# 1 ICT Laboratory Overview

## 1.1 Introduction

The ANT part of the ICT laboratory held in the summer term is meant to be solved in groups of two in an independent fashion with minimal help from tutors. You are expected to solve problems on your own and organize your work as you see fit. To provide time for questions, initial instructions and to evaluate your implementation 4 laboratory dates of approximately 4 hours each and a short setup meeting are scheduled. The whole lab is structured into an initial setup appointment (1-2h) and 4 dates (4h each).

CIT masters are expected to implement a basic OFDM point-to-point transmission chain. Also, a short presentation about one of the relevant OFDM blocks has to be given.

In the following, Section 1.2 discusses the specific goals and requirements of this lab in more detail. Then, Section 1.3 introduces the lab dates and the general timing of the lab over the whole winter term. The explanations of the specific are given in Section 2. Finally, Section 3 explains the evaluation guidelines that will be used to judge if the lab has been passed successfully or not.

## 1.2 Goals and Requirements

### 1.2.1 Requirements

This laboratory is mandatory for CIT master students. Besides different Bachelor backgrounds we expect you to have certain knowledge and skills at the beginning of the laboratory. To some degree, it is expected that you will have to research topics less well known to you, but nonetheless the following is expected:

- Self-motivated working style (researching unknown topics with minimal tutor help)
- Communications technology knowledge
- Basic knowledge of MATLAB
- Basic knowledge of presentation techniques / software (e.g., LaTeX Beamer or Powerpoint)

### 1.2.2 Goals

The following goals are targeted with this laboratory:

- Self-motivated problem solution including research and collaboration with other lab attendees

- Deepening knowledge about all the basic processing steps in multi-carrier communications from system dimensioning to equalization
- Developing programming skills in Matlab
- Practicing presentation of technical details / procedures (written and oral)

### 1.3 Time line

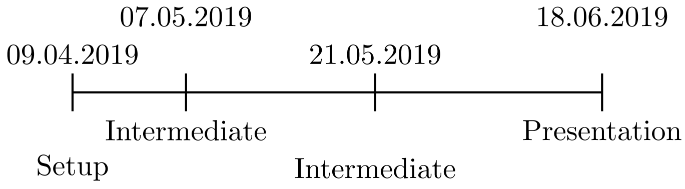


Figure 1: Time line for ICT Lab 2 summer semester 2017.

This laboratory is planned to be running alongside other courses during the whole summer term with only 5 predetermined lab dates for an overall workload of 1CP or 30h for CIT masters. Figure 1 shows the specific dates for summer semester 2018 and their individual purpose. The first date is mandatory for all students attending the lab and will be used to clarify the setup, goals and requirements again. It is expected that all students read this script beforehand and are well prepared. After this initial meeting, intermediate lab time is offered to continue the work on the task, ask questions and get help with individual problems. The last date is again mandatory and the deadline to finish the lab. The tutor will test each groups implementation according to the API requirements described in Sections 2.2.2 and 2.2.4, check the code for proper comments and fulfillment of the test defined in Section 3.4.1. Groups who do not pass this test will have to repeat ICT lab 2 in the following year. The final date of the lab will be used to check the fulfillment of the overall tasks and to present the findings of each group in a 5-7 minute presentation of maximum 5 slides.

## 2 Task Description

### 2.1 General Description

The general idea of this lab is the implementation of a complete point-to-point OFDM communication chain as illustrated in Fig. 2, including transmitter, channel and receiver. To restrict the breadth of this task, only the baseband processing at transmitter and receiver indicated by white blocks has to be implemented by each group. An equivalent baseband channel model will be provided to test the overall communication chain. This model summarizes all channel and hardware effects that are attributed to bandpass

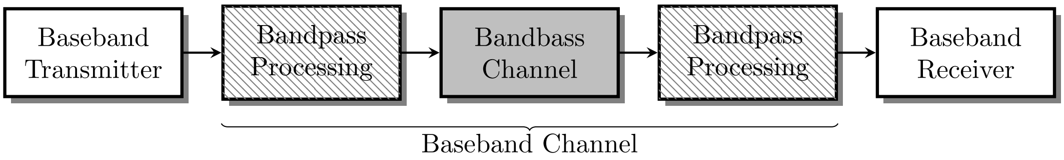


Figure 2: Overview of a point-to-point communication setup. Shaded and gray marked blocks will be provided.

processing, including but not limited to up/down conversion, amplification, antenna patterns, and so on. However, some of the bandpass effects will be included into the lab by equivalent baseband descriptions as "non-linear hardware" (see the following sections for more details).

The baseline OFDM system has to be implemented according to the specifications in Section 2.2,

## 2.2 Basic OFDM transmission

In addition to the transmitter and receiver implementation a simulation environment has to be created that uses the transmitter and receiver implementations to numerically analyze the performance of the whole multi-point-to-point communication chain. The following requirements have to be fulfilled:

- Allow simulation of different SNRs, e.g., using an outer loop.
- Save the results in terms of uncoded/coded bit error rate (BER) for different SNR choices in a vector.
- Save the results in terms of PAPR of `x` for each user at the non-linear hardware outpout
- Plot the uncoded/coded BER per user vs. the SNR.
- Plot the PAPR

### 2.2.1 Transmitter Model

In Phase 1 of this lab, the transmitter and receiver (see, Section 2.2.3) of a basic point-to-point communication chain has to be implemented according the the specifications below. Fig. 3 shows the building blocks of such a transmitter of which all white blocks need to be implemented, whereas the gray blocks will be provided. Each block is defined by its inputs and outputs and a short requirements list that describes the functionality in Section 2.2.2. Your task is the fulfillment of these requirements for each block while adhering to the specified inputs, outputs and function names. Please note, that some blocks are marked as "switchable" by a parameter `switch_off`, which means that such

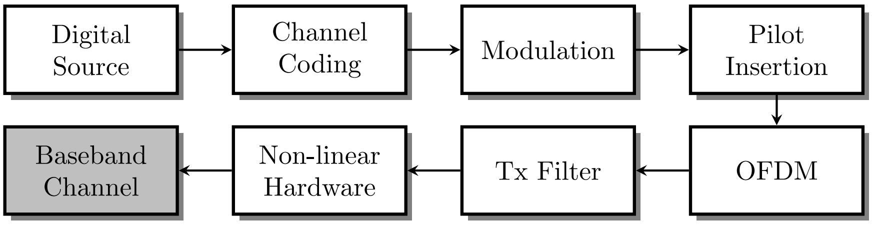


Figure 3: General structure of the Baseband Transmitter as introduced in Fig. 2 with interface to Baseband Channel. Gray blocks will be provided, white ones are to be implemented according to the specifications.

a block should not change the input data in any way if switched “off” by `switch_off=1`, i.e., output=input.

Additionally to functional requirements, e.g., a specific Tx filtering, also optional graphical output may be required. For example, the Tx filter input and output may be plotted in a figure to show changes in the shape of the spectrum. Graphical output should always be optional, i.e., controlled by a switching variable `switch_graph`, to analyze your implementation and the results as needed.

### 2.2.2 Transmitter API Definitions

```
b=digital_source(par_no,switch_graph);
```

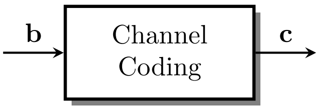
Digital source providing frames of random binary data.

**Parameters:**

`par_no` indicates the frame length.

Choose the frame length `par_no` according to the OFDM parameters.

```
c=channel_coding(b,par_H,par_N_zeros,switch_off);
```



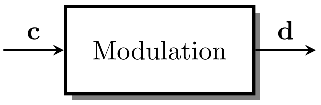
This block facilitates Channel Coding by a [7,4] Hamming code.

**Requirements:**

1. Encode the block of length `par_ccblklen` of the binary signal **b** via the [7,4] Hamming block code.

Note: the required number of bits may not be exactly achievable due to the coding. Add `par_N_zeros` zeros after encoding to alleviate this effect.

```
d=modulation(c,switch_mod,switch_graph);
```



This block facilitates Modulation of the encoded bit sequence to either 4-, 16-, or 64-QAM.

**Requirements:**

1. Modulate data to either 4-, 16- or 64-QAM with Gray mapping. `switch_mod=0` indicates 4-QAM, `switch_mod=1` 16-QAM and `switch_mod=2` 64-QAM.
2. Normalize the average symbol power to 1.

```
D=pilot_insertion(d,par_N_FFT,par_N_block,switch_graph);
```

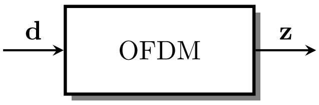
d
Pilot Insertion
D

This block facilitates the framing of data into `par_N_block` blocks of `par_N_FFT` symbols and prepends pilot data for one OFDM pilot symbol.

**Requirements:**

1. Restructure the symbol stream **d** into `par_N_block` frames to **D**.
2. Prepend a frame of `par_N_FFT` known pilot symbols for channel estimation at the receiver.

`z=tx_ofdm_mod(D,par_N_FFT,par_N_CP,switch_graph);`

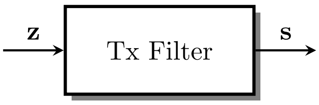


This block facilitates OFDM modulation and CP insertion. The parameters should be choosen to achieve Inter-Symbol Interference (ISI) free transmission given the channel statistics.

**Requirements:**

1. Use a standard FFT length `par_N_FFT=1024` for OFDM modulation, but keep it variable for other choices of `par_N_FFT`.
2. Chose the CP length `par_N_CP` according to the required channel statistics.
3. Serialize the `par_N_block`+1 OFDM symbols to a single time sequence **z**.

`s=tx_filter(z,par_tx_w,switch_graph,switch_off);`



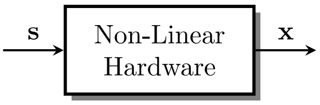
This block facilitates filtering of the OFDM symbols with a digital low-pass filter to suppress out of band radiation from OFDM modulation. This is relevant to decrease interference on neighboring channels (e.g. in WLAN).

**Requirements:**

1. Filter a block of symbols with an appropriate filter to reduce the out of band radiation 40dB below the signal level using an oversampling factor of `par_tx_w=20`.
2. Normalize the filter output signal appropriately to ensure that the power of the signal is 1.

Note: Matlab filter design tools can be used to design an appropriate filter (see, e.g., DSP exercises).

```
x=tx_hardware(s,par_txthresh,switch_graph);
```

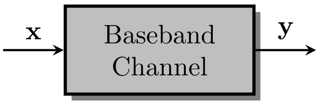


This block models the influence of an amplifier on the baseband signal by hard thresholding.

**Requirements:**

1. Implement a simple hard thresholding function that limits the *absolute value* of the baseband signal `s` such that it is linearly scaled to be smaller than 1 up to values of `par_txthresh` and clipped to 1 if greater than `par_txthresh`.
2. Ensure that the phase of `s` is not changed by this block.
3. Analyze distortions by this block with different threshold levels (e.g., weak clipping, no clipping, etc.).

```
y=channel(x,par_SNRdB,switch_graph);
```



This block models a frquency selective baseband channel that distorts the wideband OFDM signal and adds white gaussian noise to the signal.

**Parameters:**

`par_¬`<br>`SNRdB` will be used to check the performance of the transceiver chain at different SNRs (in dB).

### 2.2.3 Receiver Model

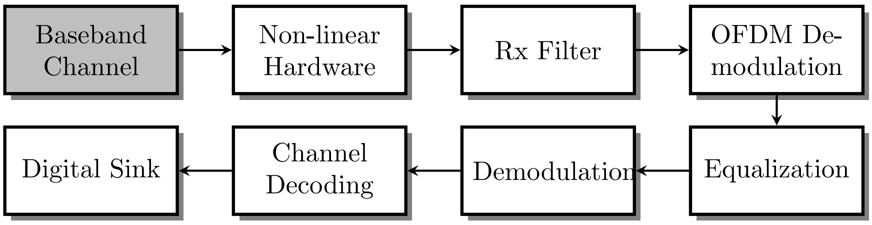


Figure 4: General structure of the Baseband Receiver with interface to Baseband Channel. Gray blocks will be provided, white ones are to be implemented according to the specifications.

The first phase of this lab also comprises the implementation of the OFDM receiver for a frequency selective block-fading channel and the overall simulation. Fig. 4 shows the building blocks of such a receiver and Section 2.2.4 details the individual blocks in terms of inputs, outputs and requirements. To simplify the task some parameters can be assumed as known at the receiver side, i.e., the scaling of the transmit signal is also known. This also applies to modulation, channel code and frame length.

In addition to the receiver implementation a simulation environment has to be created that uses the transmitter and receiver implementations to numerically analyze the performance of the whole point-to-point communication chain. The following requirements have to be fulfilled:

- Allow simulation of different SNRs, e.g., using an outer loop.
- Save the results in terms of uncoded/coded bit error rate (BER) for different SNR choices in a vector.
- Plot the uncoded/coded BER vs. the SNR.

### 2.2.4 Receiver API Definitions

```
s_tilde=rx_hardware(y,par_rxthresh,switch_graph);
```

y
Non-Linear Hardware
s̃

This block models the influence of an amplifier on the baseband signal by hard thresholding.

**Requirements:**

1. Implement a simple hard thresholding function block that is transparent and does not change the signal in any way.

```
z_tilde=rx_filter(s_tilde,par_rx_w,switch_graph,switch_off);
```

s̃
Rx Filter
z̃

This block facilitates filtering of the received signal with a digital low-pass filter.

**Requirements:**

1. Filter the received signal with a matched low-pass filter using an downsampling factor of `par_rx_w=20`, i.e., identical to the transmitter side.
2. Normalize the filter output signal appropriately to ensure that the power of the signal is not changed.

```
d_tilde=ofdm_demod(z_tilde,par_N_FFT,par_N_CP,switch_graph);
```

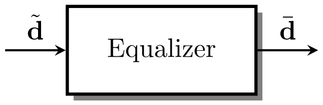
z̃
OFDM Demodulation
d̃

This block facilitates OFDM demodulation, i.e. FFT and cyclic prefix removal.

**Requirements:**

1. Use the same parameters as for OFDM modulation

```
d_bar=equalizer(d_tilde,switch_mod,switch_graph);
```

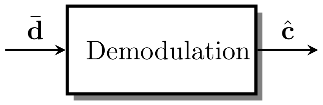


This block facilitates channel estimation using the inserted pilots and equalizes the received symbols accordingly.

**Requirements:**

1. Extract pilots and estimate channel at pilot positions
2. Equalize data

```
c_hat=demodulation(d_bar,switch_mod,switch_graph);
```

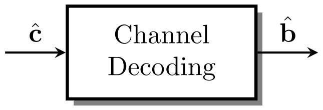


This block facilitates hard estimation of the code bits for 4-, 16- or 64-QAM.

**Requirements:**

1. Decide the received signal to 4-, 16- or 64-QAM symbols with Gray mapping to estimate the code bits.
2. Ensure proper processing in terms of the channel encoded blocks afterwards.

```
b_hat=channel_decoding(c_hat,par_H,par_N_zeros,switch_off);
```



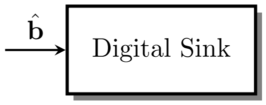
This block facilitates Channel Decoding of a [7,4] Hamming code.

**Requirements:**

1. Correct errors in the estimated code words of the [7,4] Hamming block code by syndrome decoding.

Note: Do not forget to remove the additional zeros potentially inserted at the encoder.

```
[BER]=digital_sink(b,b_hat,...);
```



Processing of the reconstructed and original signal to analyze the errors due to transmission. Here, the digital sink represents the analysis of the received and reconstructed signals. Knowledge of all other signals in the system is implicitly assumed.

**Requirements:**

1. Calculate the error in terms of the coded and uncoded bit error rate (BER).

# 3 Evaluation Guidelines

## 3.1 General Rules

Besides the solution of the task that is detailed below, we expect you to adhere to some general rules:

- Solve the tasks by yourselves.
- Write your own code and do not copy!
- Design your own slides and do not copy (pictures, too)!

Group efforts in solving the sub tasks are encouraged and expected, but we will collect the solutions of all groups at the end of the lab and test your personal knowledge about your solution. The goal of this lab is to enhance your ability to break down bigger tasks into smaller steps, organize your work and research for yourself. If you just copy the solution of other groups, you will simply limit your own benefit.

## 3.2 What to expect from the tutors?

The tutors will help you understand the tasks, may give you help finding the right information and evaluate your work to judge if you have passed or not.

Most importantly:

- Tutors will ***not*** write Matlab code for you!
- Tutors will give you hints and tips to help you to *find the solution yourself*!
- Tutors will only help you if you *follow the guidelines* and API descriptions given in this document!

## 3.3 Required Performance

To pass ICT lab 2 the following expectations have to be met. Except the compliance test, which is a hard measure checked in Matlab, compliance is rated by the tutor:

- Compliance with the tests described in Section 3.4.1 is mandatory to pass the lab.
- We expect you to write clean **and** well documented Matlab code that is easily readable by the tutor. Consider this lab to be part of a job that will be carried on by another team after you finish.
- Additionally to the compliance test, the tutor may ask you questions about your implementation to test your individual grasp of the solution.

- A short presentation about parts of the OFDM transmitter/receiver in front of all other groups of maximum 5 slides taking 5-7 minutes is expected. Therein, you should quickly explain: (1) the problem, (2) your approach, (3) the solution and (4) the final results.

## 3.4 Compliance Tests

### 3.4.1 Phase 1:OFDM Compliance Test

The compliance test at the end of phase 1 requires the following checks for transmitter and receiver:

#### OFDM Transmitter

The compliance test for the OFDM transmitter comprises the following checks:

**Digital Source**

1. Generation of a mean free binary sequence of equally probable zeros and ones
2. Figure showing the binary pattern of one frame

**Hamming Coding**

1. Correct channel encoding with generator matrix

**Modulation**

1. Correct modulation with Grey mapping
2. Figure of the modulated symbols of one block

**Pilot Insertion**

1. Framing of serial data stream into blocks of FFT length
2. Insertion of one block of known pilots for channel estimation

**OFDM Modulation**

1. Correct choice of parameters to alleviate the effects of the effective channel impulse response in the oversampled domain.
2. Pilot insertion
3. Figure of one OFDM symbol in time and frequency

**Tx Filter**

1. Bandwidth of filter output signal and side lobe suppression
2. Figure of the filter output

**Non-linear Hardware**

1. Correct clipping characteristic
2. Figure of (non-)clipped signal / Show that (no) clipping is in effect

#### OFDM Receiver

The compliance test for the OFDM receiver comprises the following checks:

**Non-linear Hardware**

1. Correct clipping characteristic

| | |
|---|---|
| | 2. Figure of received signal and signal after hardware showing that no clipping is in effect |
| **Rx Filter** | 1. Figure of the filter output<br>2. Figure showing eye pattern |
| **OFDM Demodulation** | 1. Correct OFDM demodulation and CP removal<br>2. Figure showing symbol space |
| **Equalization** | 1. Channel estimation using inserted pilots<br>2. Equalization of all OFDM channels<br>3. Figure showing equalized symbols |
| **Demodulation** | 1. Correct demodulation with Grey mapping<br>2. Figure of the estimated symbols with decision thresholds |
| **Hamming Decoding** | 1. Correct channel decoding by syndrome decoding<br>2. Figure of exemplary code word indicating corrected errors |
| **Digital Sink** | 1. Figure showing binary estimate and original signal indicating erroneous positions.<br>2. Figure showing BER |