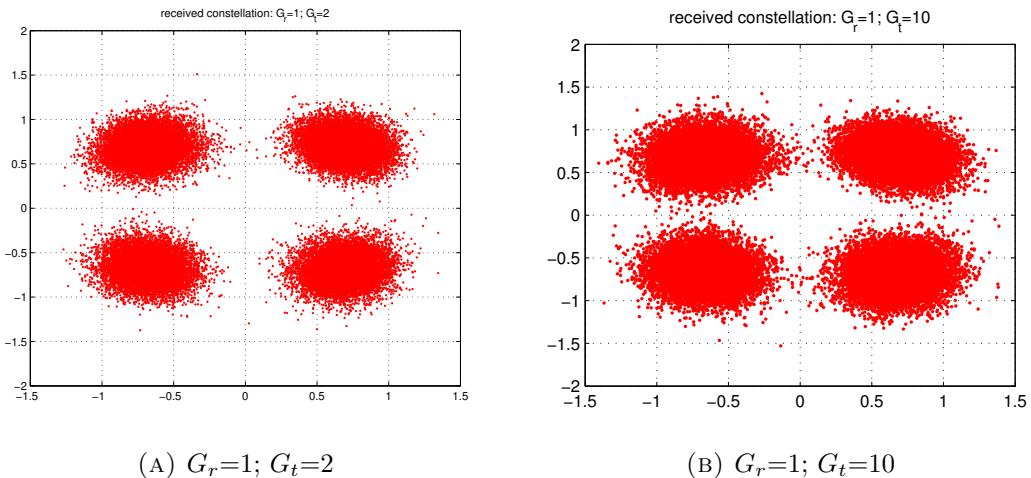


FIGURE 6.38: Received constellations for different gains

6.4.4.3 Frequency and Impulse responses for 60 GHz channel

The wide channel situated at 60 GHz also creates problems, since its characteristics are not constant for all frequencies. The estimated channel impulse and frequency responses are shown in figure 6.39.

From figure 6.39 one can see that channel impulse response has non-negligible taps situated the sampling instances from -2 to 2 . There are no other transmitters at 60 GHz, that can interfere with our system and the properties of 60 GHz guarantee impossibility of real multipath propagation, since the attenuation is very high. Therefore, the CIR more likely can be explained by hardware influence, namely filters, that create effect of multipath.

¹The negative gain is obtained by changing the RMS of the signal. The odd points are due to quantization noise

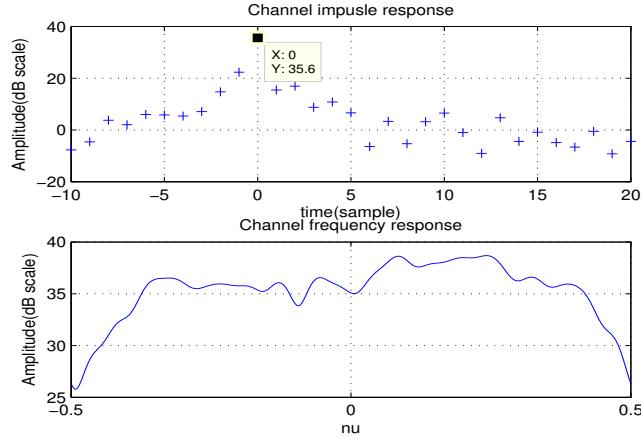


FIGURE 6.39: Channel impulse and frequency responses at 60 GHz

The channel gain can be assumed to be almost constant for all frequencies. Therefore, there is no need of tracking the channel gain within one OFDM symbol. This fact is used later in the estimation algorithm for DFT-precoded OFDM(see chapter 7). To demonstrate to which extent some particular frequencies affect the transmitted signal, the BER is computed for each subcarrier. The performance of the system depending on the frequency is shown in Figure 6.40. Comparing figure 6.40 with the channel frequency response, plotted in figure 6.39, one can observe, that the achieved BER corresponds to the channel variations. For the frequencies $\nu \in]0.5, 1]$, where the attenuation is stronger, the BER increases as expected.

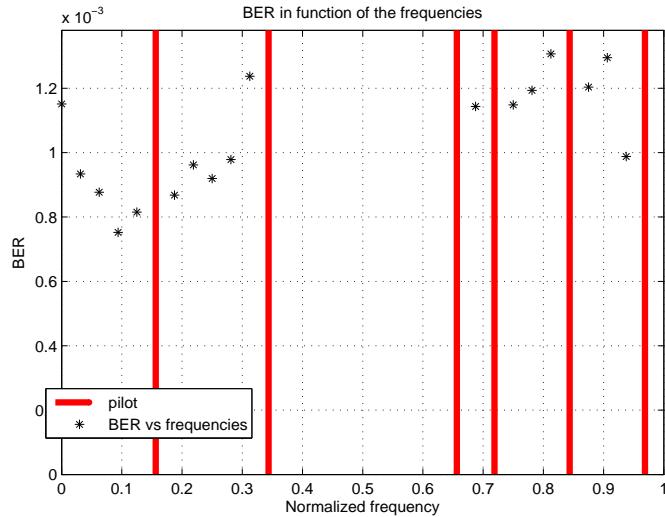


FIGURE 6.40: BER as a function of the frequency

6.4.5 PAPR

Considering the saturation problem, Peak-to-Average-Power-Ratio, PAPR, becomes one of the primary factors that should be controlled in the system. As mentioned above, a considerable drawback of the OFDM systems is that they result in high PAPR. This is an important issue that results in non-linearities of the power amplifiers used in the system and signal distortion caused by them. The PAPR is computed according to [Rugumira et al., 2011] as given in 6.26, where $\max|d[n]|^2$ is the maximum amplitude of the transmitted signal and the $E|d[n]|^2$ is the average power of the signal.

$$PAPR = \frac{\max|s[n]|^2}{E|s[n]|^2} \quad (6.26)$$

The value of the PAPR greatly depends on the number of carriers used in the system, as well as the chosen mapping. The peak value of the power can differ, from the average power, by factor of N. This problem has been thoroughly studied by a vast number of researches and several methods have been proposed to mitigate it. Among them one of the methods used in LTE mobile communications, and it has been proved to be efficient in reduction of the PAPR, DFT-precoded OFDM. After that, the same procedure that is used in a classical OFDM system, was performed. It combined the advantages of single-carrier transmission and multi carrier transmission and reduced the PAPR. This system is simulated and investigated in chapter 8.

6.5 Performance analysis

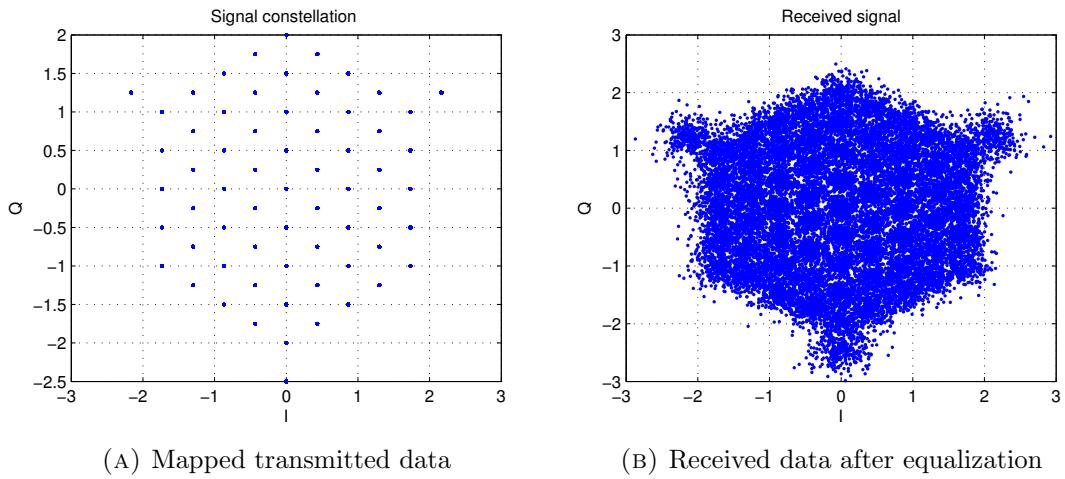
This section investigates the results for both simulated in MATLAB system and real transmissions that were processed with MATLAB. It also presents the different design parameters of our system.

Over 5GHz, to achieve maximum data rate, the transmit data is mapped according to the constellation shown in 6.7 and 128 carriers of OFDM. The transmitted mapped data is shown in figure 6.41a and the received data after channel equalization is shown in figure 6.41b. One can see that the equalization can be considered successfully, however, there is still quite large amount of noise.

With turbo coding rate $R = 0.8$ achieved effective bit rate is 72 Mbits/s achieved in simulation with BER=0. The effective rate is computed according to 6.27. However, the frame error rate was of 10%.

$$\text{Rate}[M\text{bits}/\text{s}] = \frac{\text{number of information bits}}{\text{total number of samples transmitted}} \times R \times 25 [\text{Msamples}/\text{s}] \quad (6.27)$$

This number was, of course, achievable only for 5GHz and was for that reason optimistic, since in 60GHz implementation appeared several limitations that reduced the rate. Furthermore, for a real time implementation, one of the important issues is the time of computation, since Turbo code used in simulation may require several iterations before the data can be recovered. However, this bit rate, although achieved only in MATLAB environment, signified the potential of developed system and motivated its further improvement.



It was not an easy task to analyze the implemented system since it had several aspects to be considered and since there was always a trade off between different characteristics, that one aimed to achieve: robustness and BER, ease of implementation in C++ and real-time performance, or maximum bit rate. It is also hard to compare it with the system 1. First, because these two systems were developed with different focus and second, because their complexity and methods were almost completely different. Therefore, this section aims to analyse second model by investigation of the parameters that can be adjusted in it and by analysis of the algorithms it uses, rather than to compare it with system 1.

6.5.1 Kalman Filtering

Kalman filtering demands quite large computational resources. This is of particular importance for real-time implementation. Therefore, it was interesting to investigate in simulations, how much do we gain in performance when it is used for equalization. In

figure 6.42 are plotted the received QPSK constellations after channel equalization using different methods, namely just linear interpolation, linear interpolation and filtering, and spline interpolation with filtering.

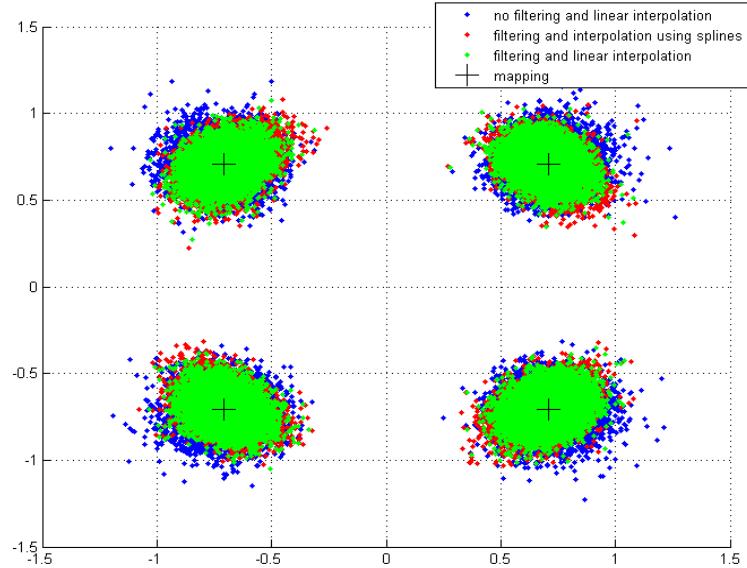


FIGURE 6.42: Comparison of diffent equalization techniques

One can see, that filtering and linear interpolation gives the best results. The difference between equalization with the Kalman filter compared to equalization that uses only linear interpolation is noticeable and can be seen as different spreading of the points in the IQ plane. However, it does not give a significant gain. This is of course more relevant for low number of points in the constellation. For 64-QAM even this difference can result in moving to another decision region and greater error probability. Therefore, both for simulation and real implementation we are using both Kalman filtering and linear interpolation. Figure 6.43 shows obtained BER for identical signals transmitted over AWGN channel in presence of phase noise for 16-QAM constellation. For all values of SNR, equalization with Kalman filtering proves to be more reliable and gives better result.

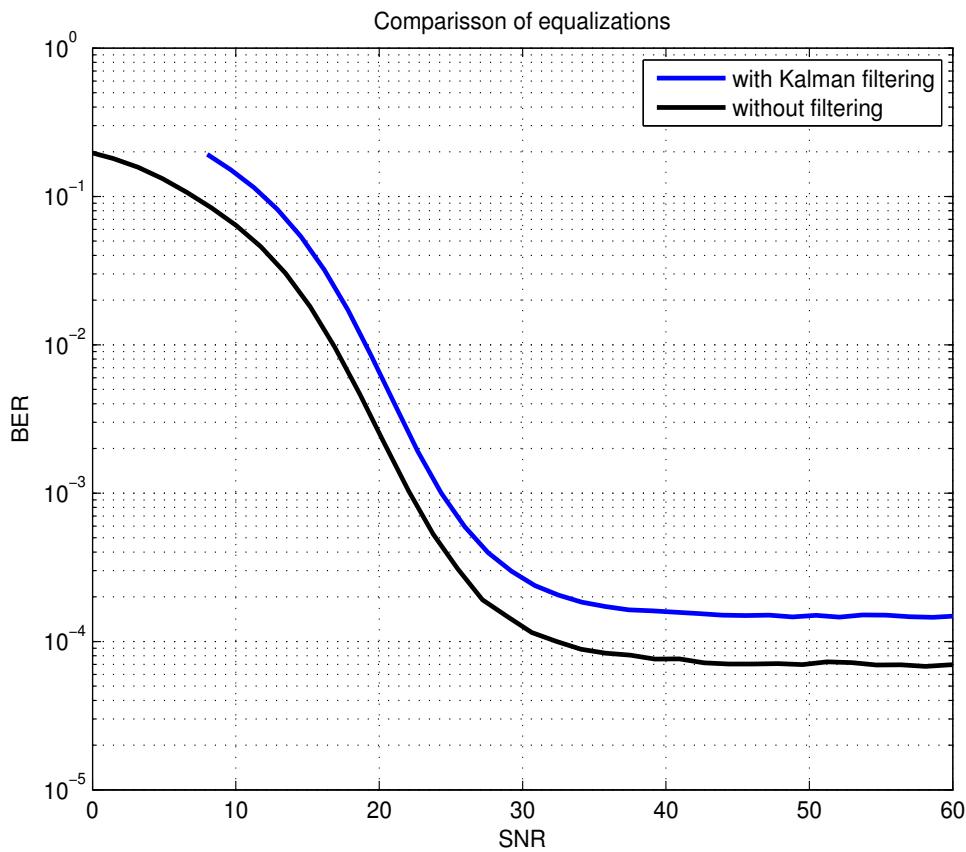


FIGURE 6.43: Comparission of equalization techniques, with 16-QAM

6.5.2 Constellations and phase noise

In order to investigate system's performance in presence and absence of the phase noise, there were performed several tests for different mappings and with different channels. First, the AWGN channel is simulated. The obtained BER plots for some constellations used in this project are shown below in figure 6.44.

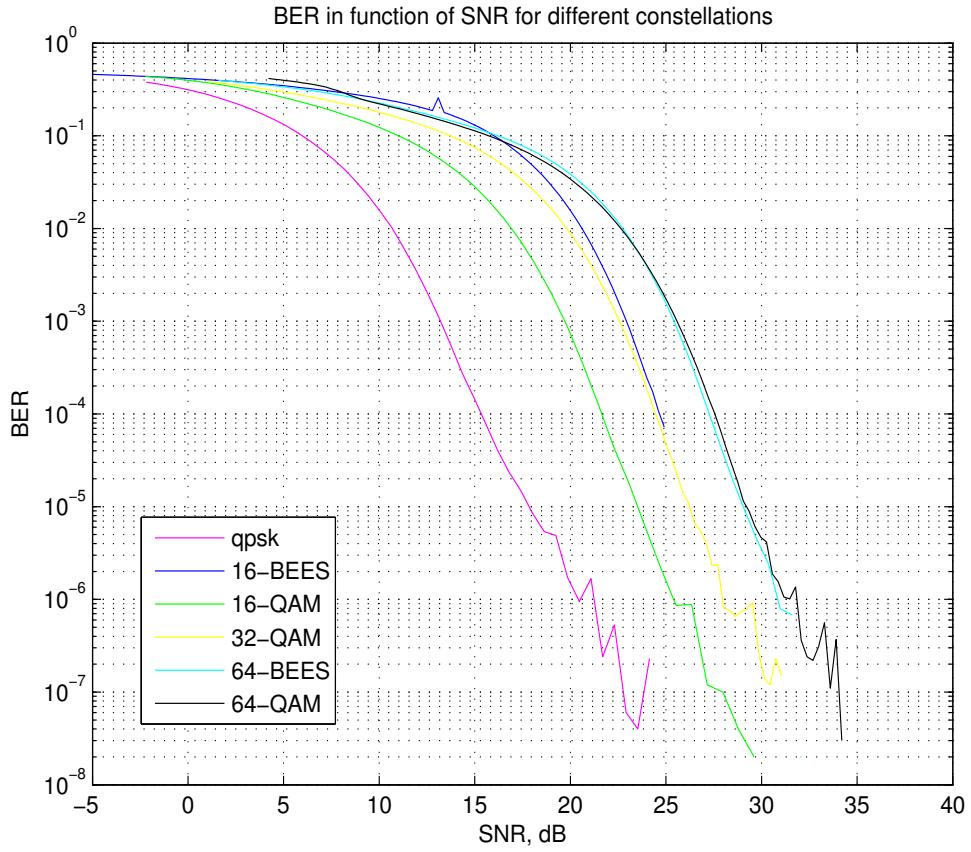


FIGURE 6.44: BER for AWGN

As expected, with the increase of number of points in the constellation, the BER is degrading. One can also see, that energy efficient constellations 16-BEES and 64-BEES show different performance.

The constellation, that was designed to improve the PAPR of the system with 16 points (conventional name "BEES16"), proved to be worse (in simulations and real transmission) than classical 16-QAM.

For real transmission, the received constellations, are shown in figure 6.45a and 6.45b and their comparison is shown in figure 6.46. Despite that the maximum energy of BEES16 is lower than for 16-QAM, since the distance between adjacent points is much lower, it results in more errors.

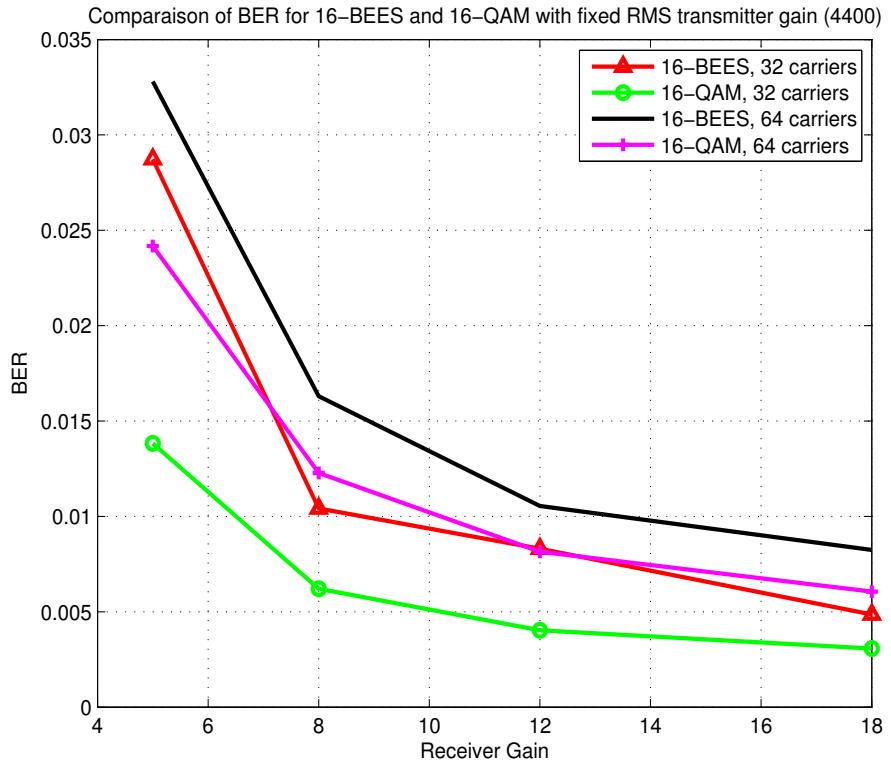
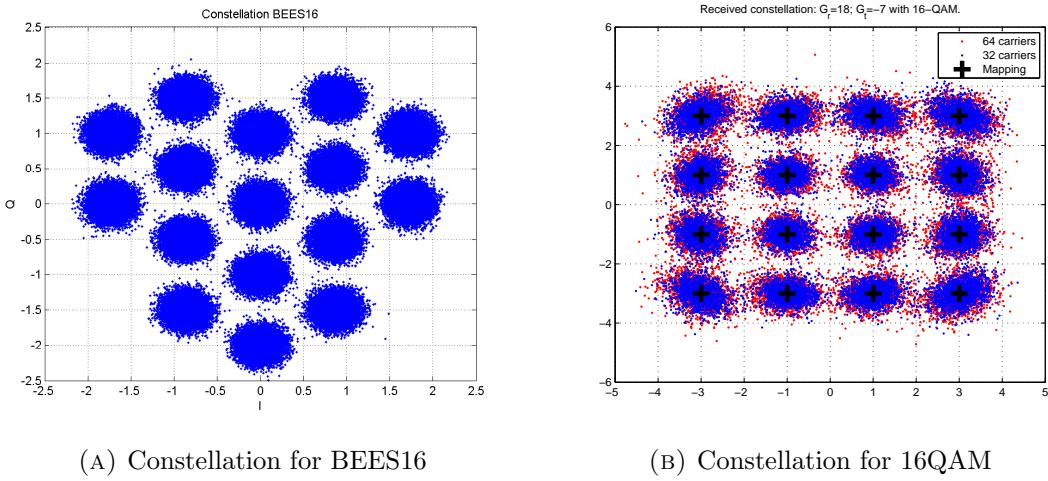


FIGURE 6.46: BER for 16-QAM and BEES16

On the other hand, 16-BEES, 32 carriers performs slightly better in low SNR and high SNR, whereas for SNR values from 10 to 25 dB its performance show worse results.

Next, the channel with phase noise is simulated and the performance is measured for several different constellations. In figure 6.47 can be seen the difference in BER for the channel with and without simulated phase noise.

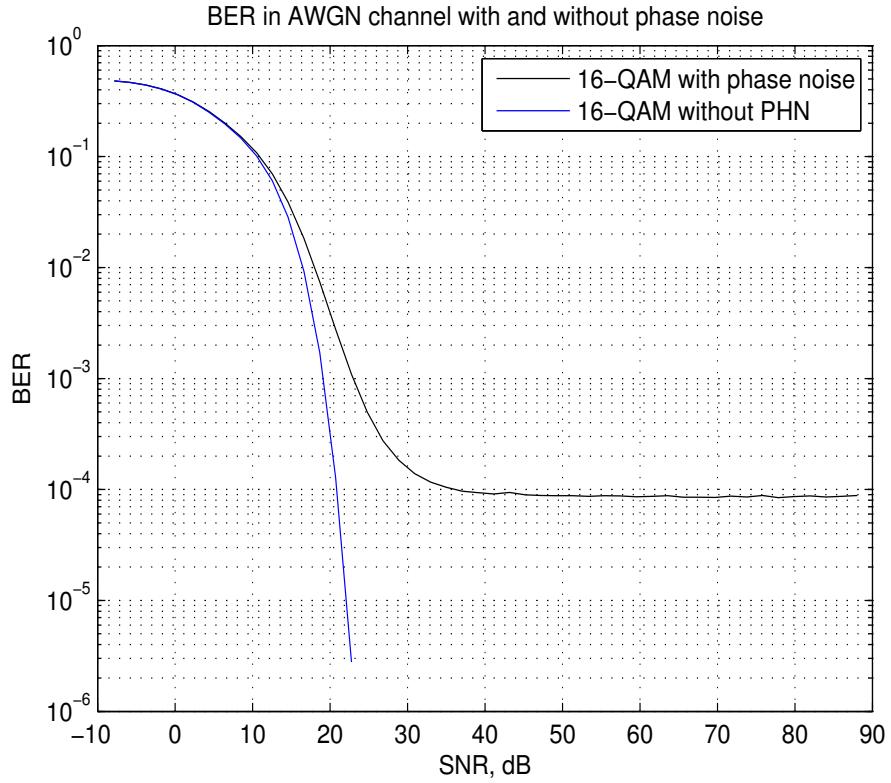


FIGURE 6.47: BER for 16-QAM

As one can see in figure 6.47, there is no major difference between these two channels, however this is only in simulated environment. Phase noise creates a floor in BER at high SNR and becomes the dominant factor that affects it.

6.5.3 Gains, RMS and BER

First, to illustrate the impact of the increase of the receiver gain to BER and the to analyse how robust in terms of resulting errors are chosen constellations , the test is performed with fixed transmitter gain and fixed RMS of the transmitted data and fixed distance of approximately 50 cm. Figure 6.48 shows BER in function of the receiver gain for classical M-QAM constellations as well as modified constellations. One can notice, that for constellations where M is low, the improvement of the BER with the increase of gain is practically unnoticeable. All constellations, except QPSK show some dependence on the receiver gain, therefore, if one aims to achieve better BER, one possible solution is to increase the receiver gain. This test is performed for an OFDM signal with 32 carriers. Next, similar experiment is made but for 64 carriers. The results of this experiment are

shown in figure 6.49. The curves have similar shape than in figure 6.48. However the number of errors increases for 64-BEES.

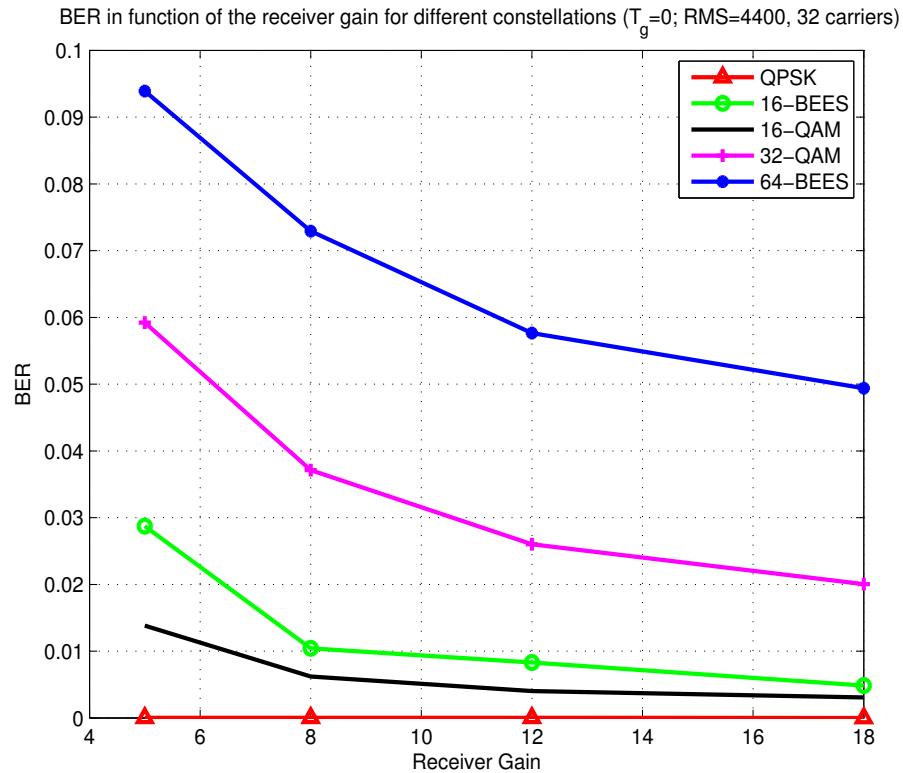


FIGURE 6.48: BER in function of the receiver gain for different constellations, 32 carriers

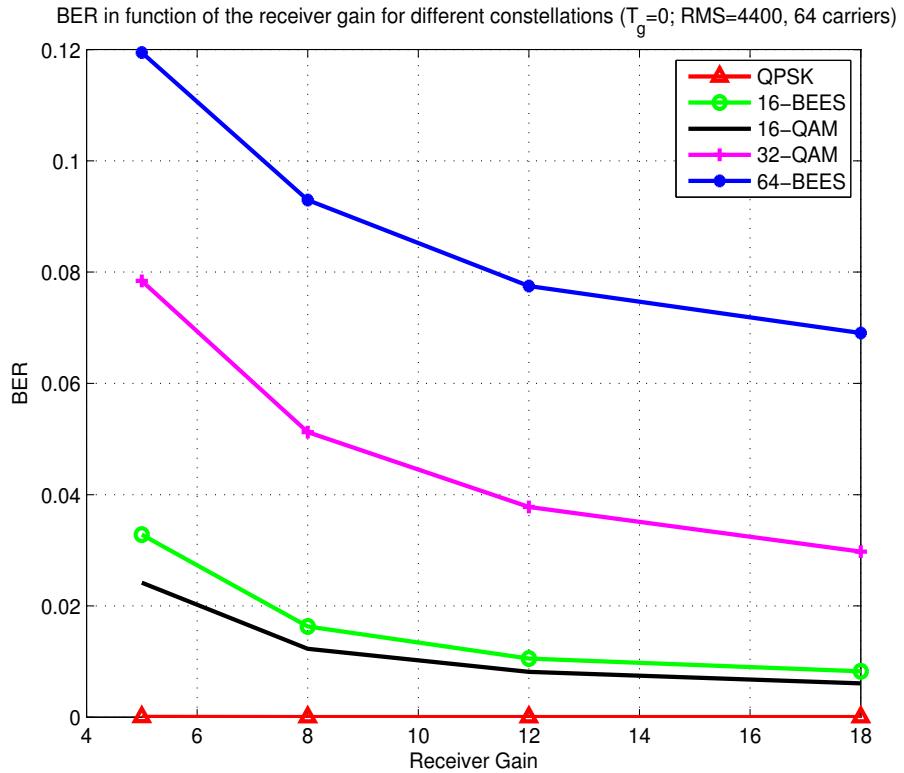


FIGURE 6.49: BER in function of the receiver gain for different constellations, 64 carriers

The increase of the BER for 64 carriers can be explained by several reasons. First, the spacing between subcarriers becomes smaller, therefore, signal becomes more subjected to ICI and is more sensitive to correction of the frequency offset. Second, with the increase of the number of carriers, the PAPR also increases which can result in additional distortions.

To demonstrate how the BER is affected by the choice of the RMS value of the transmitted sequence, it was changed in the next test and the obtained curves are plotted in figure 6.50.

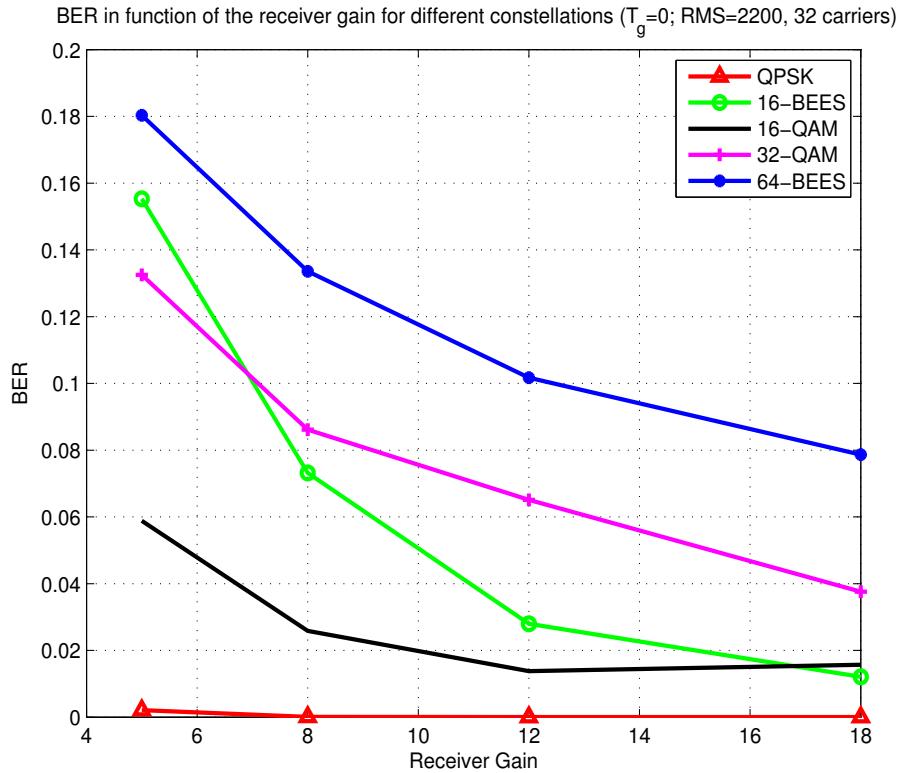


FIGURE 6.50: BER in function of the receiver gain for different constellations, 32 carriers, RMS=2200

One can see, that the choice of the RMS is important. First, there appears the dependence on the receiver gain for QPSK when RMS is set to 2200. The optimal value for the receiver gain becomes 8 dB or higher. Just by changing the RMS from 4400 to 2200, the overall number of errors increases. For example, comparing 64-QAM for gain 5 of the receiver, we got $BER = 0.12$ for $RMS = 4400$ and for $RMS = 2200$ it was $BER = 0.18$.

To check if this was true for greater number of carriers, we tested the system with $N = 64$ carriers. The results are plotted in 6.51.

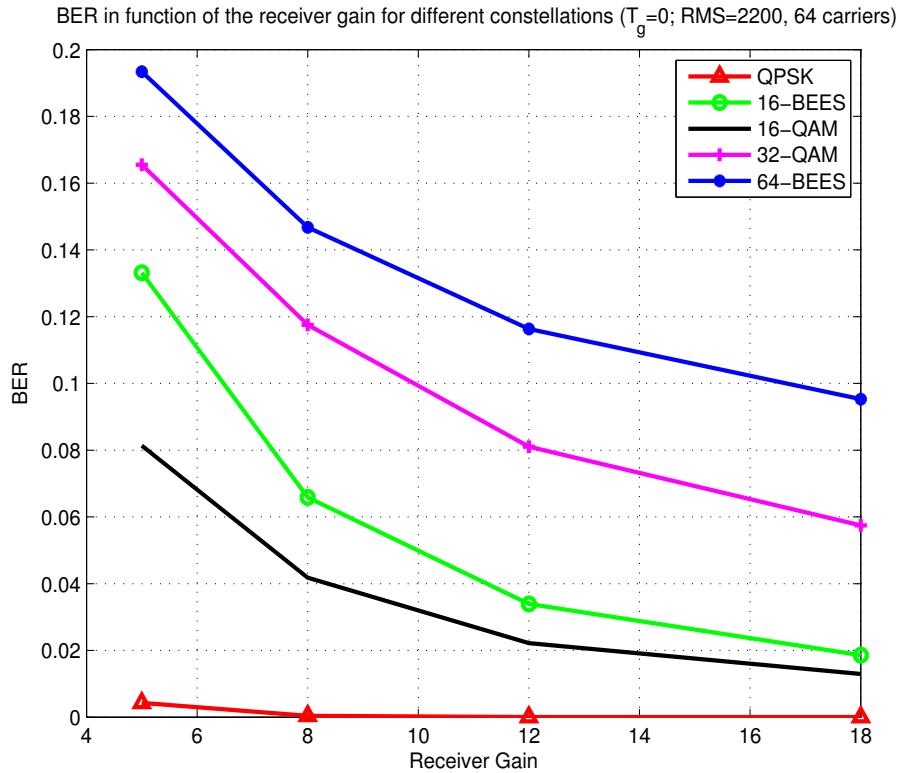


FIGURE 6.51: BER in function of the receiver gain for different constellations, 64 carriers, RMS=2200

To summarize the results, BER for 16-QAM is plotted for different RMS and number of carriers in dependence of the gain. In figure 6.52 are shown the obtained curves. The worst performance is registered for the case when RMS is 2200 and number of carriers is 64. The best performance is when RMS=4400 and number of carriers is 64. It is interesting to notice, that the increase of the number of carriers from 32 to 64 does not influence the BER significantly if the RMS is set to 4400, especially for larger values of receiver gain. It follows, that RMS value is a sensitive parameter that should be chosen with care in order to obtain expected results.

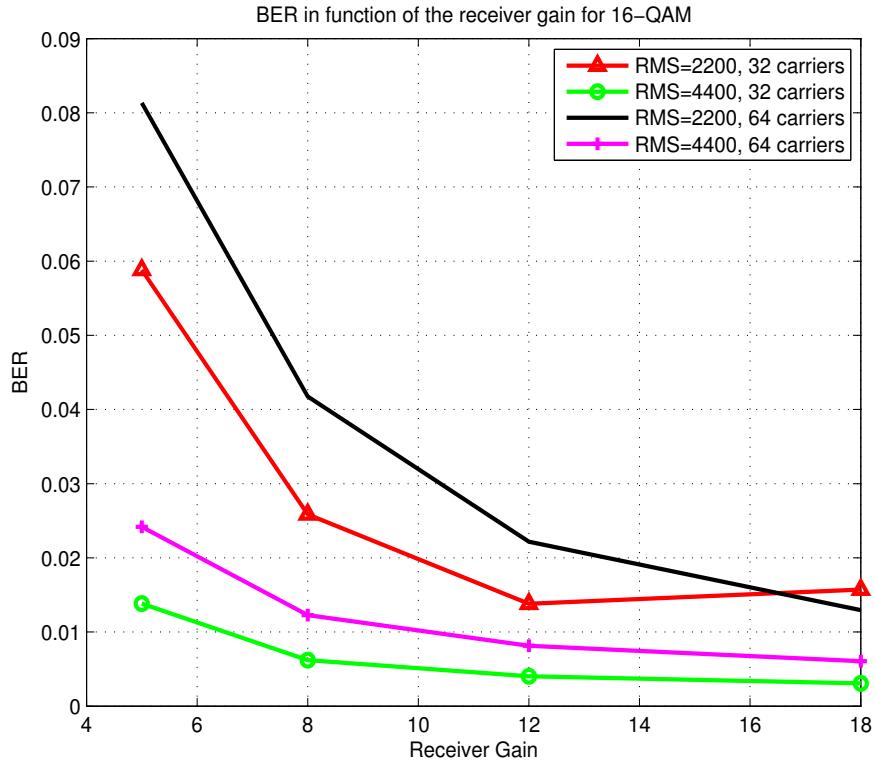


FIGURE 6.52: BER in function of the receiver gain for 16-QAM

6.6 Summary

This chapter presented System 2. It was about developing more advanced technologies compared to system 1, such as OFDM, source and channel coding, improved equalization techniques and energy efficient constellations. As in any communication system, there was a trade off in the optimization of the designed system. Several parameters can be adjusted according to the final objective. Most of the time of the group work was spent for developing, implementation, and testing and optimization of this system. During all this stages, we faced several challenges.

The maximum achieved data rate with real data in 5GHz channel and MATLAB processing for 128 carriers and turbo coding with the rate of 0.8 consisted 72 Mbits/s with the FER of 10%. In 60GHz channel we achieved the data rate of 54 Mbits/s with FER of 10^{-2} . Without coding, the data rate in 60GHz channel reached 22 Mbits/s with 0 BER and QPSK modulation and 45 Mbits/s with 16-QAM and $\text{BER} \approx 10^{-3}$.

The main issue was that 60 GHz equipment was provided when the system was already implemented. On the other hand, it might be seen as an advantage, as while working

with simpler channel at 5 GHz we were able to establish some weak points and some mistakes in the code and correct them. We also worked a lot with real data in Matlab which was crucial for understanding and improving the system, as well as comparing the C++ implementation and simulated results. Aspects of implementation of this system in C++ follow in the next chapter.

Chapter 7

OFDM system in C++

This section presents the results of the OFDM system 2 implementation done in C++, for transmission in both 5 GHz and 60 GHz. This OFDM system is an advanced communications system, designed particularly to satisfy the real-time requirements given in chapter 1. It was, of course, implemented in C++ in the USRP, in a real-time communications environment.

The URSP 2 was used at the receiver and a USRP N210 at the transmitter, for 5 GHz transmissions. Later, the provided daughterboards for 60 GHz were used. Therefore, the initial system was designed for 5 GHz communication and later adapted for transmissions at 60 GHz. Moreover, the skeleton of the USRP interface was based on the first implementation described in chapter 5.

7.1 Modifications from Theoretical Second System

This design is a contemporary design of a wireless communications system and it can be found, for instance, in LTE or WiFi. In the real-time C++ implementation, we implemented the main features that allow an OFDM based system to work properly in a wireless medium; even under bad conditions such as the ones present in the 60 GHz communications channel. This system was previously simulated in Matlab, to guarantee correct functionality and to investigate the theoretical performance. In the next subsections, the main changes done in C++, from the theoretical model, are described.

7.1.1 Transmitter

The transmitted sequence used in the real-time implemented OFDM system consists of guard bits, that are sent first, followed by the complex samples of the training sequence, and then complex samples of the OFDM data. In the end of the sequence, guard bits are introduced.

In the implementation, only 4 QAM and 16 QAM modulation were used, since the noise was too powerful to increase the constellation size further. To use higher order constellations, one could implement some channel coding scheme. Higher constellations are possible to use, based on IT++ implementation of modulation, demodulation and hard detection.

Apart from the source and channel coding, the entire C++ implementation corresponds to the block diagram of the second system, as presented in chapter 6. The team tried to implement channel coding, but did not succeed due to the limited amount of time available for the project. The IT++ library was used in C++ to help with the integration of the coding functions, but obtained results were unsuitable for a real-time implementation, therefore no results of this effort are presented here.

7.1.2 Receiver

The receiver was also implemented as shown in the diagram in chapter 6, without the last three processing blocks, which correspond to the detection algorithm. ML hard decoding (simple decision regions) was used in its place, to recover the initial binary data. The detection algorithm given in 6, was not implemented due to time limitations of the project and the priorities that were set for the hardware implementation team. Furthermore, in a real-time implementation, a channel code decoder, with a rate of 0.8 (or similar), would have severely degraded the time performance of the system.

7.2 Real-time Implementation

This section describes solutions used in the actual implementation of the system. After that, results obtained in a C++ to C++ transmission, at 5 GHz and at 60 GHz, are shown. Note that all the MATLAB plots shown here are depicted using the data from binary files written during the real-time processing in C++.

Summarizing, the chapter first presents how the C++ system was implemented, in order to produce a working system, and then the methods used to achieve the advanced requirements of the project. The most difficult problem was the decoding time of the receiver and a really simple and optimized version is explained at the end of this chapter, as well as the prototype for the demonstration.

7.2.1 Transmission

In order to create the transmitted sequence in C++, binary data is loaded from a file that is to be sent. The data is randomly generated. At the beginning of the C++ implementation, there were thoughts about having a transmission with, for instance, a text or an image file, but for the demonstration it was decided that we should design a system capable of transmitting a predefined image file with a size that would satisfy the advanced requirements of the project. The overall focus was on having a system working in real-time, which could receive and decode binary data, but for the demonstration it was considered more interesting and informative to display a transmitted image, especially for people unfamiliar with the project details.

To transmit an image in C++, we use the XOR operation on the binary data of the file with a previously generated pseudo noise sequence. At the receiver, the data can be recovered through an inverting operation and by calling Octave, we can display it.

A predefined complex training sequence optimized for this system is loaded from a file. This training sequence is upsampled by 4 when transmitting at 5 GHz and by 2 at 60 GHz, due to the increased carrier frequency offset that makes it more difficult to track in higher frequencies.

The remaining parameters of the system were also loaded from a file, such as the used pilot and the distribution of carriers, both for the data and for the pilots.

The main differences between the transmission at 5 GHz and 60 GHz were the OFDM system parameters. While at 5 GHz, one could use 89 carriers out of 128, at 60 GHz the phase noise allowed only the use of 17 out of 32. The length of the cyclic prefix, CP, and suffix, CS, also differed. At 5 GHz, CP and CS of 18 and 1 were used. At 60 GHz these lengths were set to 10 and 4 respectively.

In the system, around 200 OFDM symbols were sent at 5 GHz and 800 OFDM symbols were sent at 60 GHz, in order not to exceed the sending capacity of the USRP (even though this number, set for the test, may have been unnecessarily low).

Some problems arose with sending a large buffer directly to the USRP (passing the data to the transmitter by calling the send function of the *tx_streamer*). It seems that after transmitting some data, the USRP just sends noise and the last data that was supposed to be transmitted, is lost. In order to be able to transmit a file size of 10 kB, a simple method based on packets was used. This method consisted of an encoded predefined sequence. This way, the system was able to transmit file of greater size.

In the next stage, the signal processing on the received data was performed, as described in the theory section. It is worth mentioning that, in real-time transmission, a problem with the phase of the first OFDM symbols always appeared. This could be caused by some kind of transient regime of the receiver USRP or it could be because of the spectral leakage of the training sequence, since we have a frequency selective channel. Our solution was to insert a guard interval of low power random data before the OFDM data to avoid the problem.

After the sequence for the transmission was generated, it had to be passed to the USRP interface code, described in the implementation of the first system 5. This code was used for sending the data. Both real part and the spectrum of the sequence sent from one of the experiments can be seen in figure 7.1. As one can observe, this is a typical OFDM spectrum, as expected.

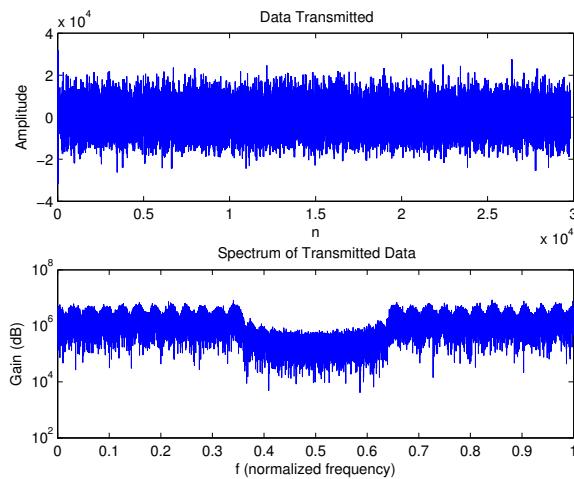


FIGURE 7.1: Example of transmitted Sequence at 60 GHz. Real part and Spectrum of data.

In terms of implementation, a more thorough description of the algorithm implemented is given next.

At first the binary data, the pilot pattern, pilot data and training sequence are loaded. Then we map the binary data by using the M-QAM function in IT++. After the data has been mapped, according to chosen constellation, the OFDM symbols have to be created. The OFDM symbols were created in a double for-loop and by using the IFFT-function in IT++, as stated in the algorithm code 1.

Algorithm 1 Function of Transmitter

```

for 0 to Number of OFDM symbols do

    for 0 to Number of Carriers do
        if Current Element Index Equal to Pilot Pattern Index then
            Element Set to Pilot Data
        else if Current Element Index Equal to then
            Element Set to Mapped Data
        else
            Element Set to Zero
    end if
    end for
    Take IFFT
    Adding Cyclic Prefix and Sufix
end for
15: Adding Training Sequence at the Front
    Adding Guard Bits at the Front and Back

```

7.2.2 Reception

Similarly to the first system implementation, the detection of the transmission was based on the power, detected at the receiver short buffer that the USRP outputs, which constantly checks the channel looking for a transmission. This was a primitive method which gave acceptable results in the beginning of the project. However, since it may cause some problems, a proper threshold had to be set, which was to be above the channel noise power. This was especially important at 60 GHz, where burst errors were detected, most likely caused by the hardware. The orientation of the boards also affected this threshold, thus forcing a modification of the threshold value before each transmission. The group studied another approach called Schmidl-Cox [Schmidl and Cox, 1997], described in chapter 6, for packet detection and frequency offset estimation. The method was not implemented in C++, but could have been a good approach in order to avoid the power detection defects referred to above. Nevertheless, there were some doubts, whether USRP/Computer overflow problem would appear. This was actually one of the main problems of the receiver part. Even though threading was implemented, overflow

could appear in the USRP, when doing heavy computation. For that reason the first method was chosen, since we were sure that it would be working properly.

After detecting the transmission, the program saved all the data in an array, to process it further. Next, another thread, which takes care of the processing, is called. In order to accomplish this, the thread scheme described in chapter 5 in the first system, is used.

7.2.3 Received Sequence

In figure 7.2 the channel effect on the received sequence can be observed. However, it is more interesting to observe its spectrum, where one can see that the LO leakage is situated at the DC component, since this subcarrier of the OFDM system is not used.

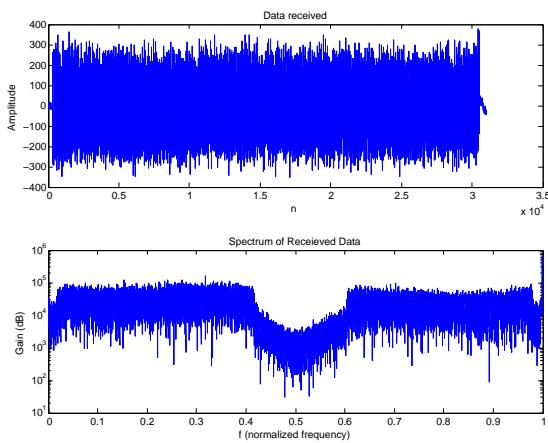


FIGURE 7.2: Example of received Sequence at 60 GHz. Real part and Spectrum of received data.

The sequence is processed in serial by a thread launched in the main program, as stated above. The main functions of the implementation were tested with the harness methodology, to ensure that all processing components were working correctly (available on Git), and the entire code was tested in a sort of harness methodology as well. At each step of the processing stages, a file with data was written and then a Matlab script read the C++ data, while in parallel the same data was processed by Matlab, in debug mode. This allowed us to check, step by step, the differences between the two implementations and track errors of the C++ implementation.

Important to note is that, in the beginning, we thought about doing the processing of the received data short buffer by short buffer, and to use different threads for each parallel part of the processing code. In the end, this was not implemented due to programming complexity and the time it would have consumed compared to the benefits we would have gained.

7.2.4 Receiver Implementation on C++

In this section, the implementation of the receiver function that processes the received data and outputs the resulting binary data will be briefly described, so that the reader can understand and be able to reproduce a similar implementation. Three prototype functions will be downloaded from the project website.

The receiver function had, as input, the complex sequence received by the USRP, represented as a *short* (real and imaginary). After the processing of this data, a binary array was filled with the decoded data and a file with that data was written. To evaluate the results and the performance of the algorithms, at each step of the decoding, data files were saved. Then at the end of the processing, Matlab/Octave was called to check and plot the results. Of course, this increased the decoding time and was not used when time performance of the program was evaluated (in order to achieve the requirements). It is also worth to mention that all data is processed in *double*, since we use the library IT++ and it is defined for this type. This is also a factor that increases the decoding time of the program but it was necessary in order to use some very useful functions.

In order to have access to all of the necessary information needed to process the received data, the *.dat* files are used. These files are shared between the transmitter and the receiver. Specifically, we read the used training sequence, the pilot pattern, the data pilot and the used pilots to equalize the channel effect.

7.2.4.1 Catching Training Sequence and Coarse Channel Estimation

The first function of the program (*synccatchchannel*) was used to catch the training sequence and give a coarse estimation of the channel, based on an FIR Wiener filter. It was implemented with the IT++ function *xcorr*. We also implemented our own correlation function, but the IT++ library was needed for the computation of the coarse channel estimation Wiener FIR filter. After trying to minimize the decoding time of the receiver, we actually realized that it was faster than the function implemented for the first model. In this function we computed the crosscorrelation of a specified small number of the first points of the received data with the upsampled training sequence. We assumed that a training sequence was present at this range between the first element of the data and the last point we chose. This is very important for reducing the time required by this function, since computing the crosscorrelation, with all the received complex samples, kills the decoding time of the program. We also used this peak to

get a gain and a phase estimation. Then, we got Σ_{YY} , Σ_{Yx} based on *cmat* and *cvec* containers from IT++ to compute the FIR channel estimation. At last, the variance of the estimation was also computed. However, some of those parameters were not used in the rest of the code but were only used in the first theoretical implementation.

7.2.4.2 Getting OFDM Data

When the start of the sequence was known, the actual OFDM data was processed. The cyclic prefixes and suffixes were removed by a function called *CPremove*. Based on the cyclic prefix, the phase shift of each OFDM data was corrected and the DFT was taken on each OFDM data. The DFT was performed, again, using the FFT available in the IT++ library. The most important part was that the OFDM symbols were stored in a *Queue* container. Using such a container was effective in terms of computation and also made the processing of each OFDM symbol easy. Further along in the program, it was only necessary to remove the OFDM symbol from the Queue, process it and put it back in the same or in a new Queue:

```

for 0 to Number of OFDM symbols received do
    Remove cyclic prefix and suffix
    Correct Frequency Offset of Data
    Take FFT
    Push OFDM Symbol to Queue dataOFDM (of complex vectors)
end for

```

7.2.4.3 Data Separation and Channel Information

Once the OFDM symbols were obtained, it was necessary to process them in order to decode the data. Here we took advantage of the Queue system described above. Knowing the Pilot and Data distribution within the OFDM subcarriers, it was possible to take the data from the initial Queue and put it on 3 other new Queues. One for the subcarriers containing data and two for the gain and phase of the channel at the pilot position (corrected by knowing the pilot as described in theory). Pseudocode found below:

7.2.4.4 Kalman Filtering of the Channel

The channel gain and phase, for each pilot carrier, was filtered using two Kalman filters defined as described in chapter 6. The Kalman filter was applied to the PilotGain Queue and PilotPhase Queue, at each time, in parallel. Using the Queue container

```

for 0 to Number of OFDM symbols do
    Pop Data from Queue dataOFDM
    for 0 to Number of Subcarriers do
        if data then Store data in cvec
        end if
        if Pilot then
            Get Channel Gain and Store in vec
            Get Channel Phase and Store in vec
        end if
    end for
    Push OFDM Symbol to Queue constOFDM
    Push Channel Gain to Queue PilotGain
    Push Channel Phase to Queue PilotPhase
end for

```

and processing in this way makes it really easy to manage the data, once we apply the Kalman filter, in each OFDM symbol. After this, we put the data back in another Queue called filtPilot (Gain or Phase). The pseudo code of the implemented function follows:

```

for 0 to Number of OFDM symbols do
    Pop Data from Queue (Gain or Phase)
    for 0 to Number of Pilots Used do
        Do one Kalman iteration on Corresponding Pilot Channel Gain or Phase
        Store data in vec
        Keep data for next Kalman iteration
    end for
    Push Channel filtered Gain or Phase to new Queue (filtPilot)
end for

```

7.2.4.5 Channel Interpolation

After filtering the measurements at the pilot positions, linear interpolation was performed, independently for the phase and amplitude, between the pilots, to get the channel estimation for all subcarriers. To do this, we extract the data from the Queues constOFDM, we performed the above mentioned linear interpolation and then put the data back into the respective Queue. Two major issues were addressed in the implementation.

First, the normalized frequencies are represented from 0 to 1. However, the physical channel ranges from -0.5 to 0.5. This implies an interpolation from 0.5 to 0.49, interpolating in the positive direction on the interval [0,1], where exit in the higher end of the interval equates entry into the lower end of the interval. The 0.49 and 0.5 frequencies were not linked together. This issue was solved by using a modulo operation.

The second issue was the periodicity of the phase. It was important to link the two pilots by the shortest path. For example, if two consecutive symbols' phases are respectively in 0.1 and -0.1, they will be linked through 0. On the other hand, two consecutive phases on 3 and -3 will be linked through π without crossing zero. We assume that the phase changes by less than 2π between two consecutive pilots.

The procedure is illustrated in the figure 7.3.

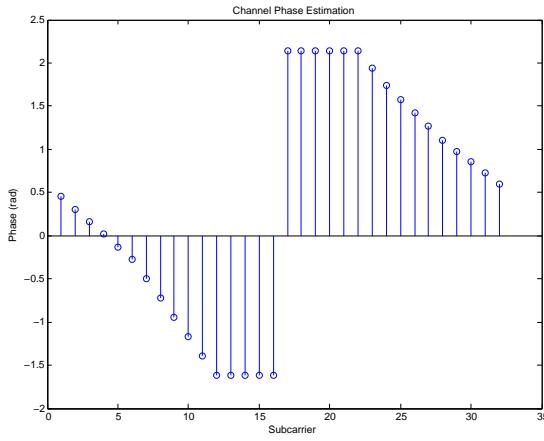


FIGURE 7.3: Example of Phase Estimation.

The pseudo code for the phase is presented in algorithm 2. The function for the amplitude is similar.

Algorithm 2 Phase interpolation

```

Get the highest frequency pilot
Pre-compute frequency difference
for All pilots in the high frequencies do
    Compute the phase difference. The result lie between -pi and pi
    for All intermediate subcarriers do
        Do the linear interpolation.
    end for
    for All subcarriers between the highest frequency pilot and 0.5 do
        Keep the phase of the highest frequency pilot
    end for
    for All subcarriers between 0.5 and the lowest frequency pilot do
        Keep the phase of the lowest frequency pilot
    end for
end for
for All pilots in the low frequencies and first pilot after DC do
    Compute the phase difference. The result lie between -pi and pi
    for All intermediate subcarriers do
        Do the linear interpolation.
    end for
end for

```

7.2.4.6 Channel Correction and Data Detection

Now we had everything necessary to detect the data. In this part of the program we simply have to push the OFDM carriers with data from the Queue (constOFDM). In the same manner the channel gain and the phase Queue was pushed. This procedure was followed by an equalization of the complex samples. Everything was then written to a complex vector and the decoding was done using the IT++ function/class *demodulate.bits* (an alternative to this is using our own function for QPSK). After this, a file with the binary data was written and the results were observed using MATLAB or Octave. These tools allowed us to get a better understanding of the data collected.

7.2.5 Results and Parameters

The received signal processing was done in several stages as described in the previous subsection. In this subsection we will state the parameters used and evaluate the results obtained during some processing stages, although they correspond to the same conclusions done in chapter 6.

7.2.5.1 USRP Parameters

The most important parameters for the USRP at 60 GHz are the antenna gains and the RMS value of the transmitted sequence. As shown in chapter 6, the best value for the transmitter gain was 0 dB and the best value for the RMS was 4400. Also, the best antenna gain to use at the receiver was 12 dB. These parameters influenced the performance of the transmission. Since the different gains could cause saturation of both amplifiers and the RMS value it was important not to introduce powerful quantization noise.

We could also have been using two attenuators of 10 dB at the transmitter I/Q signal intermediate frequency up-conversion to the carrier frequency. In this case the best value for the gain at the transmitter was around 3 dB.

Using these attenuators, in 60 GHz communication, did not force the alteration of any of the parameters from the Matlab simulation, except the RMS, which was changed to 6000, according to empiric tests. In C++ implementation we did not have the capability of easily finding these values, so the parameters were first found using the MATLAB script and then slightly adapted to C++ implementation.

7.2.5.2 Channel Estimation

The channel estimation at 5 GHz is depicted in figure 7.4. In the gain figure we can clearly see that the gain is not constant at all frequencies, but that it was approximately constant over time, making it a slowly varying frequency selective channel. Which had been predicted and which was analyzed in chapter 6. In the phase we can notice the carrier frequency offset that created a linearly varying phase.

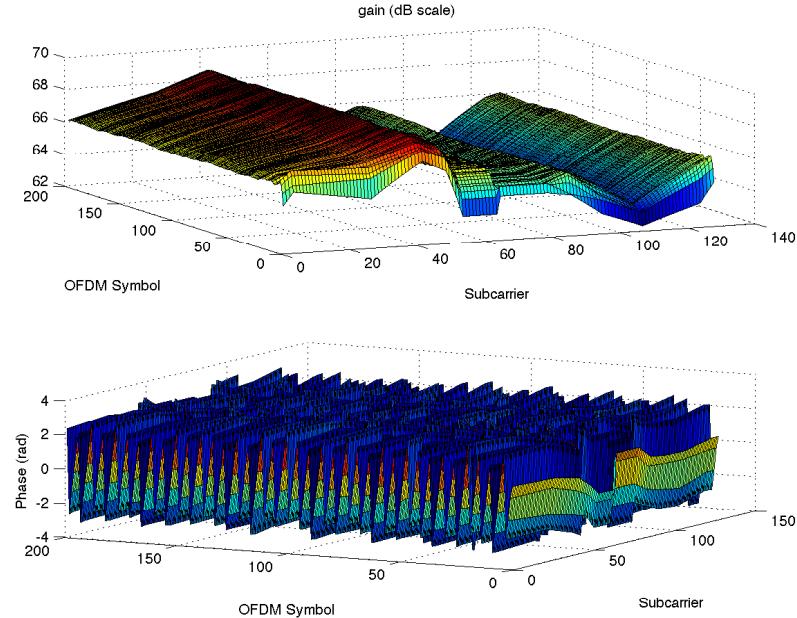


FIGURE 7.4: Channel estimation at 5 GHz for USRP 2 (Gain and Phase).

In figure 7.5 the 60 GHz channel estimation is shown. At this high frequency band the multipath propagation of the signal was low, making the estimation of the gain more constant in the frequency domain. However, owing to the hardware quality of the RF boards, the frequency offset was much larger, as can be seen in the same figure. This was one of the main reasons for decreasing the number of carriers used in the OFDM system at 60 GHz and also one of the reasons for decreasing the upsampling factor of the training sequence, so that we do not have problems of catching it.

7.2.5.3 Received Constellation

4-QAM and 16-QAM were chosen as the constellations to implemented in real-time. For both 5 GHz and 60 GHz we got zero BER for 4-QAM, but around 0.2% BER for 16-QAM, which corresponds more or less to the values obtained using Matlab. These tests were done on one transmission only, which means that getting a reliable value is

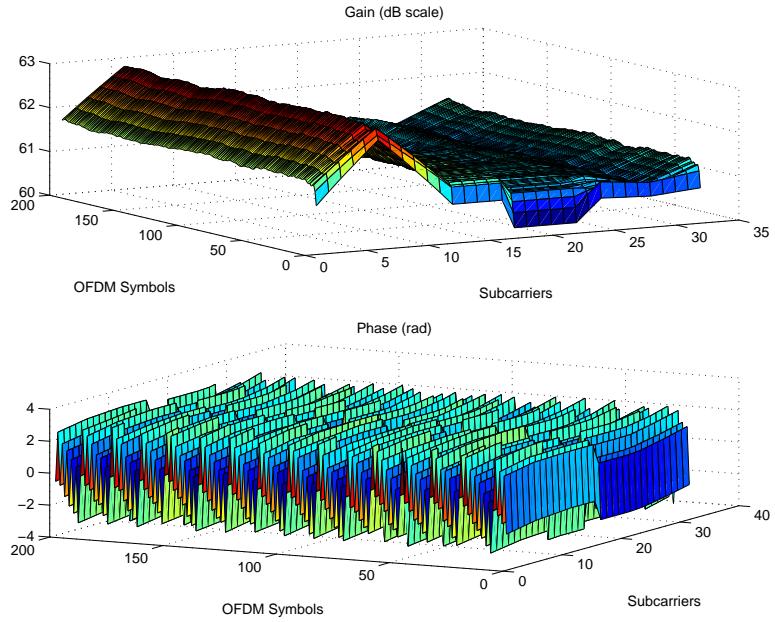


FIGURE 7.5: Channel estimation at 60 GHz (Gain and Phase)

difficult if the Matlab is not used.

For 5 GHz the achieved data rate for 4-QAM was 29.6 Mbps, and for 16-QAM, it was about 58.9 Mbps (depending on the number of errors produced). This number is slightly lower than the one attained in Matlab, since we needed to suppress 4 OFDM symbols in order to solve the problem of initial phase distortion stated earlier. It is interesting to plot the received constellation of one transmission using 4-QAM and then with 16-QAM (figure 7.6 and 7.7) to further compare the 5 GHz transmission against its 60 GHz equivalent.

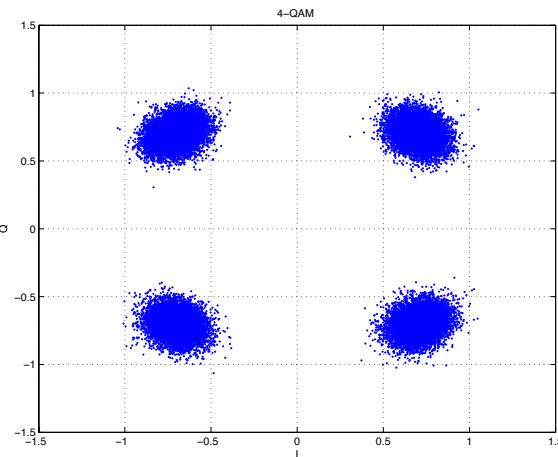


FIGURE 7.6: Received 4-QAM constellation at 5 GHz.

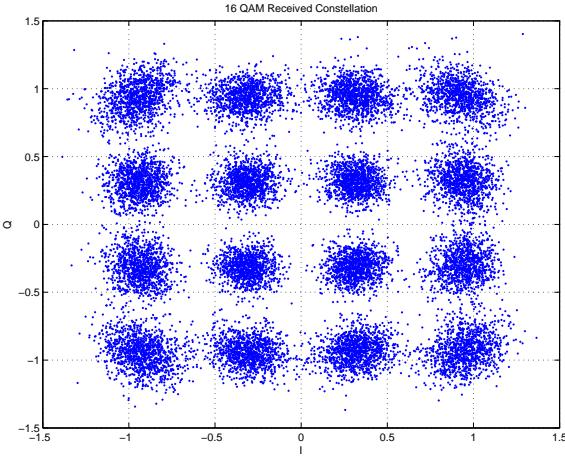


FIGURE 7.7: Received 16-QAM constellation at 5 GHz.

For 60 GHz the achieved rate for 4-QAM was around 18 Mbps and for 16-QAM was around 35.9 Mbps. Observe that these rates were lower compared to the above since the effective bandwidth was smaller and thus, the spectral efficiency was reduced. It is also interesting to plot both the 4-QAM and the 16-QAM constellation, where we can see the effects stated above, and compare it to the results obtained with Matlab transmission. These plots can be seen in figures 7.8 and 7.9.

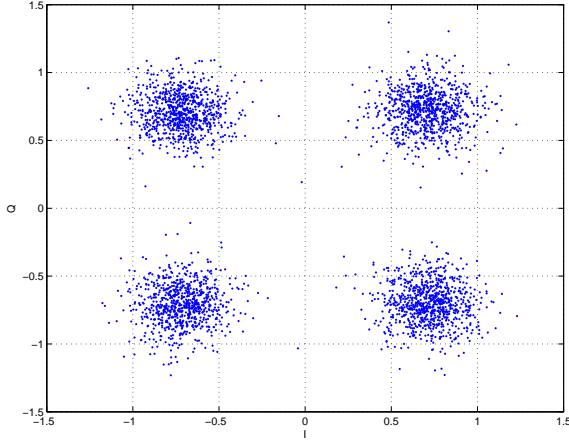


FIGURE 7.8: Received 4-QAM constellation at 60 GHz.

As thoroughly analyzed in section 6.3, we can see the severe effects of the 60 GHz channel, mainly ICI due to phase noise but also saturation of amplifiers and hardware noise.

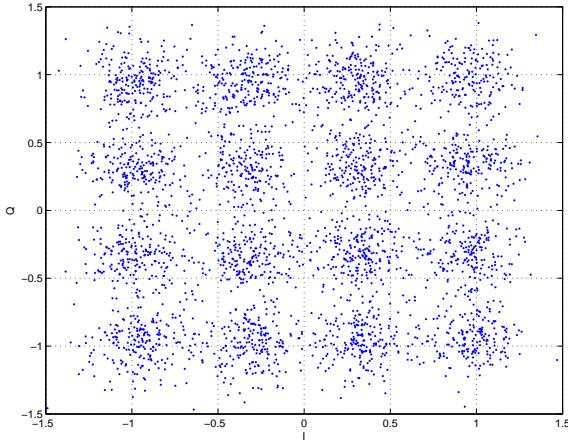


FIGURE 7.9: Received 16-QAM constellation at 60 GHz.

7.2.6 Implementation Requirements

The system implemented and described above does not fulfil the advanced requirements of the project defined at the beginning of the course. The Kalman filtering of the channel, that provided knowledge needed by the pilots, took too much time, even when it was implemented with the Queue system.

As described in section 6.5.1, the Kalman filter does not significantly improve the performance of the system which means that it can be suppressed for real-time transmission (although, sometimes we got interference in one OFDM symbol estimation since it seems the channel is deep fading when the system is not using the Kalman filter). If so, the receiver implementation can be simplified. In this section we describe the real-time implementation improved without any Kalman filtering or channel estimation at the beginning. A function that allows the transmission of files bigger than 10 kB is described, also.

The first modification was to suppress the coarse FIR Wiener channel estimation, based on a training sequence, used to initialize the Kalman filters. Packet-based transmission was introduced, which meant adapting the receiver to be able to decode more than one transmission from the same set of received data.

To detect the different packets we still introduced the training sequence in order to ensure ourselves that the packets are still decoded even if we get under- or overflows in one of the packets. Over- and underflow were rare, though. The objective was to allow

the transmitter to send several packets and consequently avoid any errors.

Without the Kalman filter, using one Queue was enough, since the processing of each OFDM symbol could be done one at a time. As described above, we took the FFT of the OFDM symbol, got the pilot and the data, interpolated the channel gain and phase for all subcarriers and finally we equalized the samples. After getting all samples we used hard detection and we got the transmitted binary file back.

If there was more than one packet to decode, the algorithm had to find the crosscorrelation of every training sequence presented in the received complex sequence. Observe that this was too computationally heavy. The problem was solved by using the peak of the previous training sequence and by knowing the size of the packet. Thus, we were able to assume that the training sequence that followed was inside a certain range. The procedure was simple and allowed the transmission of large file sizes. The processing is described in algorithm 3.

Algorithm 3 Improved Receiver

```

while Packets to Decode do
    Get Training Sequence within Short Sample Range
    Remove cyclic prefix and store data on Queue
    for 0 to Number of OFDM symbols of Packet do
        Pop OFDM symbol from Queue
        for 0 to number of subcarriers do
            if Pilot then
                Get Channel Gain and Phase and Store in vec
            end if
            if Data then
                Store data in cvec
            end if
        end for
        Pilot channel interpolation to all subcarrier
    end for
    Put data back to a bvec
end while
Hard Decoding of Samples
  
```

This makes it possible to see the time differences of the systems working with the Queue system with and without Kalman. In table 7.1, a file of 27.2 kbits, modulated with QPSK, is shown.

<i>Implementation Procedure</i>	<i>DecodingTime</i>
Queues with Kalman	53 ms
Queues without Kalman	18 ms
Improved	12 ms

TABLE 7.1: Different methods and decoding time.

As can be seen, the Kalman takes around 35 ms to decode which is not tolerable for a real-time implementation. The initial Queue system does not introduce much processing time, but we could still reduce the decoding time doing the process of each OFDM symbol once.

Chapter 8

DFT-precoded OFDM (System 3)

At 60 GHz the problems of PAPR and saturation of the amplifiers at the receiver and the transmitter became significant. One of the ways of combating these negative effects was through introducing DFT-precoding in the OFDM system. The implementation of DFT-precoding in system 2 was impossible, due to the allocation of the pilots. A completely new system was developed to incorporate the DFT-precoding block, a system where the pilots are located on the sides of the used spectrum and where the used spectrum is contained in a continuous range. This system, named system 3, is presented and analyzed below.

The third system was designed to take profit of our channel and hardware characteristics. First of all the hardware presents saturation, as previously stated. This problem can be addressed, and it's effects limited, through the use of PAPR reduction algorithm's. We investigate DFT precoded OFDM in section [8.1](#).

Another feature which was changed when moving from system 2 to system 3 was the channel estimation. After observing that the channel is actually slowly varying in time, it made sense to estimate it only once, at the beginning of the OFDM frame, considering it constant for the remainder of the frame. In this estimation, considerations on PHN are of utmost importance. A joint-estimation algorithm is presented in section [8.2](#). Assessment revealed that the CFO estimation did not provide sufficient precision. To solve this, we present some tracking methods based on two pilots, in section [8.4](#). Finally, some tests were performed in a real environment, though still in offline mode. The synchronization was once again assured by a time domain training sequence.

The block scheme of the DFT-precoded OFDM system is shown in figure [8.1](#).

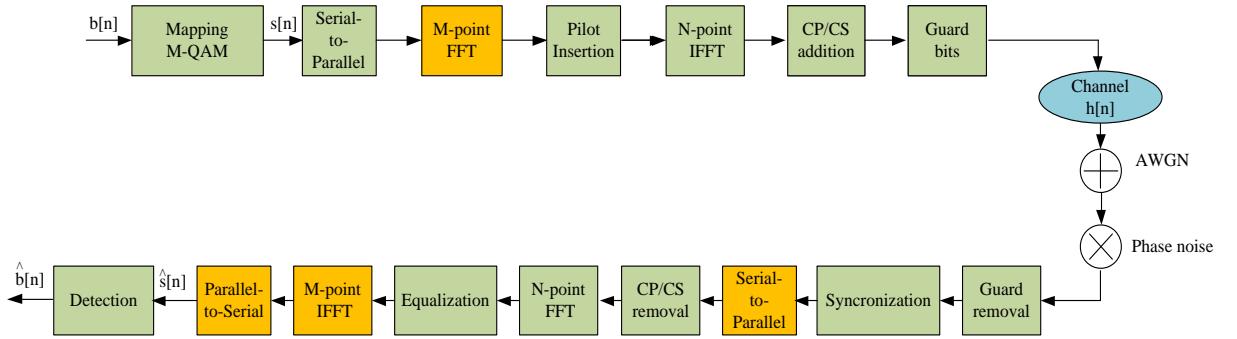


FIGURE 8.1: Block diagram of DFT-precoded OFDM system

By implementing a DFT-precoded OFDM system, we restricted ourselves to using two pilot channels, located on the channels just outside of the subcarriers used for transmitting data. Therefore, the tracking algorithm will rely exclusively on 2 pilots per each OFDM symbol.

Using techniques, described in [Liu and Chong, 2002] and [Lin et al., 2006], the following procedure has been proposed for channel estimation. The block scheme for channel estimation is shown in figure 8.2 and the global structure of a frame in figure 8.3.

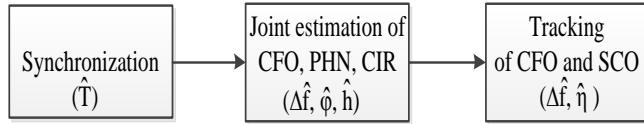


FIGURE 8.2: Block scheme for channel estimation

8.1 DFT precoded OFDM

DFT precoding is an efficient method used to reduce PAPR. It can be combined, or substituted, with additional processing methods which reduce PAPR. Other methods can include coding, companding, clipping and chirping.

The performance of DFT-precoding was simulated and evaluated in MATLAB and compared with the performance of not using any PAPR mitigating measures at all. To make the evaluation meaningful, saturation was faked by clipping the signal at a predetermined level.

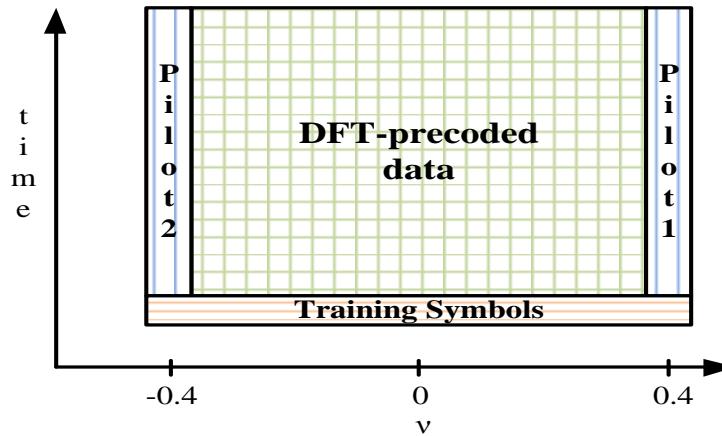
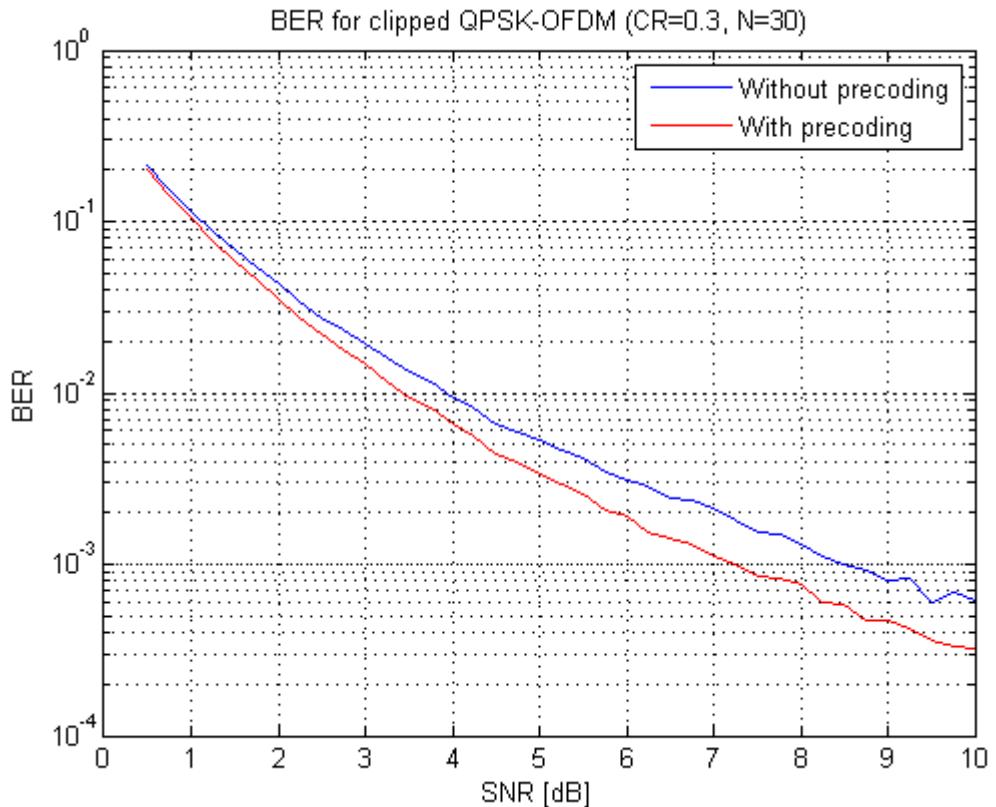
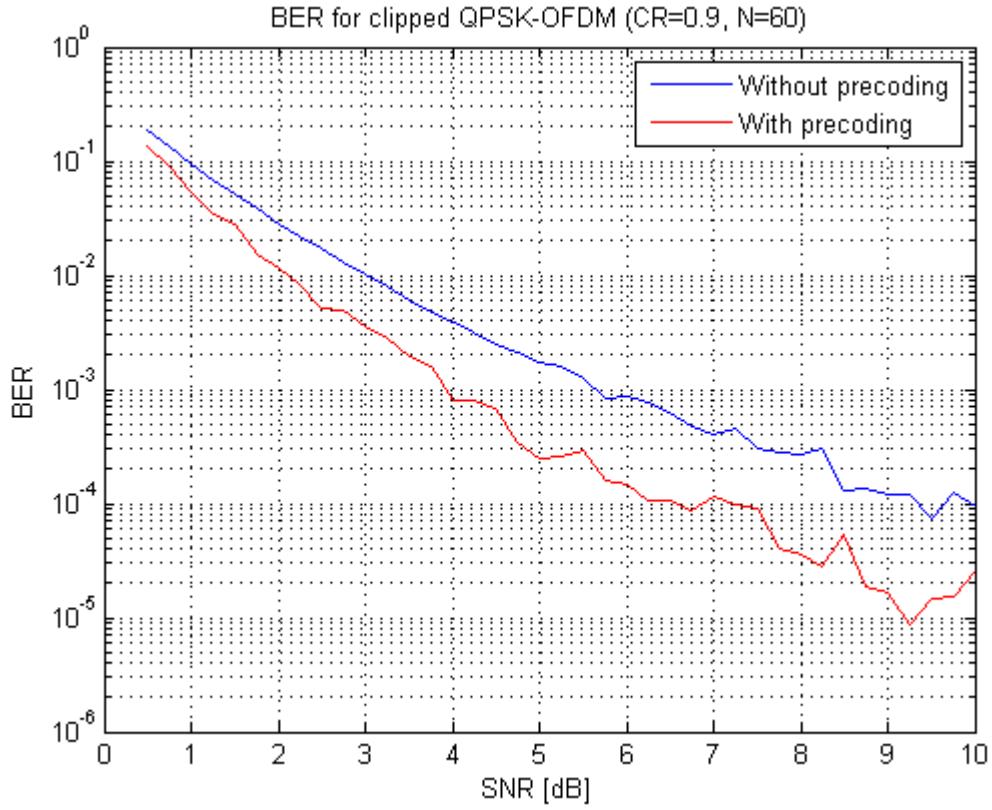


FIGURE 8.3: The structure of OFDM transmitted signal

FIGURE 8.4: DFT-precoded with $M = 30$ and clip rate 0.3

Simulations were done for an OFDM system of 128 carriers, where 66 subcarriers held QPSK modulated bits and the rest were left unused. Each simulation consisted of a transmission of 1,500,000 bits for each SNR-point used. The same signal was transmitted twice, once with DFT precoding and once without, for reference. The measured SNR's ranged from 0 to 5 or from 0 to 10. The channel only contributed with white

FIGURE 8.5: DFT-precoded with $M = 60$ and clip rate 0.9

noise, thus omitting phase noise, frequency offset and multipath components. The rationale for this was simply that the DFT precoding only changes the modulation of the data, not the pilots. They will function the same as if no precoding was used, meaning that we have no need for simulation of their performance.

As both the DFT precoded signal and the regular signal used as a reference were subject to clipping with the same CR, and both had the same mean power before clipping, both signals were left with the same PAPR after clipping (as the peaks are quite few, as opposed to the general signal, an unsignificant amount of energy is removed and both have the same peak value after the operation). Attempting a DFT of order 60, with a CR of 0.9 resulted in a PAPR of about 10 and a BER curve as 8.5. The precoded version of the signal always performs slightly better than the same signal without precoding. The drawback is increased computational time. Using a lower order DFT will decrease the computational load. Using the FFT instead of the DFT would also help, and would be natural in a final implementation. Tests with an order $M=30$ DFT and $CR=0.3$ can be seen in 8.4. The resulting PAPR is about 5.5 for both signals. As seen in the figure, there is an advantage in BER to the precoded signal.

8.2 Channel estimation

Once we managed to get time-synchronized, we aimed to estimate relevant parameters of our transmission: CFO, PHN and CIR, applying the algorithm broadly described in [Liu and Chong, 2002].

This method is based on joint maximum likelihood estimation of these parameters. The main advantage of this method is that each estimation takes other effects into account and tries to mitigate them. For that, it is important to get prior information about some of the parameters:

- L: length of the equivalent channel
- σ^2 : the power of the channel noise. The noise is assumed to be zero-mean Gaussian distributed.
- The second-order statistics of the phase noise. This is carried out by modeling our oscillator. There are two broadly accepted models for oscillators:
 - Free run oscillators: they can be modeled as Wiener processes
 - Phased-Locked VCO: it can be approximately modeled by zero-mean colored Gaussian processes, i.e. filtered Gaussian noise

The last model is used for generating phase noise and from there, we obtain the covariance matrix [Lin et al., 2006].

$$\Phi_{i,j} = \left(\frac{\pi \theta_{rms}}{180} \right)^2 e^{-\frac{-2\pi B|i-j|}{f_s}} \quad (8.1)$$

where B is the $3dB$ bandwidth of the transfer function, measured for given USRP's, θ_{rms} was the phase noise standard deviation, and f_s was used as the sampling rate.

Our estimation is based on transmitting one OFDM training symbol, which has two identical halves in time-domain. That could be achieved by filling the even frequency bins with a known training sequence and the odd with zeros. We use a maximum length training sequence, easily generated in both transmitter and receiver by Linear Feedback Shift Register (LFSR), which is QPSK-modulated. As demonstrated in 6.1.8 this symbol can also be used for packet detection.

The received OFDM symbol for each time index can be written as follows:

$$r_n = \frac{1}{\sqrt{(N/2)}} e^{j(\theta_n + 2\pi \epsilon n / N)} \sum_{k=0}^{N/2-1} h_k d_k e^{j4\pi nk / N} + w_n \quad (8.2)$$

Where N is the number of carriers, θ_n are the different realizations of the phase noise process for each time index, h_k is the channel realization for the frequency bin $2k$ and d_k is the known sample located at frequency bin $2k$.

The algorithm basically consist of computing the joint likelihood function for CFO, PHN and CIR and optimizing it. By using repeated training sequence in time domain, the CFO becomes independent of our joint function. The method performs as follows:

- Estimate the CFO. The maximum estimate we can get is $|\epsilon_{max}| = \frac{1}{N_c}$
- Compute the joint likelihood function for PHN and CIR. We optimize it assuming first PHN to be known and then estimating the current PHN values. The performance of these estimates will be shown in the simulations results.
- Finally, as CFO and CIR are assumed constant throughout the transmission, we correct them before initiating the tracking algorithm.

The estimated values can be found as follows:

$$\begin{aligned}\hat{\epsilon} &= \frac{1}{\pi} \angle \mathbf{r}_1^H (\mathbf{R}_1 \Phi_\Delta \mathbf{R}_1^H + 4\sigma^2 \mathbf{I})^{-1} \mathbf{r}_2 \\ \hat{\theta} &= [\Re(\hat{\mathbf{E}} \mathbf{A} \hat{\mathbf{E}}) + 4\sigma^2 \rho^2 \Phi^{-1}]^{-1} \Im(\hat{\mathbf{E}} \mathbf{A} \hat{\mathbf{E}}) \mathbf{1} \\ \hat{\mathbf{g}} &= (4\rho^2)^{-1} \mathbf{W}^H \mathbf{D}^H \mathbf{F}_c \hat{\mathbf{P}}^H \hat{\mathbf{E}}^H \mathbf{r}\end{aligned}\quad (8.3)$$

Where:

- $\mathbf{r}_1 = [r_0, r_1, \dots, r_{N/2-1}]$ the first half of the time-domain signal
- $\mathbf{r}_2 = [r_{N/2}, r_{N/2+1}, \dots, r_{N-1}]$ is the second half of the time-domain signal, and $\mathbf{r}_1 = \mathbf{r}_2$
- $\mathbf{r} = [r_0, r_1, \dots, r_{N-1}]$
- $\Phi = \begin{bmatrix} \Phi_{N/2} & \Upsilon \\ \Upsilon^T & \Phi_{N/2} \end{bmatrix}$
- $\Phi_\Delta = 2\Phi_{N/2} - \Upsilon - \Upsilon^T$ and is the covariance matrix of the difference between the realizations of phase noise in both halves of the received signal $\theta_1 - \theta_2$
- $\hat{\mathbf{E}} = \text{diag}(1, e^{j2\pi\hat{\epsilon}}, \dots, e^{j2\pi\hat{\epsilon}(N-1)})$
- $\mathbf{R} = \text{diag}(r_0, r_1, \dots, r_{N-1})$
- \mathbf{F} is the N/2 DFT matrix normalized.

- \mathbf{W} is the partition of the DFT matrix with L columns and \mathbf{V} is the partition of the DFT matrix with $N/2 - L$ columns so that $\mathbf{F} = [W|V]$
- $\mathbf{F}_c = [\mathbf{F} \ \mathbf{F}]$
- $\mathbf{D} = diag(d_0, d_1, \dots, d_{N/2-1})$
- $\hat{\mathbf{P}} = diag(e^{j\hat{\phi}_0}, e^{j\hat{\phi}_1}, \dots, e^{j\hat{\phi}_{N-1}})$
- $\mathbf{A} = \mathbf{R}^H \begin{bmatrix} \mathbf{F}^H \mathbf{D} & 0 \\ 0 & \mathbf{F}^H \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} \mathbf{V} \mathbf{V}^H & -\mathbf{W} \mathbf{W}^H \\ -\mathbf{W} \mathbf{W}^H & \mathbf{I} \mathbf{V} \mathbf{V}^H \end{bmatrix} \begin{bmatrix} \mathbf{F}^H \mathbf{D} & 0 \\ 0 & \mathbf{F}^H \mathbf{D} \end{bmatrix}^H \mathbf{R}$

8.3 Simulation results

In the simulation we use the following parameters:

- Number of carriers: 64
- We use a 2-tap channel, close to what we have in our real system ($L = 2$), and normalized such that the most powerful tap has 0 dB gain.
- Phase noise modeled as a coloured zero-mean Gaussian process, with a standard deviation of $\phi_{rms} = 9$ degrees and 3 dB bandwidth $W = 1MHz$
- Sampling frequency $f_s = 25$ MHz.
- QPSK modulation for the training symbol.

8.3.1 Distribution of CFO estimations

Our CFO estimation was highly dependent on the phase noise realizations. To describe the performance of our CFO estimator, we have plotted the distribution of our estimation.

For a frequency offset of $\epsilon = 0.01$ we get the following results ¹

¹Note that we express CFO as the frequency normalized by the number of carriers $\epsilon = \frac{\Delta f}{N_c}$

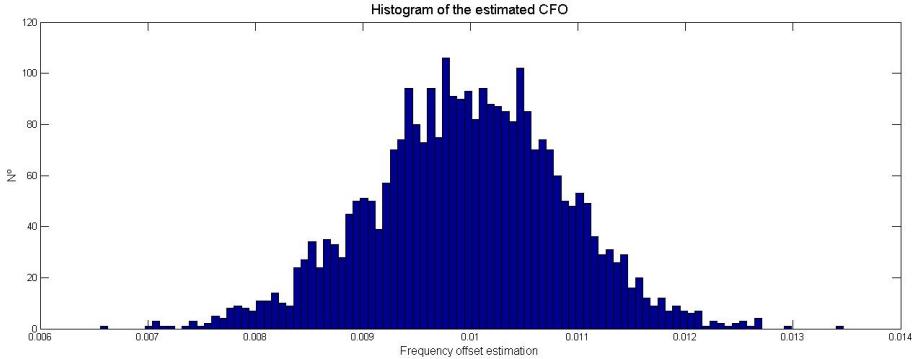


FIGURE 8.6: PHN Estimation

Our estimation follows a Gaussian bell shape with mean value $E[\hat{\epsilon}] = \epsilon$ and standard deviation $\sigma_\epsilon = 9 \cdot 10^{-4}$. Thus, our estimation is unbiased and the variance is directly the Mean Squared Error (MSE). The maximum CFO we could estimate was $|\epsilon_{max}| = \frac{1}{N_c}$ which, according to the angular domain, means half of the circle in both directions, positive and negative (positive represents the case where the receiver frequency is slightly higher than the transmitter frequency, while negative means the opposite). Furthermore, if the tracking algorithm was properly initialized, the residual error in the CFO estimation can be corrected.

8.3.2 Estimation of PHN

Our results for the PHN depend directly on the CFO estimation, done previously. We can actually see two different results: in figure 8.7, we have a rather good estimation of the frequency offset ($\epsilon_{error} \sim 10e - 4$) whereas in figure 8.8, we got some higher order errors ($\epsilon_{error} \sim 10e - 3$), which lead us to a non-constant phase shift. We were able to identify and correct this error with the pilot-based tracking .²

²Note that these numbers are based on $N_c = 32$, if one increases the number of carriers, the effect of non-constant common-phase shift will appear with smaller errors.

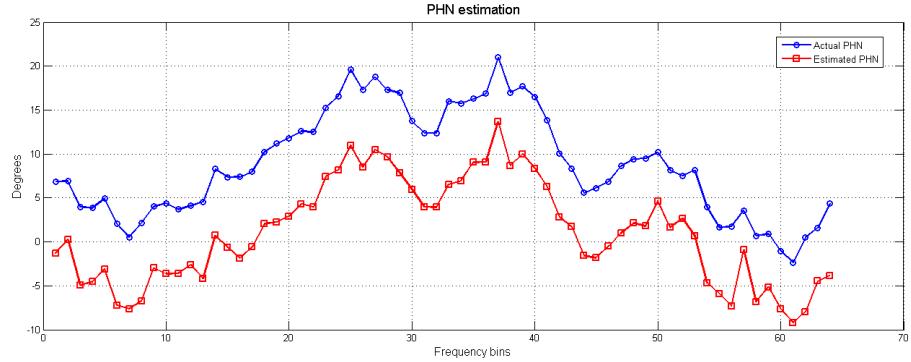


FIGURE 8.7: Phase Noise estimation (perfect CFO estimation)

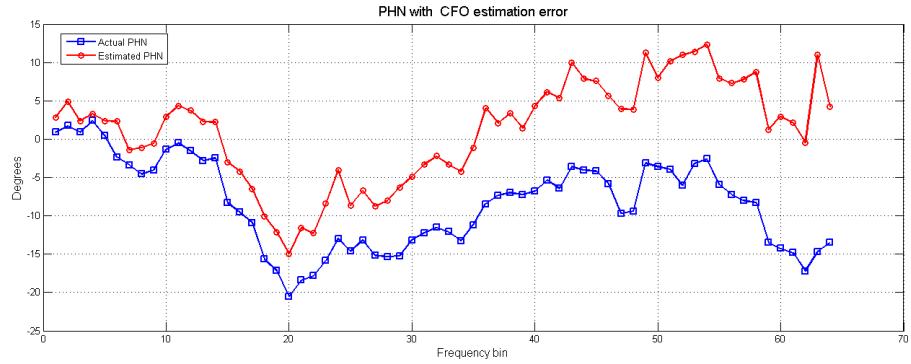


FIGURE 8.8: Phase Noise estimation (non-perfect CFO estimation)

This estimation was mainly done for removing phase noise effect on the channel estimation. To get a more reliable channel estimation, we should have corrected this CPE term before estimating the channel. This is easy to do in simulations, but has not been dealt with in tests with real data.

8.3.3 Estimation of the channel

Using reliable estimation for the other two parameters, we can see how the MSE of our channel estimation varies with the SNR of our channel, in figure 8.9.

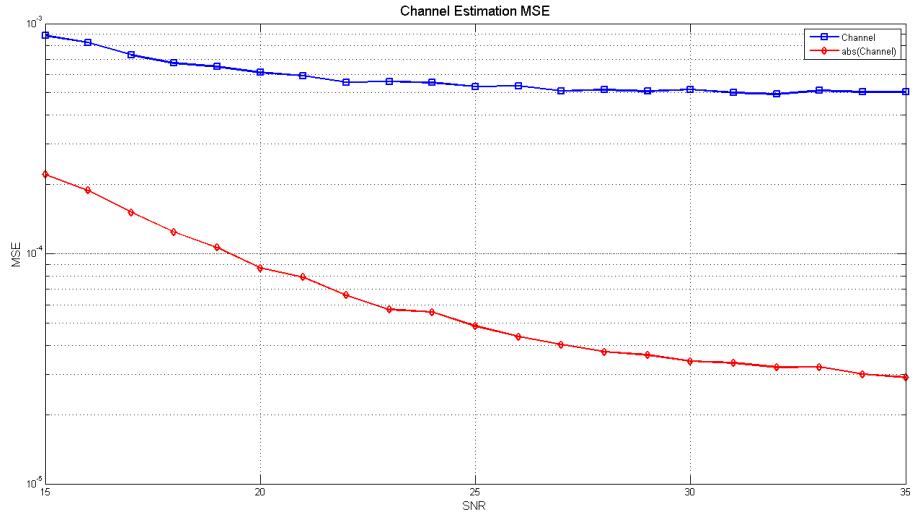


FIGURE 8.9: MSE of the channel

Our channel estimation does not improve its performance with better SNR, while the absolute value actually does. In fact, one can see that the absolute value estimation depends on the SNR, as we expected, but that the overall channel estimation does not. Furthermore, the absolute value estimation was not dominant in our estimation for rather high SNR values ($SNR \geq 15dB$). From this holds that the phase estimation of our channel does not depend on SNR and that it is the dominating term, when comparing it with the absolute value. This holds because the phase noise is much more powerful than the channel AWGN noise in the analyzed range. This can introduce some additional phase rotation in the signal and can actually be a source of errors.

8.4 Tracking of the residual CFO and CPE

The first estimation of the CFO obtained in section 8.2 is not precise enough to allow hard decision. Figure 8.10 shows the phase of the pilot after correction using the joint estimation. In addition to the noise, one can see a residual CFO. Further tracking of the phase is necessary, over the OFDM symbols. The parameters used in the simulations are the following (unless something else is stated, these are the used parameters used in all simulations):

Number of carriers	32
Number of used carriers	24
Constellation	QPSK
SNR	40 dB
$\Delta\nu$	$1.01 \cdot 10^{-2}$
Channel	$\{1, 0.2\}$
Residual CFO	$1.5 \cdot 10^{-4}$

Two different tracking algorithms are presented in this section. The first one is based on [Liu and Chong, 2002] and the second one is the Kalman filter already used in the second system, thoroughly presented in section 6.2.2. After a theoretical description of the problem, we will compare them in the presence and absence of phase noise (PHN), assuming perfect channel correction on account of the estimation in 8.2.

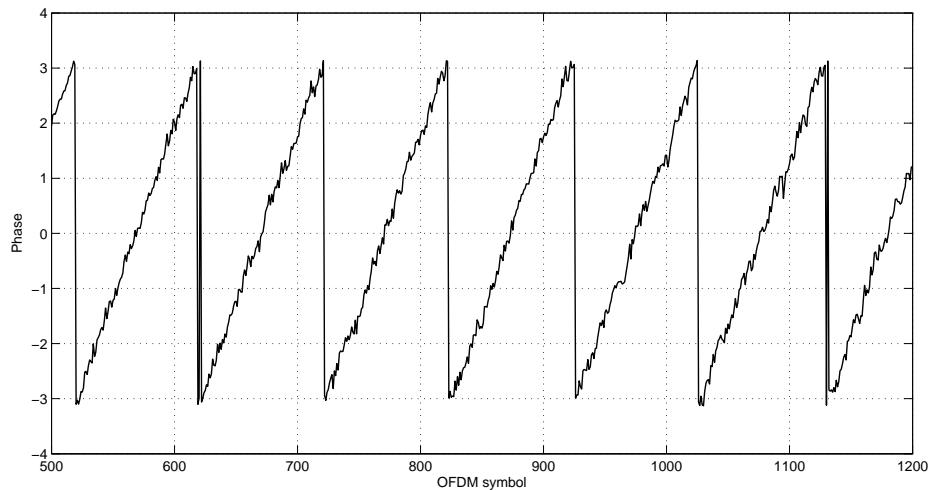


FIGURE 8.10: Phase of a pilot over the OFDM symbols.

8.4.1 Theoretical description

This description is inspired by [Liu and Chong, 2002] and does not include PHN. The interested reader is referred to that same paper, to learn more about the ideas used.

In an OFDM system using an N-point IFFT, i.e. N subcarriers, the transmitted complex baseband signal for the l^{th} OFDM symbol and N_{cp} samples as cyclic prefix is:

$$s(t) = \frac{1}{N} \sum_{k=0}^{N-1} a_{l,k} e^{j2\pi k \frac{t-(N_{cp}+lN_s)T}{NT}} \quad N_s = N + N_{cp} \quad (8.4)$$

where $a_{l,k}$ is the QPSK symbol and N_s is the length of the complex baseband signal. The receiver uses a carrier frequency f' and a T' sampling clock. We define,

$$\Delta f = f - f' \quad (8.5)$$

$$\eta = \frac{T' - T}{T} \quad (8.6)$$

Therefore, the received sample, if the channel is AWGN, can be expressed as:

$$r_l(n) = e^{j2\pi\Delta f t_n} \frac{1}{N} \sum_{k=0}^{N-1} a_{l,k} e^{j2\pi k \frac{t_n - (N_{cp} + lN_s)T}{NT}} + \nu_l(n) \quad (8.7)$$

$$= e^{j2\pi\Delta f(lN_s + N_{cp})(1+\eta)T} e^{j2\pi\Delta f n(1+\eta)T} \times \quad (8.8)$$

$$\frac{1}{N} \sum_{k=0}^{N-1} a_{l,k} e^{\frac{j2\pi k n}{N}(1+\eta)} e^{\frac{j2\pi k}{N}(lN_s + N_{cp})\eta} + \nu_l(n) \quad (8.9)$$

After demodulation by FFT, considering Δf and η small compared to the Gaussian noise, and considering fading, the received data symbol can be represented as:

$$z_{l,k} = e^{j2\pi\Delta f(lN_s + N_{cp})(1+\eta)T} e^{\frac{j2\pi k}{N}(lN_s + N_{cp})\eta} H_l(k) a_{l,k} + n_l(k) \quad (8.10)$$

The rotated phase can now be represented as:

$$\varphi_l(k) = \underbrace{2\pi\Delta f(lN_s + N_{cp})(1 + \eta)}_{CFO} \underbrace{T}_{small} + \underbrace{\frac{2\pi k}{N}(lN_s + N_{cp})\eta}_{SCO} + \underbrace{\phi_l^H(k)}_{channel} \quad (8.11)$$

Considering the channel constant over more than one OFDM symbol allows the derivation of a different estimation algorithm. The two-dimensional linear least square estimator (LLSE) presented in [Liu and Chong, 2002] is given by:

$$\hat{\Delta f} = \frac{\sum_{i=1}^L \sum_{s=1}^M (l_i N_s + N_{cp})(\varphi_{l,k_s} - \phi(k_s))}{2\pi M T \sum_{i=1}^L (l_i N_s + N_{cp})^2} \quad (8.12)$$

$$\hat{\eta} = \frac{\sum_{i=1}^L \sum_{s=1}^M (l_i N_s + N_{cp}) k_s (\varphi_{l,k_s} - \phi(k_s))}{2\pi/N \sum_{s=1}^M k_s^2 \sum_{i=1}^L (l_i N_s + N_{cp})^2} \quad (8.13)$$

8.4.2 Comparison of the two algorithms

The performance of the two algorithms is studied in the presence of phase noise, AWGN, DFT-precoding, perfect synchronization and perfect channel estimation.

First of all, one can see in figure 8.11 that the phase of the pilot after correction is centered around 0 and of similar variance for the two algorithms. One can conclude that both of them should work correctly.

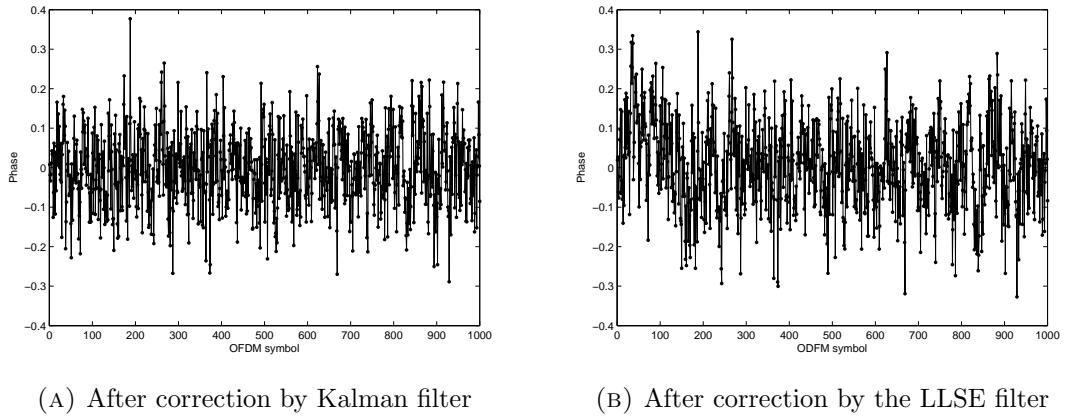


FIGURE 8.11: Phase of the pilot after correction

Looking at the BER for the two algorithms for different CFO, one can see in figure 8.12 that the Kalman filter (dashed line) gives better results for all CFO's ($\Delta\nu$). For smaller offsets, it can be a matter of several dB's in performance. As the offset increases, this discrepancy goes towards infinity.

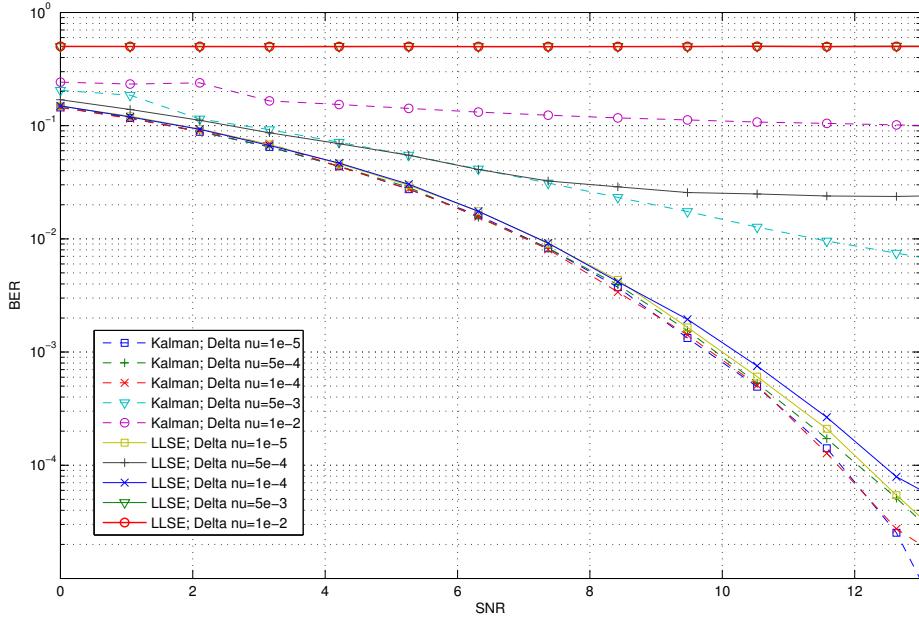


FIGURE 8.12: Comparison of the BER between LLSE tracking and Kalman filtering for different CFO

The two constellations are depicted in figure 8.13. One can see that the Kalman filter is less noisy in amplitude than the LLSE.

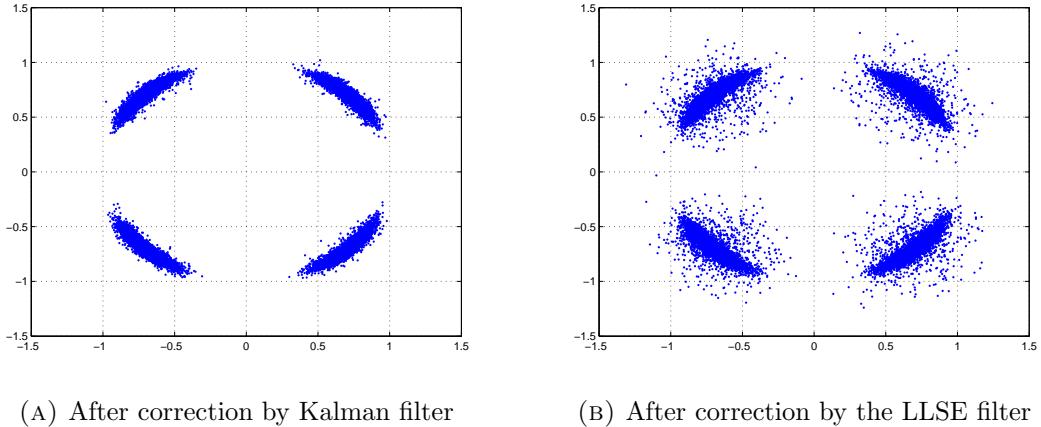


FIGURE 8.13: Simulated received constellation

8.4.3 Performances of the Kalman filter

The Kalman filter is used to track the phase changes in the two pilot subbands. The question is, how to combine them. We first thought about just averaging them, to reduce

the variance of the estimate. Although it sounds reasonable, there will be a few cases, where the estimated phase of each pilot could cancel out due to the fact that we are working in modulo between $[-\pi, \pi]$, and thus, give an inaccurate estimate, rotated by π . Thus, we may average taking this into account this fact computing the estimate. Finally we applied the following correction:

$$\hat{\phi} = \frac{\phi_{pilot_1} - \phi_{pilot_2} \bmod 2\pi + 2\phi_{pilot_2}}{2} \quad (8.14)$$

One can study the impact of the SNR on the BER in presence or absence of phase noise, with or without DFT-precoding. Once again, the channel estimation is considered perfect. $\Delta\nu$ is set 10^{-3} . One can see the penalty given by the phase noise (plain line), and the slightly better performances of the DFT precoding.

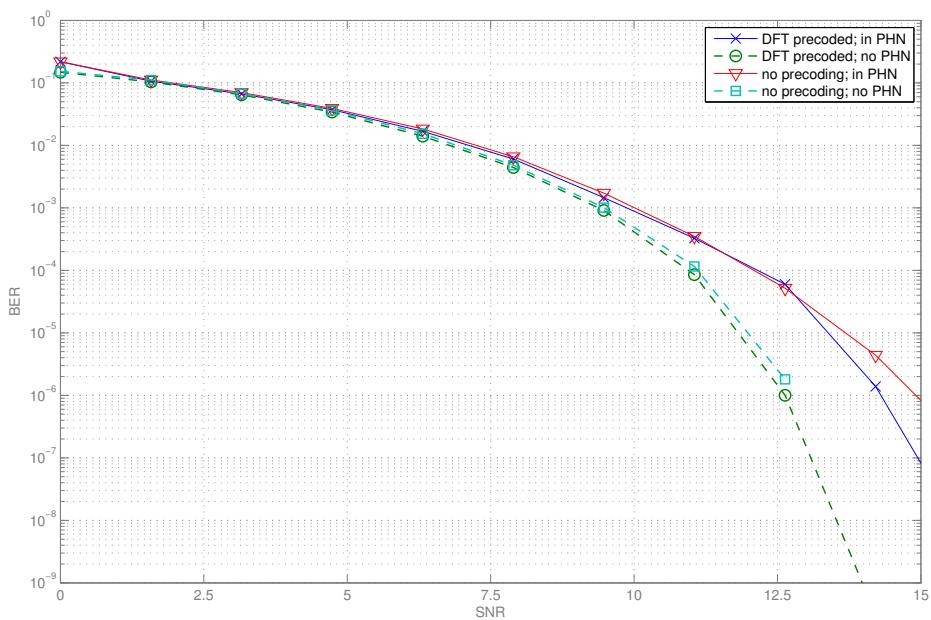


FIGURE 8.14: BER in Kalman tracking in presence of not of PHN, with or without DFT precoding

Furthermore, it's useful to see how it behaves for different residual frequency offsets (assuming perfect channel estimation). It can also give us some insights about the general system, since it will set the maximum error allowed in the estimation of the CFO, done in the previous module.

$$\epsilon = \hat{\epsilon} + \epsilon_{error}$$

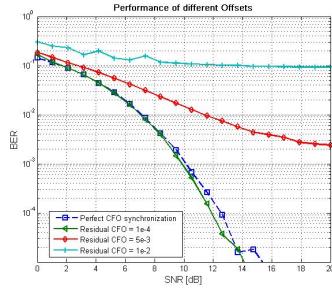


FIGURE 8.15: Performance analysis

From figure 8.15 one can see that our tracking algorithm can easily resolve residual frequency offsets of order $\epsilon_{error} \sim 1 \cdot 10^{-4}$. It is actually better than the perfect synchronization scheme where we didn't apply the tracking and thus couldn't deal with the CPE. For much higher frequencies, our tracking algorithm converges to a non-zero BER as SNR increases, but can still give us good performance in $\epsilon_{error} \sim 1 \cdot 10^{-3}$, if we use channel coding. For higher order residuals, the phase evolved too fast and so, we were unable to track it.

8.5 Simulation results of the system 3

In this section we simulate the results of the overall system performance based on the techniques described before. The basic system parameters used in our simulations are:

- Number of carriers: 64
- We use a 2-tap channel, close to what we have in our real system ($L=2$) and normalized such as the most powerful tap has 0 dB gain.
- Phase Noise modeled as coloured zero-mean Gaussian process, with standard deviation $\theta_{rms} = 9$ degrees and 3 dB bandwidth $W = 100kHz$
- Sampling frequency $f_s = 25$ MHz.
- 16-QAM modulation for data, and QPSK modulation for the training symbol.

8.5.1 Bit Error Rate evolution with SNR

To analyze the system performance, one usually refers to the BER evolution with respect to SNR, seen in figure 8.16. This gives us the system behavior and limitations in different environments.

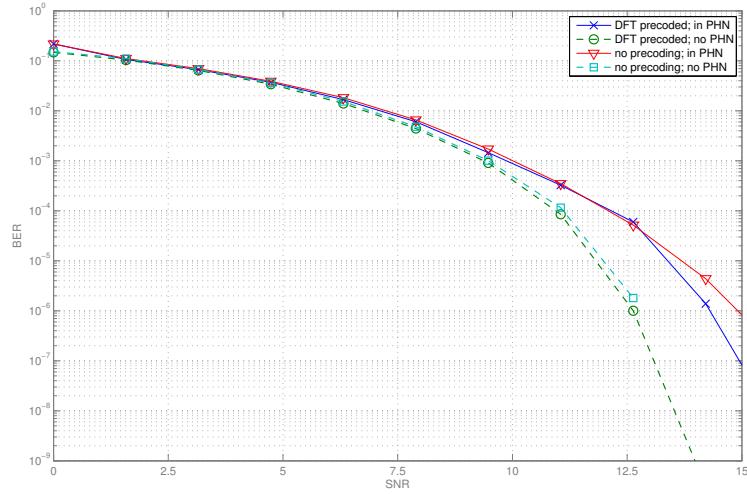


FIGURE 8.16: Performance analysis

The main conclusion we get from figure 8.16 is that our system will converge to a non-zero BER. Another way of putting it: we will never reach zero BER as $SNR \rightarrow \infty$ ³. This result comes from the fact that the phase noise is always constant and dominates over AWGN noise when we are in high-SNR environment.

Using DFT-precoded in simulations cannot give us any improvement regarding BER performance, since we do not model amplifier saturation. Thus, we will carry out the comparison between second and third model through simulation.

8.5.2 Performance analysis for different CFO

The effect of CFO increases with the carrier frequency. The oscillators become more unstable and the offset is much greater, in absolute terms. From this the need for proper CFO estimation arises. The impact of not correcting the offset can be seen in figure 8.18, for different CFO values. Besides, one can see in figure 8.17 that the system behavior becomes independent of the frequency offset within the operative region. As a reminder, the maximum frequency offset that our system can deal with is

$$-\frac{1}{N_c} \leq \epsilon \leq \frac{1}{N_c}$$

³SNR referring to the relation between signal and AWGN, without taking into account the phase noise added from the oscillators.

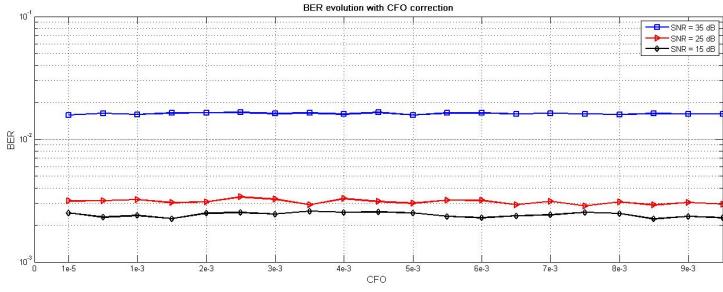
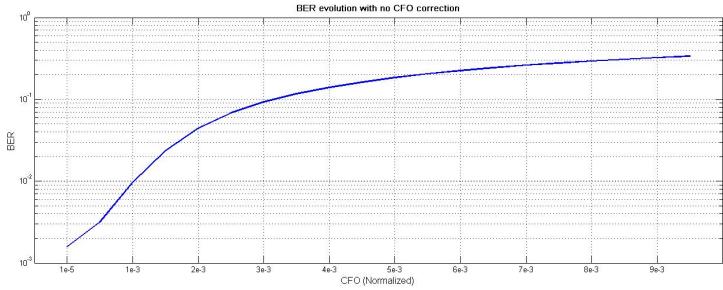


FIGURE 8.17: Performance analysis

FIGURE 8.18: BER evolution without CFO correction ($SNR = 30dB$)

8.5.3 Received constellation

The received constellation can also give us some insights about the performance of the system. One can see the impact of each block by plotting the received constellation with or without using some blocks. We found it interesting to compare the following blocks:

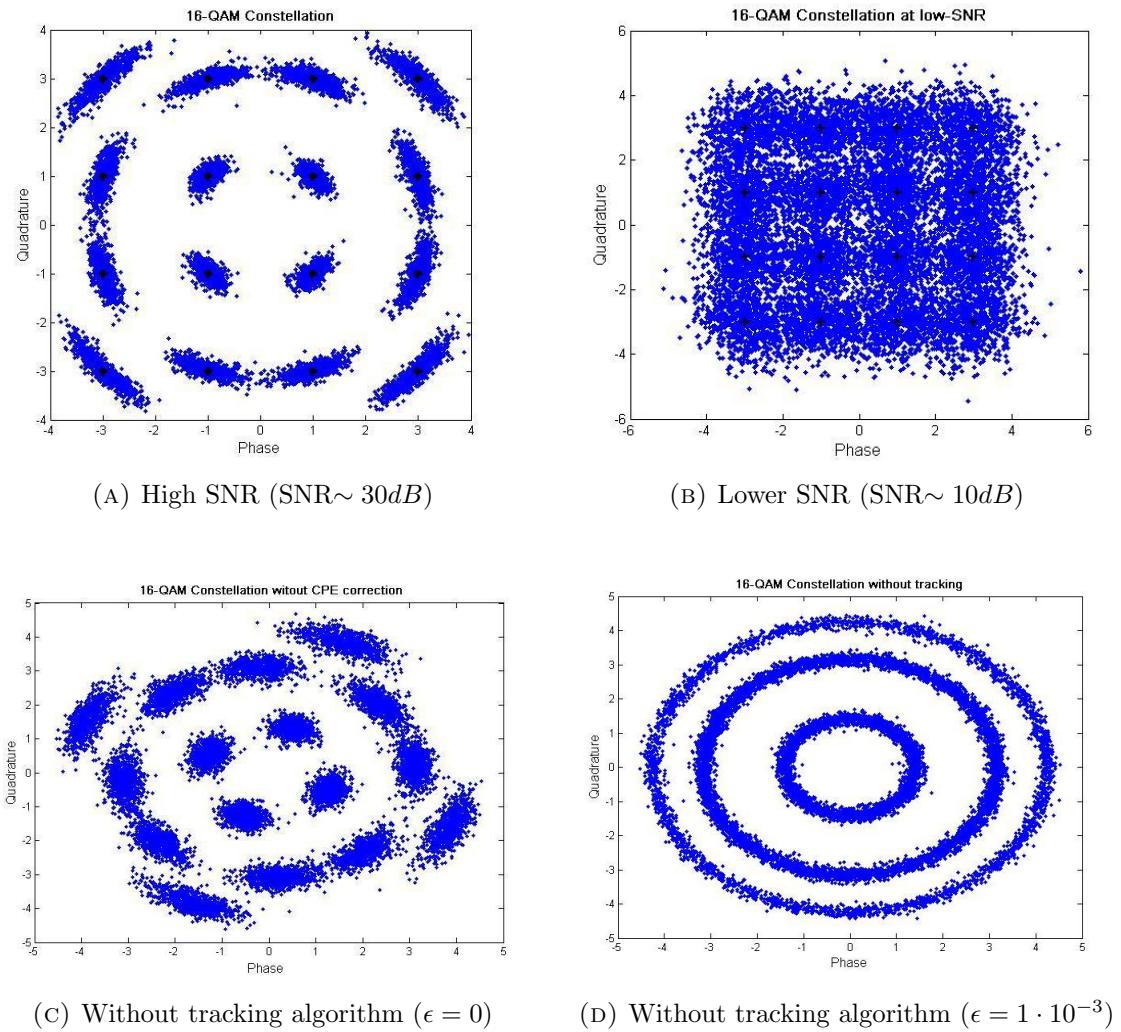


FIGURE 8.19: Received Constellations for different receiver configurations.

The results with high SNR, seen in figure 8.19a, show that in this case, phase noise dominates over additive Gaussian noise. This explains why the BER converges to a non-zero value as $SNR \rightarrow \infty$, as we are not able to correct ICI caused by phase noise. As long as phase noise dominates over AWGN, we will have the same performance in terms of BER, independent of SNR, in high-SNR regime. In low-SNR regime, seen in figure 8.19b, AWGN will dominate and thus BER will increase, confirming what was stated in section 8.5.

The impact of the estimation and tracking algorithms on the received constellation can be seen in figures 8.19c and 8.19d. After $\hat{\epsilon}$ correction, our signal still has some residual ϵ_{res} , which leads to a slowly increasing phase rotation. The impact of not tracking it can be seen in figure 8.19d, where the noise in the phase does not allow us to decode properly, resulting in a $BER \sim 0.4$. Along with the residual ϵ_{res} we also have CPE due

to phase noise and channel phase estimation error. Both of them lead to some rotation in the constellation, as is shown in figure 8.19c.

8.6 Real transmission results

Testing the system on a real transmission revealed a BER of 0.23 for 24/32 subcarriers, QPSK, and DFT-precoding, compared 0.24 without DFT-precoding. The BER was more or less insensitive to the parameters of the joint estimation (except for the number of channel taps) as stated in the original paper.

Using a 16-QAM modulation raised the BER to 0.29. With this scheme, the transmission speed is close to 75 Mbps. The transmissions with 64 subcarriers showed a BER of 0.5.

The results were independent of the length of the cyclic suffix, when a cyclic prefix of at least 10 was chosen. The number of channel taps should be set to at least 3. The tests on how to best use both pilots, showed that averaging was more favorable than correcting the high frequencies (HF) independently with the HF pilot and the low frequencies (LF) with the LF pilot.

The estimated channel gain and phase are presented in figure 8.20. After channel correction, the gain of the pilots (ideally 0 dB) is presented in figure 8.21a. One can see an error of less than 2 dB. On the phase, seen in figure 8.21b, one can see the residual CFO which is to be corrected afterwards.

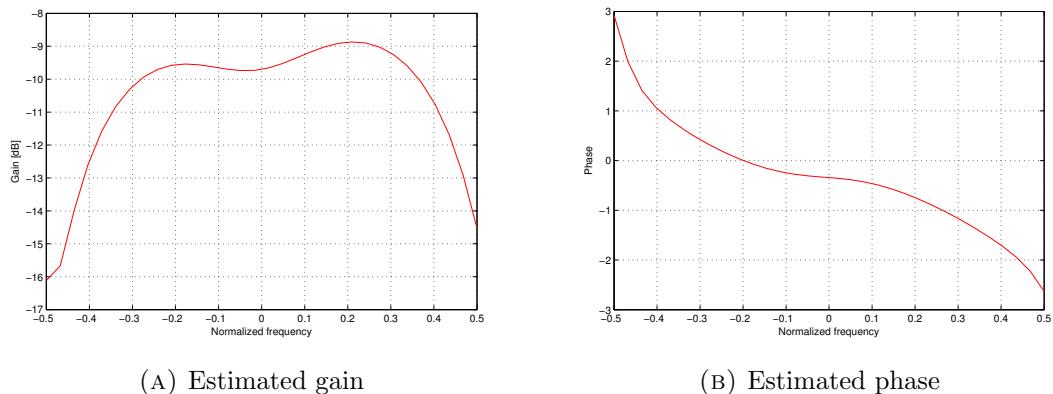


FIGURE 8.20: Estimated channel

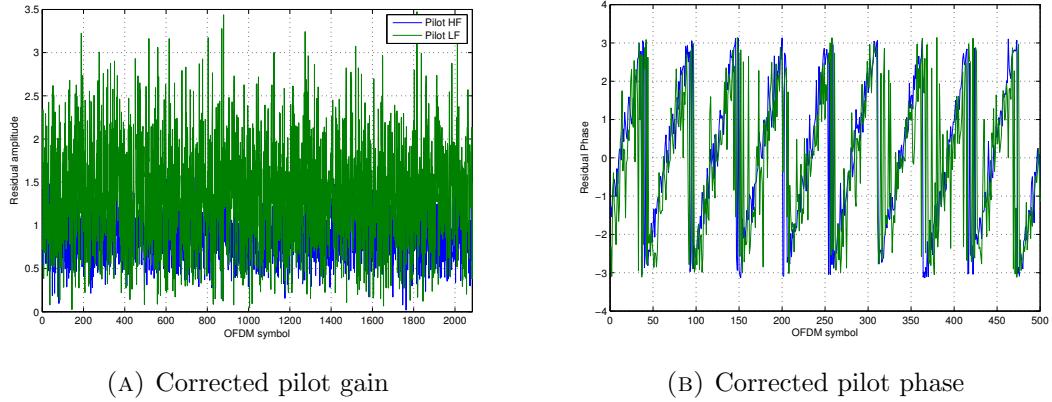


FIGURE 8.21: Pilots after channel correction

The residual phase on the pilot, after correction by the tracking algorithm, is presented in figure 8.22. One can see a value centered at zero for both pilots, though they are both quite noisy.

This system, even if promising in theory, fails to show good BER. It should probably be improved by using more pilots. Indeed, the residual CFO after the joint estimation is too high for the tracking algorithm, which can be seen in figure 8.21b and 8.12. The error scheme is presented in figure 8.23.

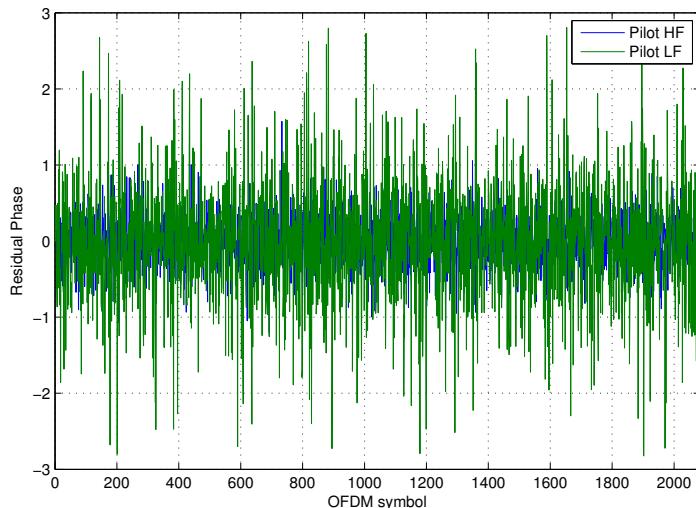


FIGURE 8.22: Phase of the pilot after correction

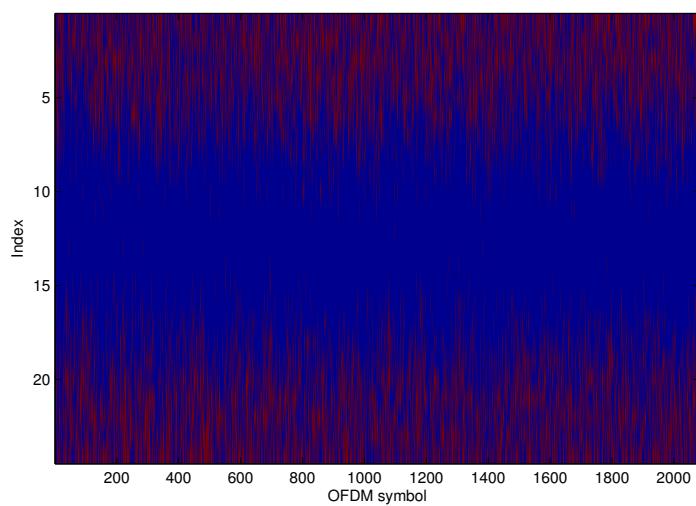


FIGURE 8.23: Position of the errors on the I component

Chapter 9

Conclusion

This project has been a valuable experience of working as a group and solving practical problems together, to achieve the final goal. It demanded various skills from group members and contributed a lot to better understanding of previously studied courses and their application in real life.

9.1 Project Management

The organisation of the project itself was a challenging task since the number of members in the project group was quite large. However, one can consider it to have been solved successfully. The organization of the work has been done in an efficient way and took into account previous experience and background of each member, in that way maximizing everyone's contribution and making the achievement of the results stated beneath possible.

9.2 60 GHz communications

Systems working in the 60 GHz frequency band have huge potential for future wireless communications. We were able to use the advantage of such high frequency and wide channel bandwidth for obtaining high data rates. We also resolved the main issues that we were faced with during design and implementation of the communications systems. The main problem turned out to be channel estimation and the best solution for that was Kalman filtering and linear interpolation. Regarding the C++ implementation, a great

challenge was the processing time. It was significantly improved relative to the very first implementation and required working with threads and other optimization techniques. In the end, this also improved the programming skills of the hardware implementation group a lot.

9.3 System 1

The objective of system one was to familiarize the team with the USRPs, thus constituting a simple but necessary step in the process. It gave the team a possibility to set up a simple communications system, allowing us to identify and face some problems, thereby helping us to avoid similar problems for more complex systems. Some problems, though, were left to explore later. For example, in the first system, the phase noise was not really a problem, the noise was not significant, there was no saturation and the LO leakage was on the side of the channel. However this model was a challenge to code in C++, as it made it necessary not only to understand C++, but also how to use the USRPs.

The implementation of system one gave quite good results in terms of data rate: 12.5 Mbits/s; in terms of decoding delay: 18760 bits decoded in around 16 milliseconds; and in terms of BER: a constant zero error rate. However the method used the LO leakage to identify LO offsets. This method could not have been used in the 60 GHz module. Consequently, a different method had to be developed for 60GHz.

9.4 System 2

The second model was implemented in 60 GHz, but was designed using the 5 GHz-USRPs. It was devised to have a lot of adjustable parameters, e. g. the number of carriers, modulation and channel estimation techniques. This system achieved higher data rates and prepared a background for facing a new challenge: the transmission at 60 GHz, which meant a new channel, more phase noise and a more significant frequency offset. This implementation was more challenging in terms of theory, especially theory related to channel estimation.

The implementation of this second system in 5 GHz provided really high data rates, firstly thanks to the increased amount of used bandwidth (around 20 MHz were used

from a total bandwidth of 25 MHz), and secondly, thanks to the size of the constellation (16QAM, 32QAM and 64BEEs). These improvements resulted in a non-zero bit error rate, which was corrected by using turbo coding. Furthermore, they made the system reach a data rate close to 72 Mbits/s. For some realizations, though, there was a non-convergence of the turbo codes. When using the 32QAM constellation, the frame error rate was almost zero. In order to avoid too long decoding time, the C++ implementation did not include channel coding, leading to a more slightly lower data rate of around 30 Mbits/s.

When this second model was implemented using the 60 GHz-USRP, new problems appeared: more phase noise, higher saturation and a significant frequency offset. The employed counter measure was to make some modifications to the transmission parameters, but no modifications in the system model. The modifications resulted in a reduction of the data rate, but as we get less interference we get a more stable noise power, decreasing the frame error rate of the turbo code. The data rate with coding was 54 Mbits/s (with a non-zero frame error rate) and 45 Mbits/s (with a frame error rate of 0). In terms of real time implementation, as no coding was done, we succeeded to reach a data rate of 18 Mbits/s. However, due to more complex functions, the required real time delay was not achieved, as 18 ms was spent to decode a codeword of 27000 bits. The required decoding time was 0.5 ms. Considering our best result (12 ms for a 27.2 kbits file) and a data rate of 1 Mbps (lower than actually achieved), we achieved the real-time requirement of 13.6 ms.

This model was efficient but had one problem: the PAPR, which resulted in saturation at the transmitter side. Being the limiting factor, it needed to be eliminated or at least suppressed. This became the goal of the third system.

9.5 System 3

This system, implemented in order to use DFT-precoded OFDM, required a completely new approach in order to do estimations in OFDM. The system was simulated and tested on real data with Matlab, as was done with the other systems, but due to the limited time of the course, this third system was not implemented in C++.

The general idea is to estimate the channel and the carrier frequency offset through the use of one OFDM training symbol and content ourselves with two pilots to track the phase. The main drawback is the inability, in the current implementation, to handle too great carrier frequency offset. The channel estimation and the tracking algorithm are limiting factors. In our case, a too great residual frequency offset after the first correction turned out to be detrimental to the tracking algorithm. The achieved rate was close to 57Mbit/s, but with a BER of 0.29.

However, the estimation of the channel proved to be accurate and additional CFO estimation techniques could be easily implemented to improve performances on the tracking. Another possibility is using additional pilots.

9.6 Discussion

In this project we concentrated on soft implementation at computer level instead of integrated circuits. This was a good choice, allowing us to define our system. However, handling a communications systems requires a lot of complex calculations, and it is impossible to design a fast communications system without implementing certain steps of it in hardware.

One huge drawback of our system is the fact that we do not use any feedback from our channel. Of course, in our case it was not possible to implement this, due to the huge time delay. Consequently, seeing that the channel depends a lot on the position of the transmitter and the receiver, it is necessary to tune the gains, the RMS and the constellation that should be used, manually. Because of this, it is nearly impossible for anyone else to use the system, as it needs tuning to work well. This improvement would also present the possibility to use a dynamic allocation of power onto the different subcarriers, in order to get a better data rate.

9.7 Future work

Since this area of study within civil applications is pretty new there are a lot of subjects for future work. Looking at the study conducted in this project one can gather a few of them.

- Lowering of the system's delay by using FPGA.

- Implementation of a system, which would provide information about the channel, as described above.
- Modification of the coding to be more efficient for constellations with large number of points; including the calculation of the extrinsic log-likelihood by going back to the constellation, in the iteration.
- Improvement of the channel estimation to optimize the parameters used within the systems.
- Trying to achieve non-LOS communication over 60 GHz is something that was not touched upon in this project and that would greatly raise the number of possible applications.
- Implementation of the third system, or specifically, a more conventional system in C++, would be interesting since implemented systems are usually very application-specific and seldom strictly following theoretic results.
- Usage of MIMO antennas, more specialized antennas.

In the following subsection, we present some of the items in more detail.

9.7.1 System delay

There are many simple ways to reduce the delay. For example, it is easy to code a correlation of a constant sequence with constant amplitude (it corresponds to the application of a constant sign and then the application of a summation) on an FPGA. It can probably allow for the USRP to start the communication with the computer once the synchronization is done. Moreover it is also possible, in a context where we have already decided the number of carriers, to use a hardware component to do an FFT, in order to only transfer the information carried by the data carriers and the pilot carriers, through the Ethernet cable. In other words, to reduce the processing time, one could use specialized components, hard programming or a DSP-board, which are more commonly used in such a context. Doing these modifications may also reduce the risk of overflow.

9.7.2 Antennas

The advantages and disadvantages of different types of antennas are many but the most important aspect to consider is whether the location of the receiver is known to the transmitter, since for the omni-directional and the directional antenna the location of

the receiver needs to be known or partly known beforehand in order to achieve a good connection [Emami, 2013].

Omni-directional antennas together with some form of noise apart from AWGN, such as a human body, presents huge problems when transmitting at 60 GHz, even at short distances (10m). Antenna gain is an important factor. Looking at a theoretical example, the total gain that is needed for a reasonable connection is 30 dBi. In order to achieve this high level of gain, directional antennas need to be implemented [Huang and Wang, 2011].

Chapter 10

Technology Economy Leadership

The aim of this chapter is to build a short business plan based on the scenario of creating a company out of the project group. The target of the business plan is Anna Jerbrandt who acts as an angel investor or a venture capitalist, i.e. a potential investor, who is interested in the company. The business plan is intended to show the viability of the product, the smartphone module which will be explained later, through a market analysis and the intended strategic choices. The stated short term goal of the venture is to design a device which can be taken to the market. The long term goal is to be bought up by a larger player that can provide the company with the resources needed for further development.

10.0.3 Structure

This chapter will loosely follow the template of a business plan set up by the European Commission [[Competitiveness and Innovation, 2012](#)]. Looking at business plans written in other entrepreneurial courses (Entrepreneurial Marketing, New Venture Creation, Start-up Business Case) this template covers the most important parts of a business plan.

Before the business plan, a short background paragraph will cover what has been done in the project, in order not to force the reader to read the entire report. The business plan, which follows, will be presented in the following order:

- Business summary: summarizing the company, the management and the products
- Market analysis: covering the characteristics of the market together with a SWOT to determine our place

- Strategy, Marketing and Implementation: covering the strategic choices and plans

Lastly, the chapter will be completed with a concluding paragraph reflecting on some of the conclusion that we have drawn from completing this chapter.

10.1 Methodology

The report is a concoction of two different niches. First and foremost, we have the technical report. In the technical report, we show how to design and evaluate a 60 GHz communications system. The work is a combination of ideas taken from prior work, referenced according to the Harvard referencing system, and ideas and results produced by the group. When deciding what references to use, we have often relied on course literature from previous courses, for describing the underlying theory. For solutions to problems that we have been faced with, we have searched primarily the IEEE database, and secondly similar related databases suggested by KTHB Primo, for recently published articles and manuscripts, discussing ways to deal with those problems. Upon finding a selection of methods, we have often tried implementing several of them to evaluate their performance, both in simulations and also in our implemented system. Some have proven too hard to implement, either being too hard to understand or infeasible due to computational complexity. All of these borrowed solutions were originally designed for this purpose, making them suitable in a scientific point of view. By this, we mean that they have already been evaluated thoroughly and also reviewed by the scientific community.

10.1.1 Theoretical framework

The second part (this chapter) of the report is a business plan/report blend, striving to showcase the business potential of a commercializable product based on what we have been researching, and demonstrating in the technical report. A not so subtle undertone of the business plan will be us writers trying to communicate what we have learned during our studies at Industrial engineering and management. As we set out to make this clear, we sat down and thought about what topics could be related to the idea of a business plan. We decided on incorporating material from the courses Industrial Dynamics, Industrial Management, Leadership, Project Management, Industrial Marketing, Entrepreneurship, Research Methods in Industrial Engineering and Management, Technology Business Leadership and Operations Management: Organization

and Control.

The way we have tried to incorporate the courses is by either referencing the material from the actual courses, when this was suitable, or referencing sources of information that can be used in conjunction with the concepts presented in the course. Some concepts are applied without explanation, such as the cost breakdown of the product (something that could be seen as representing the Industrial Management course, but which is too obvious to require an explanation). Other concepts, such as that of salients in industrial dynamics, requires and receives some more presentation.

Some courses have a higher degree of representation in the report, for instance industrial marketing. This is not because of our love towards marketing, but rather it is derived from the fact that it makes more sense to talk about marketing than gender equality in the report we are trying to write (taking the guise of a business plan). A chapter on financing would have made some sense, but was considered less important than the parts we chose to focus on and was therefore omitted. This was primarily due to the fact that a financial part would have consisted of an Excel document with numbers based on guessing which would have benefited the reader very little.

10.2 Background

There is a constantly increasing demand for higher wireless data rates. Manufacturers are always releasing new TV's, projectors, smartphones and digital storage units; pushing the need for high speed transmission of data between devices. With 60 GHz communication it has already been proven that data rates of over 25 Gbit/s can be achieved, enabling 4K material to be streamed wirelessly in 60 FPS. Pushing this technology would enable the wireless streaming of media to go from being a reverse-salient to actually being the salient technology, in media streaming [Davies, 1996]. This could be used to reduce the importance of source coding, decreasing the computational load for the involved devices, or to move greater amounts of data altogether. The movie "Ben Hur" could be buffered in a matter of seconds, an advanced 3D-cad used for printing your favourite design chair could be downloaded in the same amount of time. The short extensions of optical fiber, from the backbone network to a family housing for instance, which can cost several thousand dollars for a few meters, can be substituted with wireless connection performing the same function at a fraction of the cost.

10.2.1 60 GHz communication

During recent years, civil studies of wireless systems have moved from lower, more crowded frequencies towards the less used millimeter-wave frequencies situated in the 30-300 GHz band. Around the 60 GHz band there exists, globally, a large amount of unlicensed bandwidth that is free to use for commercial as well as scientific purposes (just as the 2.4 and 5 GHz bands that are used for wireless LAN today). The affluence of available free bandwidth makes it excellent for implementing high data rate wireless communication systems [Huang and Wang, 2011].

The 60 GHz channel is limited by high oxygen attenuation and path loss, which renders it inefficient when transmitting at longer distances. This can be regarded both as a disadvantage and an advantage, since the relatively short useful distance, tens of meters, also makes it secure, interference limited as well as enabling frequency reuse [Yong et al., 2011], meaning that a lot of these systems can be set up relatively close to each other without reducing the performance of each system.

10.2.2 Communication system

The goal of the technical project, the main focus of the course so to speak, was to design a communications system transmitting at a frequency of 60 GHz. This presents a number of difficulties which the group have been working to solve during the last few months. The work was divided into three parts, one part of the group worked out the theory to make transmission possible. One part of the group implemented the theoretical solutions and the project leader worked out the managerial difficulties and wrote the report. Based on this project, a business plan is presented below. The business plan aims to capitalize on the knowledge gained from the project by designing and producing a 60 GHz module for modular smartphones.

10.2.3 Modular Smartphone

The idea of a modular smartphone builds on the problem of fast evolution in certain parts of the smartphone component market making smartphones obsolete, even though they are still in very good shape and performing their owners tasks at a sufficient level. For instance, a consumer who mainly uses the smartphone for calling, messaging and surfing (which sums up most users) may very well be benefited by a more modern smartphone which does all these things better than the last one. However, just considering what components actually needed upgrading to increase the performance accordingly, it

may turn out that the only difference between the current generation smartphone and the next generation can be found in the processing. This is where the modular smartphone enters.

When the difference between the current generation and the next generation is primarily found in computational power, it is less interesting for the consumer to buy a new device for \$500. What the modular smartphone provides is the opportunity to pay \$50 for the CPU-module, a simple plug in and a self calibrating software, turning the current generation smartphone into the next generation for a tenth of the cost. Also, a user more interested in long battery time can install several batteries in their device, making it possible to go on for weeks without recharging. There are currently a couple of associations working on these systems, the most prominent ones being Phone Bloks, which is an independent organization, and Project Ara, which is an in-house Google venture. The most outstanding of these two is Project Ara, which is backed up with more financing and a development team lead by a formed DARPA top researcher [[McCracken, 2014](#)].

The concept in both these cases is that the developer (Phone Bloks or Project Ara) supplies the skeleton smartphone, which is a stripped down smartphone that has several slots for plug-in modules, and other associations design and manufacture the modules. Our intention is to become one of these associations, supplying the market with a module which facilitates 60 GHz communication.

10.3 Business plan introduction

This chapter is intended to work as a short business plan for this project. The focus of the plan is to outline why and how it would be viable to exploit the knowledge gained from this project by starting a small start-up company. The aim of the company would be to develop and sell a 60 GHz module for the PhoneBloks/Project ARA market.

10.3.1 Proposed product

Our ambition is to design a solution for 60 GHz transmission that can be packaged into a project Ara modular smartphone, and sold through their distribution channels. The module is plugged into the smartphone-shell and immediately offers communication with other 60 GHz devices. It will initially be offered at a premium price, as the product will target early adopters with low price sensitivity in the earlier stages of the market

evolution. As the volume of the market grows, and the hardware gets cheaper, we will release the product at three price levels, one low end, one middle and one high end version where the performance and the price increases for each step. This way, we can keep tapping into the high end, high margin market, while also getting the advantages of volume and contributing to the growth of the market.

Our short term goal is to develop a system that satisfies the basic conditions in order to produce a module making communication over 60 GHz possible; following the template of the Project ARA modular phone. However, our long term goal is to be seen and bought by a bigger player with more experience but that is lacking a specific Project ARA 60 GHz team. Such a player could, for example, be Samsung, Ericsson or Qualcomm. It is reasonable to assume that it would be considered easier, faster and cheaper for them to acquire our company and supply us with the experience and monetary resources that we need, instead of creating a new business unit and recruit new people.

10.4 Management summary

The team consists of 8 students at KTH. The students are of varying nationalities and skills. Two Swedish students are in charge of managing the project as well as making sure that the projects complies with Swedish rules and formalia. The rest of the student are able to either produce and understand the necessary theory as well as theoretical results (such as Matlab verification and validation); or able to implement the, from the theory-group, suggested system into the suggested hardware via *C++* programming. Several of the members are multi-skilled, able to both understand and work with the theory as well the hardware. This makes it easy for the company to switch focus between theorizing and implementing. Since there is a dynamic relationship between the theorization and the implementation this is greatly helping the efficiency of the group and the company.

10.4.1 Team gaps and Other personnel

During the course the team has been mentored by Per Zetterberg at KTH and he is intended to work as a mentor, having decades of experience within the area. Per will also make sure that any gaps that are identified, that cannot be filled by any of the company members, are filled by an outside consultant. However, as of now, no obvious gaps can be found within the company. If we choose to sell the company, several consultants will need to be used, for example to cover the group's gaps within corporate law.

10.5 Market analysis

Following this section is a market analysis of the smartphone market and specifically the potential modular phone-market. We reflect upon the market size, growth and trends in order to explain why we think this market has potential for our business idea. In this section we also look at the profitability and our target group. Following that is the industry cost structure, the distribution channels and the critical success factors. Lastly a SWOT-analysis is used to analyse our Strengths, Weaknesses, Opportunities and Threats in the market.

10.5.1 Market size (current and future)

Smartphone sales worldwide third quarter 2013 equal to 250 million units, 45,8 % higher than the same period 2012. Android over 80% , IOS at just over 12% and Microsoft at 3.6% [Pettey and van der Meulen, 2014].

Smartphone sales revenue of around \$ 50 billion per quarter 2013 [Agency, 2013].

What can be seen is that this is a huge market with high margins for two big players. What can also be observed is that most of the smartphones sold are using Google's Android operating systems. Google is however not one of the big players within the handset sales market.

10.5.2 Market trends

The market size is currently increasing and has done so for the last years as can be seen in figure 10.1.

As can be seen in figure 10.1 Android is the dominating operating system, followed by IOS, these are the major players. We can observe that Microsoft have around a 3 – 4% market share being third and that there really is no other major competition within the market.

10.5.3 Market growth rate

Smartphone sales revenues exceeded that of ordinary mobile phone sales in 2013 [Pettey and van der Meulen, 2013]. Added to this is the fact that in the second quarter of 2013 the number of units sold increased, compared to the second quarter of 2012, with 52.3% measured in year over year growth. This represents the highest growth rate in

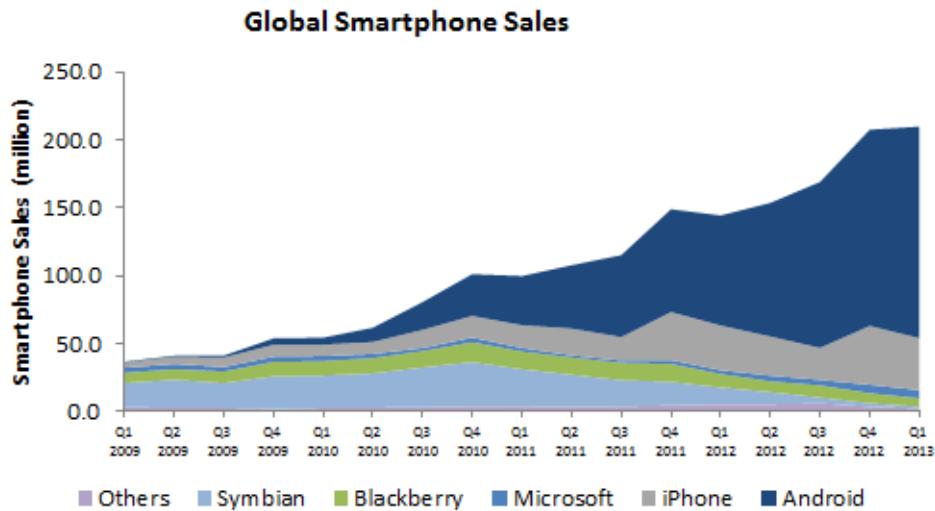


FIGURE 10.1: Smartphone sales trend worldwide 2009-2013 [Singh, 2013].

Worldwide Smartphone Sales to End Users by Vendor in 2013 (Thousands of Units)

Company	2013 Units	2013 Market Share (%)	2012 Units	2012 Market Share (%)
Samsung	299,794.9	31.0	205,767.1	30.3
Apple	150,785.9	15.6	130,133.2	19.1
Huawei	46,609.4	4.8	27,168.7	4.0
LG Electronics	46,431.8	4.8	25,814.1	3.8
Lenovo	43,904.5	4.5	21,698.5	3.2
Others	380,249.3	39.3	269,526.6	39.6
Total	967,775.8	100.0	680,108.2	100.0

FIGURE 10.2: Smartphone sales worldwide 2013 [Pettey and van der Meulen, 2013].

five quarters. Compared to the first quarter the second quarter number of sold units was up 10% [Nagamine, 2013].

10.5.4 Market profitability

Looking at how the market profits are divided among the players it can be seen that it is almost exclusively Apple and Samsung that are doing profits. Where Apple stands for 56% and Samsung at 53% 2013. This being a number higher than 100% derives from the fact that many of the players within this market are losing money, doing "negative" profit [Pettey and van der Meulen, 2013]. For these sales Apple is having a margin of between 49–58% looking at 2010-2012 sales and Samsung is having a margin of 22–25% [Levine, 2012].

10.5.5 Potential market size

Google's vision is that their modular smartphone will be a game changer in the market of cellular phones. The target customer is not primarily the wealthy and technically interested, but the peruvian farmer that has never even had a feature phone. The goal is to make this phone the number one choice for anyone interested in buying a phone. This vision is bold, at the least, and a realization of the entire vision is not altogether likely. There is a great risk that Samsung and Apple will continue their domination in the mature markets. This would be a problem to us, since our product is not mainly oriented towards the Peruvian farmer.

10.5.5.1 Target group

If we suppose that the modular phone could reach a market penetration of 5 percent, the same as several mid-size producers, that would imply 46 million units worldwide. For the mature markets, the United States and western and northern Europe, that would imply slightly below 10 million smartphones on a yearly basis [[Hughes, 2014](#)] and [[Jeronimo, 2013](#)]. Recent findings show that a large portion of 3G and 4G subscribers are actually using more than their monthly data limits, by plan. For 4G devices, about 10% max out their limits even though they have limits above 5 GB, 2013. The corresponding number for 3G was 6%. Looking at the numbers from one year before, one sees that the numbers for both categories have doubled and that more people have also gone from 3G to 4G. We can expect that at least 10% of the smartphones sold on the mature markets will go to the group of subscribers with the highest data usage [[Zarrella and Hibberd, 2014](#)].

Added to this, looking at [10.5.3](#) one can observe that the market has a great growth rate that during the last years have even increased, making the potential target group even larger as time goes by. Even if we see a decline in this rate, the growth is sustainable and substantial.

10.5.5.2 Target Analysis

Calculating on the supposed 10 million modular smartphones on the mature markets, we have a primary target group of 1 million customers per year, in the initial phase. Further down the line, in a few years as things are progressing now, more and more out of the 46 million would also move towards a behaviour where they would benefit from the technology, but then we would also be at the stage where we go for a widening of

Preliminary iPhone 5 Bill of Materials and Manufacturing Cost Estimate Based on Virtual Teardown
(Costs in U.S. Dollars)

Components / Hardware Elements	iPhone 5 Hardware Comments	iPhone 5 Model		
		16GByte	32GByte	64GByte
Pricing without Contract		\$649	\$749	\$849
Total BOM Cost		\$199	\$209	\$230
Manufacturing Cost		\$8.00	\$8.00	\$8.00
BOM + Manufacturing		\$207	\$217	\$238
Major Cost Drivers				
Memory				
NAND Flash		\$10.40	\$20.80	\$41.60
DRAM	1GByte LPDDR2	\$10.45	\$10.45	\$10.45
Display & Touchscreen		\$44.00	\$44.00	\$44.00
Processor	A6 Processor	\$17.50	\$17.50	\$17.50
Camera(s)	8 Megapixel + 1. 2 Megapixel	\$18.00	\$18.00	\$18.00
Wireless Section - BB/RF/PA	Qualcomm MDM9615+RTR8600+Front End*	\$34.00	\$34.00	\$34.00
User Interface & Sensors		\$6.50	\$6.50	\$6.50
BT / WLAN	BTv4.0 + Dual-Band Wireless-N	\$5.00	\$5.00	\$5.00
Power Management		\$8.50	\$8.50	\$8.50
Battery	Assumed 1800mAh	\$4.50	\$4.50	\$4.50
Mechanical / Electro-Mechanical		\$33.00	\$33.00	\$33.00
Box Contents		\$7.00	\$7.00	\$7.00

* - Assumed

Source: IHS iSuppli Research, September 2012

FIGURE 10.3: Production cost of iPhone 5 components [Fiegerman, 2012].

the price range. For the first phase, the single version release, the focus will be on the 1 million target group.

10.5.6 Industry cost structure

Comparing hardware component suppliers, service providers and handset makers, one finds that the highest gross profit is captured by the service providers but that the greatest operational profitability relationship is the reverse [Dedrick et al., 2011].

Observing this relationship we will probably end up in the middle of a supplier or hardware component supplier. This means that our gross profits on each sold module wont be that high, but that our operational margins will be higher. This since we are not weighed down by any huge costs apart from the ones originating directly from the product.

Looking at the iPhone 5 with an estimated production cost of \$207 for production and labour and the retail price of \$649 [Fiegerman, 2012], while knowing that the margins are between 49 – 58% [Levine, 2012] costs can be estimated. A quick estimate makes the margin to about \$330 and thus the difference ($330 - 207 = 123$) gives us the cost of everything outside of production of the phone such as *R&D*, distribution, marketing. Looking at Apple the marketing cost is low compared to competition and distribution will be low due to economies of scale; what will drive the costs are the Research and Development. Looking at other handset makers this will probably not be the same.

Samsung for example spends three times as much on advertising than Apple for their mobile phones [[Ante and Connors, 2013](#)].

We can use this approximation in order to estimate a highest production cost of our modules. The Iphone 5 price of \$649 can be seen as a good estimate for the price of a premium phone, however, with Apple having one of the highest margins in the business [[Levine, 2012](#)] their phones are also more expensive as can be seen in the numbers of phones sold contra the profit made. This makes a premium phone production cost of \$200 – 350 reasonable. Looking at specific components of a smartphone, one can easily see that the driving costs are the Display & Touchscreen, Wireless section and the Mechanical/Electro-mechanical parts [10.3](#). Since most of these components, not Mechanical/Electro-Mechanical probably, will be included in the smartphone skeleton, the remaining cost driving components are processor \$17.5, memory \$20, cameras \$18 and Mechanical/Electro-Mechanical at \$33. This amounts to another \$88.5. Assuming that these features will all be sold at a 100% mark-up, the final consumer price for a high end smartphone will be \$227.

10.5.7 Production cost of module

The foremost cost driving components for the module will be the 60 GHz transmitter and the 60 GHz receiver, along with the FPGA or ASIC which will be used for the signal processing. For the initial production, in a low volume-high price sales scheme, it will be more feasible to use an FPGA. Also, among FPGA's, it makes more sense to use a slightly aged processing unit, since prices for less complex units are much lower than for high-end units. A sufficiently powerful FPGA can cost \$10, according to our contact at Xilinx. The 60 GHz Tx and Rx, on the other hand, can cost \$150 each, according to preliminary contact with the sales representative of a major manufacturer. Seeing that the margins are typically quite high, 40% was mentioned in Qualcomm's 2013 financial statement, it is possible that a high volume customer could negotiate a reduced price, especially considering there is more than one company producing this equipment. Supposing a 50% reduction in margin on the producers behalf, the cost would instead be \$120 per component. Considering the manufacturing costs uncovered in table [10.3](#), we expect the actual manufacturing to be relatively cheap in comparison to the expensive components. On behalf of the low volume, but also the small number of components, we approximate the manufacturing cost to \$5 and the casing another \$5. Summing up these terms:

- FPGA \$10

- 60 GHz Tx & Rx \$120 to \$150
- Casing \$5
- Manufacturing \$5

we expect our unit cost to land at \$260 to \$320.

As the technology grows more established, we can expect the pricing of the transmitter and receiver components to converge with the prices for the BlueTooth/WLAN component, which also contains the signal processing, in table 10.3, reducing our unit cost to something close to \$15. A lower unit cost will be a prerequisite for the price segmentation strategy, but the absolute levels will be left open for now.

10.5.8 Conclusions on Cost

With an end consumer price for all components, apart from ours, at \$227, there will be a vast gap of \$422 between the high end Ara phone and the latest iPhone. If the product can motivate a price of \$400, a very high-tech smartphone will still be available at less than the average price for the latest generation smartphone and the production and development costs can be recovered.

10.5.9 Distribution channels

As the device is so closely connected to the module base, the smartphone skeleton (made by Google), it will make sense to use the same distribution channels. It is Google's intention to set up an e-commerce site for all related products, so this would be the natural and most important channel of distribution. They will also set up physical automated kiosks, where the system suggests modules depending on parameters such as how you currently use your phone or how you behave in social media [McCracken, 2014].

A drawback of using a Google-administered distribution channel is that they will control distribution, meaning that they will control our margins. Other similar channels already in use, such as App-store [Beckman, 2011] for smartphone apps or Steam [Long, 2013] for digital games, demand a 30 percent commission for all sales that are administered. In the short run, we are obliged to use this distribution channel, but hopefully we can set up a separate distribution channel further down the road.

10.5.10 Critical success factors

10.5.10.1 In the market

- *Market at critical mass.* As previously explained, the release of this product will be critically hampered if there are no compatible products that it can connect to. Since this is a common problem to all developers in the market, trade associations such as the Wi-Fi alliance and [WirelessHD](#) have a high rate of association among the developers. As a consequence, it is very easy to get up to date information on what is happening in the market and how work is progressing among the developers. Some products are already on the market, mainly wireless equivalents of HDMI cabling, and many more will roll out in the coming years.
- *Market penetration of modular smartphones.* Selling a module to a modular smart phone limits the market to those customers that are interested in the modular device. If the smart phone doesn't take off, there's no use in taking this module to the market.

10.5.10.2 In the product

- *Size of module.* The products that are on the market right now are too large to be integrated into a smartphone. MDK (Module Development Kit) specifications for project Ara give an idea on what delimitations, in size, must be accommodated in the design. If the module cannot meet the requirements, it cannot be used.
- *Cost of module.* This is a premium product, so of course the pricing should be rather steep. Working with target pricing, we must identify an ideal target price for the device, the feasibility of meeting the price and finally a level of acceptable costs [[Omar, 1997](#)]. If the hardware prices don't match the target, we cannot go forward with the production. In the closely related smartphone market, only two producers are currently profitable, and we must remain vigilant to the risk of committing to a project that cannot meet profitability targets.
- *High speed bus and memory in the phone base.* To make the device useful to the consumer, the phone that the module is docked into must be able to handle the high data rates that will be generated by the device.

10.5.11 Competitor analysis

Currently, no other producers have announced that they're working on modules for the project Ara smartphone. However, there are several companies, such as Wilocity,

Perasotech, Blu Wireless Technology and Nitero for instance, that are working on this technology. The main focus is to produce systems for high data rate mobile backhaul and chipsets that can be used by other developers. As project Ara is still quite low key in the market, most developers are working towards projects with lower risk, such as supplying components for large telecommunications corporations.

Our competitors are more experienced in this technology, and they have more financial muscle. They will move fast once they see the opportunity. Our way of dealing with this is to start early and to be ready to release the product once the modular smartphone is launched.

10.5.12 SWOT analysis

The intention of the analysis is to lay bare the strengths and weaknesses that could be specific to us, when compared to our likely competitors. Opportunities and threats are a combination of factors that are specific to us, as well as for the market we are approaching.

10.5.12.1 Strengths

Early at work - Since the modular smartphone is still at a somewhat early stage, there is no overcrowding on the supplier side yet

Clear focus - Staffed with a young workforce with a single focus on the product, we know that we can work with devotion

10.5.12.2 Weaknesses

No prior product - Since the development team is comprised of young people without any prior experience in taking a product to market, it would be very beneficial to team up with a more experienced partner.

Lacking funds - As the development is idea-driven, we cannot expect to take this to the market ourselves. We must either sell a portion of or the entire company.

10.5.12.3 Opportunities

Burgeoning niche - Early presence at the module market will hopefully open up the possibility to expand into other kinds of modules for the smartphone.

Expanding market - Even though the potential market is currently quite small, there is a great chance that the modular smartphone will have a good chance at high penetration in developing markets, due to the ambitious pricing. This could be beneficial if we're ready to tap that market with low cost products.

10.5.12.4 Threats

Failure to penetrate mature market - As our plan relies on an initial phase where we sell the product on the high-end mature markets, and where we recover the sunk cost from project development, low market penetration on the mature market would mean that we can only sell low margin products.

Experienced competitors - There are many competitors that are involved in development of 60 GHz technology. There is of course the risk that these developers also try to go for a similar solution.

Functionality of future WiFi - There is a great risk that this technology will be incorporated into future WiFi-setups. This would imply that 60 GHz access could be standard also in the modular smartphone base, meaning that our product would be redundant. If this happens, it will happen in a couple of years or so, leaving us a couple of years to recover our sunk costs.

10.6 Operations plan

In the project we are working with agile software development in order to ensure that we are working efficient within the correct areas. We have based our project management on the book *Agil Projektledning* found in [Gustavsson, 2011]. When we have been working with the software we have made sure to work in cross-functional teams helping each other, being motivated by a project leader. Focus has continuously been on coding applications that work; documentation has been important but it has been the primary goal to get functions working and then explaining it to the others. In order to be certain of delivering the correct products we have had continuous input from the project leader

and the mentor. When problems have risen focus have been on adapting the group and dynamically solving the problems instead of following a strict set of work tasks. The entire group has been united in believing that strict hierarchies and rules wont be the solution, however, everyone is agreeing that goals, milestones and activities, as well as structured teams and roles needs to be set up, in order for everyone to know what to do and where the project is going. We always try to measure progress as finished working functions, and later, finished working systems. Building systems first as a group of simple functions and then refining the functions in order to perfect the system. A code-driven framework has been set up in order to test the specific functions separately as well as testing entire units of functions such as the transmitter.

We have been working in weekly iterations, where each week, in standardized meetings, the different groups (the theory group, the implementation group and the project leader) have explained what activities have been done, what is left and if they are proceeding according to the Gantt-chart, as well as their dynamic sense of how much work they think is left to complete. In each team, a member is acting as a customer representative being responsible to the project leader acting as the project owner, making sure that the customers expectations are being met. Added to the meetings is that we are constantly focusing on keeping each other in the loop and asking for help when we get stuck or when we are behind schedule, as well as offering help when we are in front of schedule. Pair programming has been continuously used when designing more complex functions, or when putting together larger units such as the transmitter. This has been used in order to ensure the quality of the work as well as making sure that complex functions are understood by two of the members.

Right now we have finished our implementation of a fully functional system transmitting over 60 GHz. The system contains both a transmitter and a receiver. We are now entering a phase were we need to determine what hardware will be needed and what supplier fit our purpose best. When we get the new hardware we will focus on implementing our code within it in according to our agile work methods. When the product is completed it will be put into the Google ARA store. At this point the company will enter a maintenace and upgrade phase where the main purpose will be to make sure that the product works as it should and that bugs are solved and the software updated.

At a later stage we will, instead of selling and maintaining one premium product, release 3 modules in different price range in order to cover all the demands of the market.

10.6.1 Activity plan

First of all we need to research the appropriate system and implement it into a prototype hardware unit in order to get a system that is robust and effective and in order to be able to do thorough testing and debugging. The main task of this step is to determine what the system will look like and what hardware will be required. This in order to acquire hardware that is just powerful enough to handle our application in order to push the price of the hardware down.

Secondly we need to actually acquire suitable hardware and implement the system from the first paragraph on that hardware. A testing and debugging period will be needed here as well in order to get everything to work the way we want to with the new hardware. In this step good relations with one or more suppliers need to be made so that we can keep as small of an inventory as possible while still maintaining a low price.

The last step is concerned with actually selling the product and continually updating and maintaining the product and our customer relations making sure we stay competitive. Quality in delivering and maintaining our product will ensure that we get a competitive edge as well as make sure that our current customers stay happy, spreading the word and continuing to buy our products.

10.7 Marketing plan

In order to explain our marketing strategy we use a marketing mix method. We explain and motivate below how the company will handle Price, Product, Promotion and Place in order to succeed within the market.

10.7.1 Product

We aim to please customers buying the PhoneBloks/Project Ara concept. The ones with the correct basic smartphone will be able to enhance it with our module. The customer we are looking at is a not too price sensitive, moderately or highly technically savvy person who needs the perks of being able to transmit and receive in the 60 GHz spectrum, primarily the promise of high data rates. The short term potential of the technology is transmission at rates of 4 Gbps. The long term theoretical potential of the technology is rates at 25 Gbps, very far beyond the 0.1 Gbps often supplied by the current high performance wireless modems. The implications should be apparent to any

experienced user of the internet in general and streamed media in particular. In order to meet these demands continuous product development will take place to make sure that the product/s stays competitive.

10.7.2 Business location and Marketing budget

As a consequence of how Google wishes to structure the venture of project Ara, there are already some inherent features concerning how to move this product. First of all, all modules available must be certified by Google and sold through their distribution channel. Google will suggest suitable modules to presumptive buyers on the basis of their (the customers') prior behaviour and how they use their smartphones. This will hopefully help clients find the product. Another important way of reaching the target customers, the early adopters, is to advertise the device in media directed towards the technically savvy. These two channels should suffice to reach our target customers.

Regarding the budget, we cannot put a figure on this for the time being. As Google will urge users to go through their distribution channels, there is no use in trying to work out ways for direct sales. This can change in the future, though, and selling via a channel under our control will hopefully increase our share of the revenues.

10.7.3 Pricing and Price differentiation

Launching a product in the consumer electronics market implies certain constraints on the useful product lifespan as well as the extent of the pay-off period [Prasad, 1997]. The development cycle-time is very short and the period of profitability is commonly around 4 years. The most profitable period is generally the dawn of a new market segment, or new technology.

When pricing a device, it's common to think of the price as a figure composed of the cost per unit and a desired mark-up, a model called cost-plus pricing. Another way of considering the pricing problem is to view the optimal price as the value ascribed to the product, perceived-value pricing [Cannon and Morgan, 1990], the cost left as a fixed parameter and the mark-up, i e the difference between the price and the cost, being the value created by the seller. The method of perceived-value pricing requires more knowledge about the market and what the buyers are willing to pay. This knowledge can be hard to acquire, but if successful the method also offers a chance at maximizing the profits. This is where we want to be.

To fully maximize the profit, it is often preferable to work out a scheme for differential pricing, or price discrimination. In [Pitt et al., 2001], it was commented that there is great difference in what revenues and profits can be derived from different customer segments on the market. An example was the Mexican cellphone company which could trace 90 percent of its sales to the top 10 percentage of active customers and that about 80 percent of their customers accounted for less than 10 percent of the sales. Even though figures varied for different sectors and companies, the pattern is very much present and can be expected to apply to our product as well. Our response to this is to launch a single high-end version of the product with a price skimming strategy, meaning that we intend to reach the market at a very early stage with a relatively high price, then let the price slide downwards. This implies a temporal discrimination, where the early adopters will have to pay a higher price. Once competition presents itself, we release a mid- and low-end version and leave the price more constant over time. This way, we can try to establish an ideal price for each segment. As there is little possibility of the product generating further revenue after the initial sale, and since there will be no lock-in effect working in our favour, there's no use in going for a low price strategy to gain market shares. The only objective is to recover as much as possible in the profitable first stage, then serve the market with a great product.

The intended price of the project Ara phone will be \$50. This will include some, but not all of the normal functionality of a modern smart phone. In order to "upgrade it", by adding modules, to the status of a high end smart phone, one would need to buy a high performance processor at \$35, additional memory at \$30, MEMS at \$20, camera at \$30 and an additional battery at \$15. This will, all in all, amount to a retail price of \$180, which is quite close to the release price of \$199 for [Google Nexus 4](#). The prices have been approximated using typical hardware prices for Iphone 5, which we consider to be a premium product, in combination with typical gross margins for large scale manufacturers of the respective components [Trefis, 2014] and [Olson, 2013]. The reason for the relatively low price in comparison to Apple's and Samsung's phones is that Google representatives have stated that the objective of this venture is a game changing product, rather than profits.

"This is potentially the only thing out there that could turn into a five-billion-unit kind of device. The fact that that will accrue benefits to Google downstream is so obvious that I don't have to make a business case."

Paul Eremenko, Project Leader Project ARA

Also, a large portion of the hardware cost of a premium smartphone derives from components that are already included in the project Ara phone, i.e. the first \$50 will give you a lot of the functionality that is needed in a phone.

A conclusion from this is that we can set a relatively high price and still have a competitive price for the entire smartphone, including modules and everything. Another conclusion is that many modules will probably congregate at a price slightly below the price of the base unit. This may work as something of a psychological cap for the price. What we consider a reasonable price for our model is something within the span of \$40 – 60. The target gross margin that we are aiming for is 50%. This leaves an interval at \$20 – 30 for Bill-Of-Materials, BOM.

10.8 Concluding chapter

Looking at the chapter above, we have tried to target the business plan so that a potential investor would be interested and be more likely to invest (as is the purpose of the business plan). However, looking closer at this idea we feel that it would be hard to accomplish anything else than possibly get bought and included in another company. What would probably be more likely is for a different company to just try to recruit the members of the group, thus going around the actual investment in the company.

As can be concluded when looking at the unit cost, ranging from \$260 - 320, there is no chance at reaching the stated target cost of \$20 - 30 for the time being. Even though the component costs will decrease in the near future, and keep on decreasing, we will have to look into the possibility of increasing the retail price. As stated in the Product Cost section, there may be room for a steep increase in the target retail price of the module, but this matter must be studied further.

Finally, we felt that a business plan made it possible for us to showcase what we have learned during our years in Industrial Engineering and Management, in a way that would not have been possible in an essay.

Bibliography

- VS Abhayawardhana and IJ Wassell. Common phase error correction for ofdm in wireless communication. In *Proc. IEEE Global Telecommun. Conf (Globecom'02)*, volume 1, pages 17–21, 2002.
- Mohamed A Aboul-Dahab, Esam AAA Hagrass, and Ahmad A Elhaseeb. Papr reduction based on dft precoding for ofdm signals. *International Journal of Future Computer and Communication*, 2013.
- Yonhap News Agency. Samsung world's no. 1 smartphone seller in q1: data. *Global Post*, 2013.
- Spencer E. Ante and Will Connors. In the smartphone race, money talks for samsung. *The Wall Street Journal*, 2013.
- Ana Garcia Armada. Understanding the effects of phase noise in orthogonal frequency division multiplexing (ofdm). *Broadcasting, IEEE Transactions on*, 47(2):153–159, 2001.
- Mel Beckman. What the app store future means for developers and users. *PcWorld*, 2011.
- Hugh M Cannon and Fred W Morgan. A strategic pricing framework. *Journal of Services Marketing*, 4(2):19-30, 1990.
- Executive Agency For Competitiveness and EACI Innovation. General guidelines for developing a business plan, 2012.
- Andrew Davies. Innovation in large technical systems: the case of telecommunications. *Industrial and Corporate Change*, 5(4):1143-1180, 1996.
- Jason Dedrick, Kenneth L Kraemer, and Greg Linden. The distribution of value in the mobile phone supply chain. *Telecommunications Policy*, 35(6):505-521, 2011.
- Paulo SR Diniz, Eduardo AB Da Silva, and Sergio L Netto. *Digital signal processing: system analysis and design*. Cambridge University Press, 2010.

- Shahriar Emami. *UWB Communication Systems: Conventional and 60 GHz*. Springer, 2013.
- Seth Fiegerman. The 16 gb iphone 5 costs apple \$207 to build [report]. *Mashable*, 2012.
- Erico Guizzo. Closing in on the perfect code. *IEEE Spectrum*, 41(3):36–42, 2004.
- Tomas Gustavsson. *Agil projektledning*. Bonnier Utbildning AB, 2011.
- Kao-Cheng Huang and Zhaocheng Wang. *Millimeter wave communication systems*, volume 29. John Wiley & Sons, 2011.
- Neil Hughes. Apple's iphone led 2013 us consumer smartphone sales with 45% share - npd. *AppleInsider*, 2014.
- Kazi Mohitul Islam, Habib Muhammad Nazir Ahmad, Chowdhury Akram Hossain, and AKM Arifuzzman. Performance comparison between traditional and gray-mapped 16-qam scheme with ofdm in both awgn and rayleigh fading channel. *International Journal*, 2011.
- Francisco Jeronimo. The western european mobile phone market declines in 1q13, driven by the smartphone slowdown, says idc. *IDC Analyze the Future*, 2013.
- Sarah J Johnson. *Iterative error correction turbo, low-density parity-check and repeat-accumulate codes*. Cambridge University Press, 2009.
- Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- Dan Levine. Apple u.s. margins for ipad about half of iphone: filing. *Reuters*, 2012.
- Darryl Dexu Lin, Ryan A Pacheco, Teng Joon Lim, and Dimitrios Hatzinakos. Joint estimation of channel response, frequency offset, and phase noise in ofdm. *Signal Processing, IEEE Transactions on*, 54(9):3542–3554, 2006.
- Shou-Yin Liu and Jong-Wha Chong. A study of joint tracking algorithms of carrier frequency offset and sampling clock offset for ofdm-based wlans. In *Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on*, volume 1, pages 109–113. IEEE, 2002.
- Neil Long. Humble store revenue split preferable to steam and gog, says phil fish. *Edge*, 2013.
- Harry McCracken. Google project ara modular phone. *Time*, 2014.

- Marc Moeneclaey. The effect of synchronization errors on the performance of orthogonal frequency-division multiplexed (ofdm) systems. In *Proc. COST 254 'Emerging techniques for Communication Terminals'. Toulouse, France. Paper 5.1. Pag. 5. July 7-9 1997. Enseeiht. 6 figg, 1 tab*, 1997.
- Nektarios Moraitis and Philip Constantinou. Propagation modeling at 60 ghz for indoor wireless lan applications. In *Proc. IST*, volume 2, 2002.
- Kathy Nagamine. Idc worldwide mobile phone tracker. *IDCPress Release*, 2013.
- Parmy Olson. Samsung posts record profits in q3 as handset growth slows. *Forbes*, 2013.
- Ogenyi Ejye Omar. Target pricing: a marketing management tool for pricing new cars. *Pricing Strategy and Practice*, 5(2):61-69, 1997.
- Matthew G Parker, Kenneth G Paterson, and Chintha Tellambura. Golay complementary sequences. *Wiley Encyclopedia of Telecommunications. Wiley Interscience, Chichester*, 2002.
- Christy Pettey and Rob van der Meulen. Gartner says annual smartphone sales surpassed sales of feature phones for the first time in 2013. *Gartner*, 2013.
- Christy Pettey and Rob van der Meulen. Gartner says smartphone sales accounted for 55 percent of overall mobile phone sales in third quarter of 2013. *Gartner*, 2014.
- Leyland F Pitt, Pierre Berthon, Richard T Watson, and Michael Ewing. Pricing strategy and the net. *Business Horizons*, 44(2):45-54, 2001.
- Sweden Post och Telestyrelsen. Pts frekvensplan, 2014. Accessed: 2014-04-10.
- Biren Prasad. Analysis of pricing strategies for new product introduction. *Pricing Strategy and Practice*, 5(4):132-141, 1997.
- Proakis. *Digital Communications 5th Edition*. McGraw Hill, 2007.
- Donald Reay. *Digital Signal Processing and Applications with the OMAPL138 Experimenter*. John Wiley & Sons, 2012.
- Georgia Rugumira, Gang Wang, Xin-Lin Huang, Nguyen Thi Dieu Linh, and Jue-Ran Li. Papr reduction in precoded ofdm system via carrier interferometry. In *Communication Technology (ICCT), 2011 IEEE 13th International Conference on*, pages 456–461. IEEE, 2011.
- Magnus Sandell and Ove Edfors. A comparative study of pilot-based channel estimators for wireless ofdm. *Lulea Univ. of Technol., Lulea, Sweden, Res. Rep. TULEA*, 1996, 1996.

- Tapan K Sarkar, Robert Mailloux, Arthur A Oliner, Magdalena Salazar-Palma, and Dipak L Sengupta. *History of wireless*, volume 177. John Wiley & Sons, 2006.
- Timothy M Schmidl and Donald C Cox. Robust frequency and timing synchronization for ofdm. *Communications, IEEE Transactions on*, 45(12):1613–1621, 1997.
- Kai-Uwe Schmidt, Markus Muck, Xavier Miet, and Jens Schönthier. Estimation of large frequency offsets using 5ghz wlan training symbols “. In *IST Mobile & Wireless Communications Summit conference, Lyon, France*, 2004.
- Yushi Shen and Ed Martinez. Channel estimation in ofdm systems. *Application note, Freescale semiconductor*, 2006.
- Sameer Singh. Global smartphone market share trends q1 2013: Android extends lead over iphone, windows phone performance mixed. *techthoughts*, 2013.
- Team Trefis. Qualcomm earnings preview: Facing asp and margin concern despite holiday season boost. *Forbes*, 2014.
- Su-Khiong Yong, Pengfei Xia, and Alberto Valdes-Garcia. *60GHz Technology for Gbps WLAN and WPAN: from Theory to Practice*. John Wiley & Sons, 2011.
- Danielle Zarrella and Jen Hibberd. New global research on lte usage demonstrates 4g’s positive impact on mobile operators’ business, 2014.