FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Digital Communications

Lab Course

Erlangen – February 2023

Contents

Course Information **1**
0.1 General Information 1
0.2 Lab Guidelines 2

1 Digital Transmission of Data **5**
1.1 Introduction, Background, and Motivation 5
1.2 Purpose 6
1.3 Lab Environment 6
1.3.1 Transmitter 6
1.3.2 Receiver 8
1.4 Homework 13
1.5 Lab Exercises 15
1.5.1 Signal Generation at the Transmitter 15
1.5.2 Coherent Receivers — Transmission over AWGN Channel 18

2 Implementation of Transmitter and Receiver in MATLAB **20**
2.1 Introduction, Background, and Motivation 21
2.2 Purpose 22
2.3 Lab Environment 22
2.3.1 Oversampling factor 23
2.3.2 Transmitter 23
2.3.3 Channel 24
2.3.4 Receiver 25
2.4 Homework and Lab Exercises 26
2.4.1 Transmitter 26
2.4.2 Channel 29
2.4.3 Receiver 30
2.4.4 BER calculation 33

3 OFDM 36
3.1 Introduction, Background, and Motivation 37
3.1.1 Orthogonal Frequency-Division Multiplexing 37
3.2 Lab Environment 39
3.3 Lab Exercises 40
3.3.1 OFDM Transmitter 41
3.3.2 OFDM Receiver 42

4 Signal Space Representation 44
4.1 Introduction 45
4.2 Lab Environment 45
4.3 Signal Space Representation 46
4.3.1 Orthogonality 47
4.3.2 Orthogonalization 48
4.4 Homework and Lab Exercises 52
4.4.1 Information Transmission Using Signal Elements 52
4.4.2 Gram-Schmidt Procedure 56
4.4.3 Frequency Shift Keying 58

5 Signal Processing in MIMO Systems 59
5.1 Introduction, Background, and Motivation 59
5.2 Lab Environment 61
5.3 Homework and Lab Exercises 62
5.3.1 Signal Constellations 62
5.3.2 Channel Models 63
5.3.3 SISO 64
5.3.4 SIMO 65
5.3.5 MIMO 67

Course Information

0.1 General Information

Venue	Room lab room 06.018, Cauerstraße 7, 91058 Erlangen
Times	Morning Group: 09:00 - 14:00
Dates	Project #1: 2023/02/27 Project #2: 2023/02/28 Project #3: 2023/03/01 Project #4: 2023/03/02 Project #5: 2023/03/03
Prerequisites	Digital Communications / Digitale Übertragung
Credits	2.5 ECTS credits (= work load of 75 hours time)
Reference	Johannes Huber, Robert Schober, *Digital Communications*, lecture notes
Lab work	5 lab exercises
Homework	5 homework sets due in class before lab work
Grading	on passed/not passed basis
Contact	Dr.-Ing. Clemens Stierstorfer `clemens.stierstorfer@fau.de`

0.2 Lab Guidelines

Homework

Students are required to come prepared with the pre-lab reading work. In addition, each of the five projects has a set of homework problems marked by an **H**. For example in **H-0.1** your first homework is given.

Homework H-0.1

> Carefully read the instructions for project #1 and solve all the homework problems marked by **H-1.X**!

Homework assignments *(one per pair!)* should be uploaded to StudOn no later than the beginning of the assigned time slot (08:00 or 14:00). Your assignments should be written neatly, should contain clear and complete solutions.

In the Lab

There are five projects in this course. Students are required to complete each project during their assigned time slot (length: approx. 5 hours). For the first project you solve the tasks using a Matlab GUI to control our hardware in the lab room. For the projects 2 to 5 everything has to be solved by programming in MATLAB.

The work in the lab is to solve the problems marked by an **L** in this manual. An example lab exercise is shown in **L-0.1**.

Lab Exercise L-0.1

> In the first project solve all the lab exercises marked by **L-1.X** (with **X**$\in \mathbb{N}$).

Documentation

Make sure you document your work in the lab! For the first project you might just write down your results and take screenshots of the oscilloscope. In the remaining projects clearly organize and save your MATLAB code.

Tip-0.0: Documenting your Work on StudOn

> Please upload your (home)work to StudOn! You can scan your solution sheets or take photos. Preferred data format is pdf. MATLAB code, i. e., the m-files can be uploaded directly to StudOn.
>
> The uploads of the homework sets are due on the time your projects start (08:00 or 14:00), your lab work documentations are due on the respective next day (again 08:00 or 14:00).

Grading policy

Grading is done on a passed/not passed basis. In order to pass, at least fair results in all homework sets and lab exercises have to be achieved. Furthermore all mandatory quizzes on StudOn must be solved sucessfully. The Matlab code of the láb exercises has to be an unique solution by every team and must be presented to a supervisor before leaving the lab. If a group cannot explain their code or it is a copy of another group, the whole lab course will be graded as not passed for this group(s).

In Case of Illness

If you are ill, please let us know before the experiment. Possibly, an alternative appointment or the participation via ZOOM can be arranged.

Starting the lab environment

Inside the lab room: Please start MATLAB using the shell script `./start_dicoLab.sh`. It is then guaranteed that all paths are set correctly. Some files for projects 2 to 5 can be downloaded from StudOn.

Project 1

Digital Transmission of Data

1.1 Introduction, Background, and Motivation

In this first experiment you will study the basics of digital data transmission. Today, digital transmission techniques are widely used in many common applications, e. g. mobile telephony, digital video broadcasting etc. Advantages of digital transmission techniques over analog schemes like amplitude modulation and frequency modulation are a much better power efficiency and a tremendous flexibility. The focus of this experiment is on the most popular technique for digital data transmission, so-called pulse amplitude modulation (PAM).

The experiment consists of three parts: first, the structure of a generic digital PAM transmitter is analyzed. Second, the receiver of such a system is studied. The latter requires to establish an entire digital transmission system consisting of transmitter, channel, and receiver. Detailed descriptions and discussions of the optimal design of digital PAM transmission systems are part of any textbook on digital communications, e.g. [Pro00, Hay00]. Here, in the context of this lab course, we use the system model and the notation introduced in the lecture notes to “Digital Communications” [HS22] or [HS23]. The last part of this chapter is a short introduction of modified PAM modulation methods.

1.2 Purpose

The aim of this experiment is to demonstrate the effectiveness and efficiency of digital transmission schemes. You will learn how to design a transmitter for digital pulse amplitude transmission and how to represent the digital data by complex coefficients. You will understand the design rules and constraints for the basic pulse shape in the time domain and the frequency domain and will know how to develop the optimal receiver for digital transmission systems. You will get to know the effects of some transmission disturbances on the received signal. Finally, you will understand some modified concepts, improving standard PAM transmission.

1.3 Lab Environment

In this experiment you work with a MATLAB-based simulation environment which can model several digital transmission techniques. The lab computers are equipped with powerful analog-to-digital converters (ADC) and digital-to-analog converters (DAC). In the first part of the experiment, just the DACs are used to generate physical signals from the software-based digital transmission model. That is, the digital transmission schemes selected and simulated in the software are analyzed based on "real" voltages and currents.

In the second part, two of the lab computers are combined into a complete digital transmission system comprising transmitter, channel, and receiver. One of the computers acts as the transmitter, i. e., the MATLAB-generated signals are D/A converted and transmitted to the receiver using patch cables (coaxial cables with BNC connectors). The other computer acts as the receiver; the physical signal is A/D converted and then again fed into a MATLAB-based receiver structure. The (AWGN) channel is modeled by an electronic noise generator.

In the third part, the systems are again just used to generate the transmit signals of modified PAM transmission.

1.3.1 Transmitter

In order to simulate the transmitter of a digital transmission system the respective MATLAB program has to be started.

Then, you should see a MATLAB GUI as depicted in Fig. 1.3.

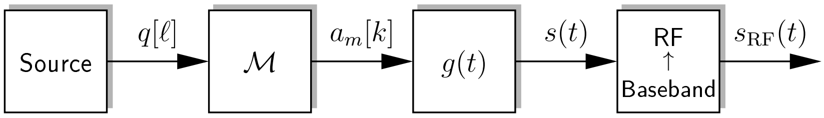


Figure 1.1: Block diagram of transmitter structure of a digital transmission system.

The transmitter is built from several individual blocks as depicted in Fig. 1.1. These blocks are subsequently described in some more detail.

1.3.1.1 Data Source

The data source emits a pseudo-random sequence of binary symbols $q[\ell] \in \{0, 1\}$. Zeros and ones occur with equal probabilities.

1.3.1.2 Binary Labeling (Bit Mapping)

As we focus on digital PAM, the binary labeling in block $\mathcal{M}$ assigns binary m-tuples to (complex) amplitude coefficients $a_{m[k]}$.

The system supports several different sets of amplitude coefficients, i. e., signal constellations. In Table 1.1 a list of the implemented signal constellations and constellation sizes is given.

Table 1.1: Supported signal constellations

	Size of Constellation					
Type	2	4	8	16	32	64
ASK (unipolar)	×	×	×	×	×	×
ASK (bipolar)	×	×	×	×	×	×
PSK	×	×	×	×	×	×
QAM		×		×		×
STAR				×		
CROSS					×	

1.3.1.3 Pulse Shaping

The transition from the discrete-time domain signal to the (usually still complex-valued) continuous-time domain signal (complex baseband) in a digital transmission system model is performed by the pulse shaping. In PAM transmission the basic pulse shape $g(t)$ simply is weighted with the (complex) amplitude coefficients $a_{m[k]}$. The (spectral) shape of the pulse significantly influences the spectrum of the RF signal; the waveform of the pulse shape in time-domain domain is crucial for a transmission not suffering from inherent interference.

In the simulation environment used in this experiment, the pulse shaping is still located in the digital domain. That is, the continuous-time signal is simulated by an over-sampled discrete time signal.

The equivalent complex baseband signal can be seen at the output of the D/A converters if the respective check box is set, i. e., no RF modulation is used. The inphase component is at channel A, the quadrature component at channel B.

1.3.1.4 RF Modulation

The radio-frequency signal is generated from the baseband signal by multiplying with sinusoidal waveforms with the carrier frequency f_c. This step is also implemented in MATLAB. The physical signal which can be then analyzed at channel A of the D/A converter is the result of a simple D/A conversion of an entirely digitally modeled block diagram. The second output at channel B is a synchronization signal required for the combination of two computers into an entire transmission system.

1.3.2 Receiver

Starting the receiver shows you a MATLAB GUI as depicted in Fig. 1.4. Please make sure, that you do not run the transmitter and the receiver in parallel on one computer using two instances of MATLAB. This will lead to severe conflicts with the A/D and D/A cards.

The incoming signal at the computer acting as receiver is first A/D converted and then processed in an entirely MATLAB-based receiver structure which is sketched in Fig. 1.2. The individual parts of this block diagram are briefly explained below.

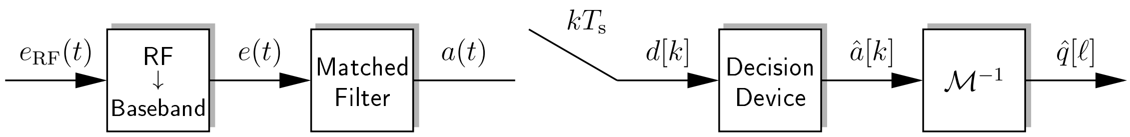


Figure 1.2: Block diagram of receiver structure of a digital transmission system.

1.3.2.1 RF/Baseband

First, the incoming noisy radio-frequency signal $e_{\mathrm{RF}}(t)$ is demodulated, i. e., a baseband signal $e(t)$ is generated by multiplying with the carrier and low-pass filtering.

1.3.2.2 Filtering

In the baseband, the next step at the receiver is a (matched) filter to reduce the noise and optimize the SNR for the sampling. The filters impulse response can be chosen from a given selection of pulse shapes.

1.3.2.3 Sampling

After the input filter, the pseudo-continuous signal $a(t)$ is sampled in order to obtain a stream of discrete symbols $d[k]$ on the symbol raster of T_{s}.

1.3.2.4 Decision Device

The time-discrete samples $d[k]$ are then fed into the decision device to obtain estimates $\hat{a}[k]$ on the initial coefficients $a_{m[k]}$.

1.3.2.5 Demapping

The last step is to retrieve the binary data stream from the estimates $\hat{a}[k]$. $\mathcal{M}^{-1}$ is the inverse operation to $\mathcal{M}$ and yields the estimated binary data symbols $\hat{q}[\ell]$.

1.3.2.6 Disturbed Received Signals

At the receiver, we can model the effects of several problems which are likely to occur on the channel. The impact of frequency offsets, phase offset, and mismatched sampling instances can be tested on the received signal.

1.3.2.7 Error Counters

— not active —

Readings for Lab 1

[HS23] Johannes B. Huber and Robert Schober, *Digital Communications*, Lecture Notes, Erlangen, October 2022.

[HS22] ———, *Digitale Übertragung*, Lecture Notes, Erlangen, April 2022.

[Kam08] Karl-Dirk Kammeyer, *Nachrichtenübertragung*, 4 ed., B. G. Teubner, Stuttgart, March 2008.

[Pro00] John G. Proakis, *Digital communications.*, 4th ed., McGraw-Hill, New York, NY, USA, 2000.

[Hay00] Simon Haykin, *Communication systems*, 4th ed., John Wiley & Sons, New York, NY, USA, 2000.

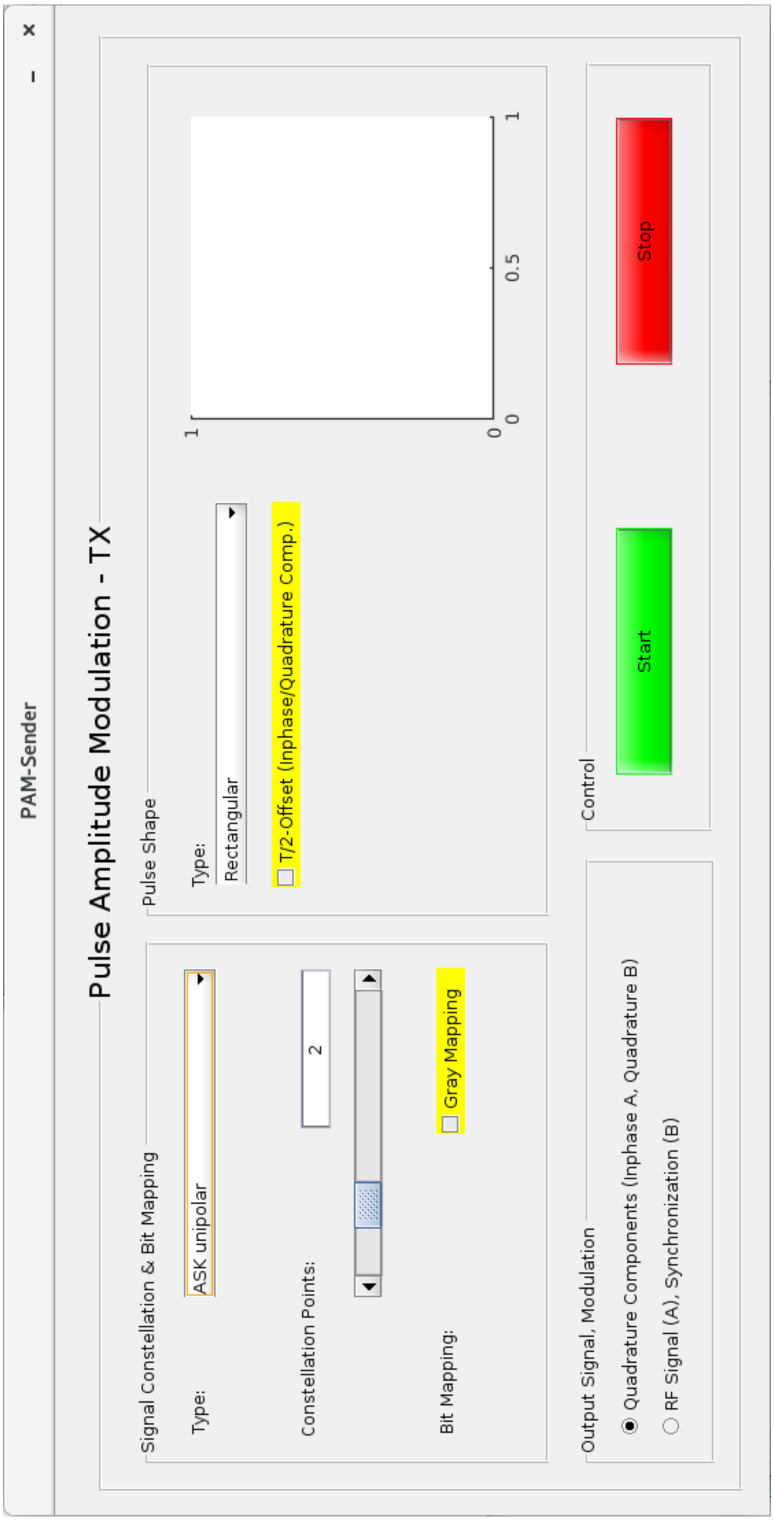


Figure 1.3: MATLAB GUI of PAM transmitter.

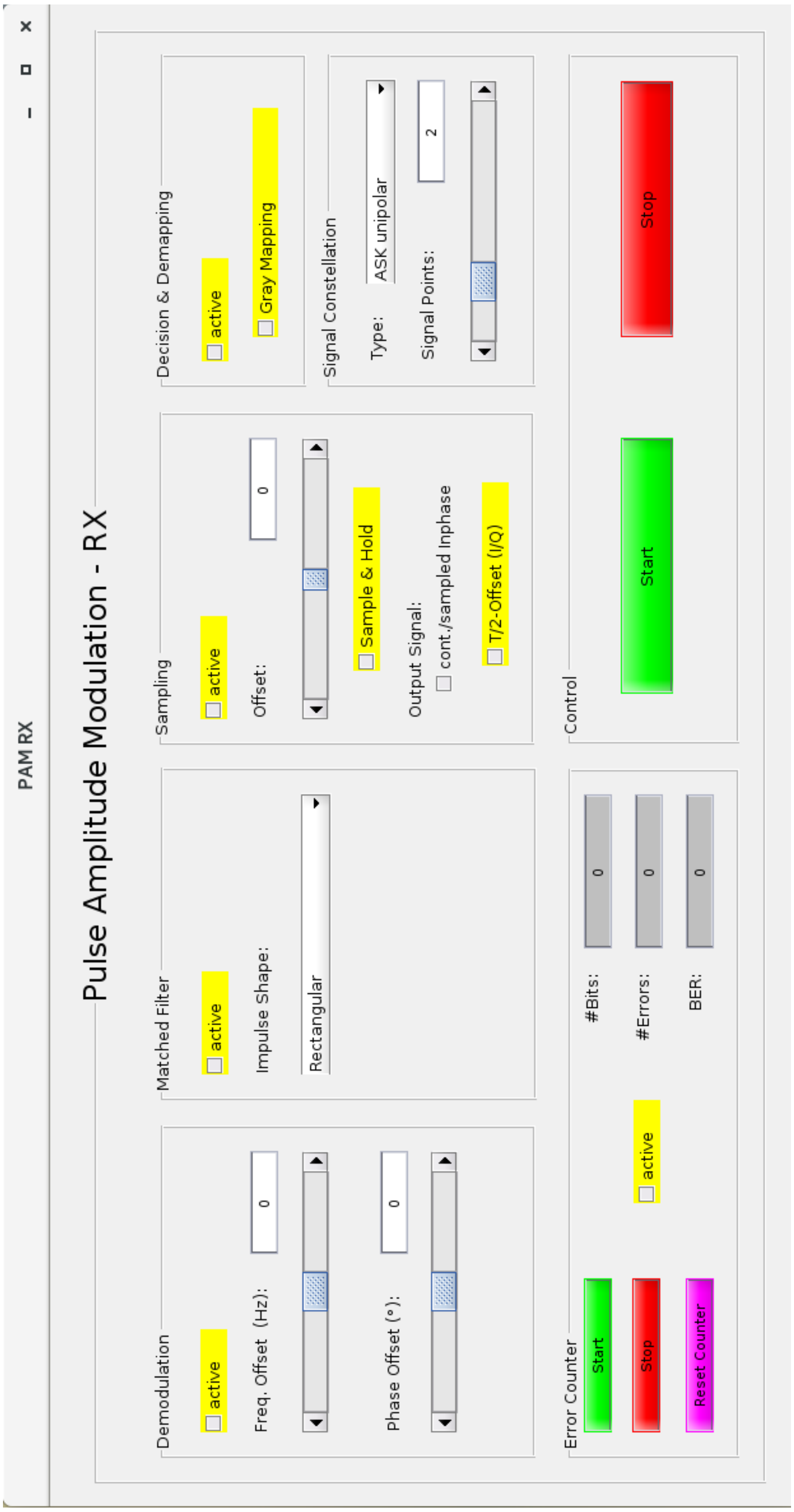


Figure 1.4: MATLAB GUI of PAM receiver.

1.4 Homework

Homework H-1.1

Plot a 4-QAM signal constellation. How many bits can be represented by one signal point? Label the signal points with binary labels (words) using a Gray code.

Homework H-1.2

What is the so-called *X-Y mode* of an oscilloscope?

Homework H-1.3

The *peak-to-average power ratio* of a signal constellation is defined as

$$\text{PAPR} = \frac{\max_m\{P_m\}}{\mathrm{E}\{P_m\}}\,,$$

P_m ($m = 1, \ldots, M$) being the power of the m-th signal point.

Calculate (by hand and analytically) the PAPR of the equivalent complex baseband signals for the following signal constellations: 2-ASK (unipolar/bipolar), 8-ASK (unipolar/bipolar), 16-QAM, and 8-PSK.

How are the PAPR and the so-called *crest factor* ($\rightarrow$ Google) related?

Hint: Use the graphical representation of the signal constellations in the ECB to determine the PAPR. The squared distance of a signal point from the origin is proportional to its power P_m.

Homework H-1.4

How are the spectra of the radio-frequency and the baseband signal related?

Homework H-1.5

How is the optimal input filter at the receiver called? What is the optimization criterion in its derivation?

The combination of the basic pulse shape at the transmitter and the receive filter has to fulfill the so-called Nyquist criterion. Give the definitions of this criterion in both time and frequency domain and sketch them for an exemplary pulse shape.

Homework H-1.6

Why is it helpful to minimize the variation of the envelope of the transmit signal?

1.5 Lab Exercises

1.5.1 Signal Generation at the Transmitter

The generation of the pulse-amplitude-modulated signal at the transmitter side is analyzed. First, the focus is on the (equivalent complex) baseband signal, i. e., the radio-frequency signal is not considered. We will have a look at both, time-domain and frequency-domain properties of the signal. The RF signal is studied at the end of this section.

1.5.1.1 Signal Constellations (at the Transmitter)

In order to illustrate and analyze the properties of the signal constellations, a “rectangular pulse shape” has to be selected. The filter switch at the output of the connector boxes has to be in position “0” (no filter is applied to the generated signal). The selection for the output signal in the GUI has to be quadrature components; both outputs A and B have to be connected to the oscilloscope.

Lab Exercise L-1.1

Select a bipolar ASK signal constellation from the drop-down list with 2 signal points. Display the resulting signals on the oscilloscope. Determine the symbol rate ($1/T_\mathrm{S}$) and the data rate ($1/T_\mathrm{b}$, uncoded transmission) of the system for these parameters!

Increase the number of signal points to 4, 8, and 16 and again compute $1/T_\mathrm{S}$. What can you tell about the data rate ($1/T_\mathrm{b}$) in these cases (again uncoded transmission)?

Lab Exercise L-1.2

Now select a QAM signal constellation from the drop-down list. Start with 4 signal points and again determine the symbol rate ($1/T_\mathrm{S}$) and the data rate ($1/T_\mathrm{b}$, uncoded transmission)! Compare your results to **L-1.1**.

Increase the number of signal points to 16 and again compute $1/T_\mathrm{S}$. What can you tell about the data rate ($1/T_\mathrm{b}$, uncoded transmission)?

Lab Exercise L-1.3

Again select the 4-QAM signal constellation in the GUI and then switch the oscilloscope in the X-Y mode. Adjust the intensity of the signal graph such that a clear signal constellation is visible. Explain the relation between the signals over time and the signal constellation.

Display also some other signal constellations from Tab. 1.1 on the oscilloscope!

Lab Exercise L-1.4

Verify your analytical results from **H-1.3**, i. e., measure the PAPR of the physical signals of the signal constellations listed in **H-1.3**.

Note, you cannot directly measure the powers! You can measure voltages (peak, amplitudes, top etc.) using the oscilloscope and you can measure RMS voltages using either the Agilent multimeter (prefered) or the oscilloscope. However, a measurement always refers to just one of the two quadrature components.

Lab Exercise L-1.5

Compare the results for unipolar and bipolar ASK. How do you rate the two techniques in terms of the PAPR? Give one advantage for both variants of ASK.

1.5.1.2 Pulse Shaping

The properties of the transmit signal are–apart from the signal constellation–mainly determined by the employed basic pulse shape $g(t)$. In the following, we study several basic pulse shapes with respect to their suitability to be used in digital transmission scheme. In particular, the spectra of the pulses are analyzed (baseband transmission).

The filter switch at the connector boxes has to be set to position “1”, i. e., the generated output signals (inphase and quadrature component) are smoothed with a low pass filter.

Lab Exercise L-1.6

First, use a unipolar 2-ASK signal constellation. Study the time-domain inphase component of the equivalent complex baseband signal for a rectangular pulse shape and a cosine-roll-off pulse shape, respectively.

Repeat these steps for a bipolar 2-ASK, a 4-QAM, a bipolar 4-ASK, and a 16-QAM signal constellation.

Lab Exercise L-1.7

Compare the spectra of the ECB signals of a bipolar 2-ASK using a rectangular pulse shape and a cosine-roll-off pulse shape, respectively.

Here, the effect of the smoothing filter can be clearly seen. Put the switch into both positions "0" and "1" and compare the results.

1.5.1.3 The Radio-Frequency Signal

After the weighting of the basic pulse shape $g(t)$ with the channel coefficient $a_{m[k]}$, the baseband signal has to be transformed into a radio frequency signal centered around the carrier frequency f_c.

Lab Exercise L-1.8

Display and study the radio frequency signal of the following signal constellations in both, time domain and frequency domain: 2-ASK (unipolar/bipolar), 4-QAM, and 16-QAM.

First use a rectangular pulse shape and then a cosine-roll-off pulse shape.

Lab Exercise L-1.9

Verify your results from **H-1.4**.

Lab Exercise L-1.10

How does the size of the signal constellation M affect the spectra of the radio signals (Note, the data rate $1/T_\mathrm{b}$ is constant per quadrature component.)?

1.5.2 Coherent Receivers — Transmission over AWGN Channel

For the next experiments two of the lab systems have to be combined. One system acts as the transmitter, the other one acts as the receiver in a transmission scenario. An illustration for the practical implementation of this combination is given in Fig. 1.5. The low pass filters of the connector boxes should be active in the RF part (TX and RX) and inactive at the output of the receiver to the oscilloscope.

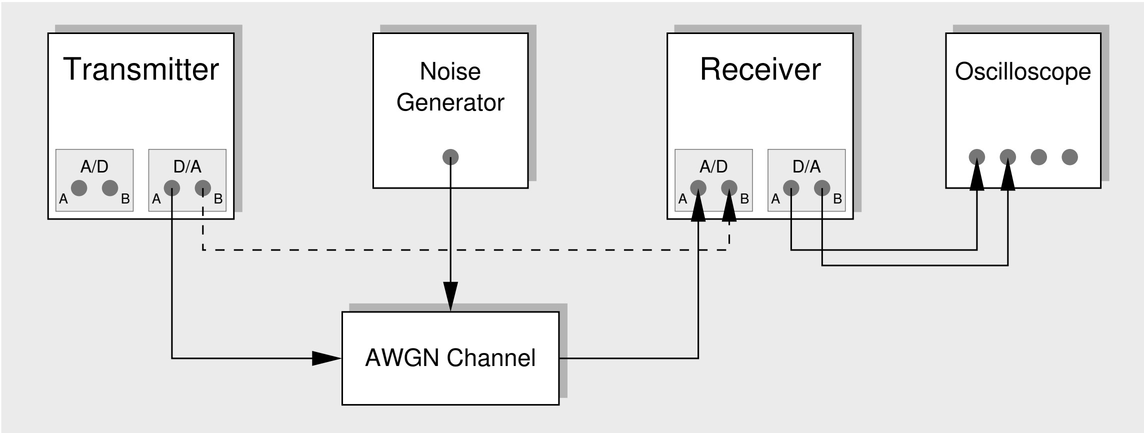


Figure 1.5: Transmission system using two of the lab computers.

In the following section, we finally want to study the signal constellations at the receiver side. This goal needs some preliminary steps which are described in the following.

Select a unipolar 2-ASK signal constellation and the $\sqrt{\cos}$ pulse with a roll-off factor of 0.94 at the transmitter. Study the inphase and the quadrature component of the demodulated (from RF) time-domain signal at the receiver.

Lab Exercise L-1.11

Describe the effect you can observe regarding the amplitude and the phase of the signal. How can you compensate for these effects?

Lab Exercise L-1.12

Activate the appropriate matched filter and study the resulting eye patterns. Identify the best time instance for sampling.

Lab Exercise L-1.13

Now activate the sample and hold check-box and describe the resulting signal. What happens if you vary the sampling time instance?

Lab Exercise L-1.14

Use the oscilloscope (in X-Y mode) to display the phasors of the *sampled* received signal for 2-ASK (unipolar/bipolar), 4-QAM, 16-QAM, 4-PSK, and 8-PSK.

Lab Exercise L-1.15

How do frequency offsets and phase offsets affect the signal constellation? Use a unipolar 2-ASK and a 4-PSK signal constellation and vary the respective parameters in the receiver GUI.

Lab Exercise L-1.16

How does a mismatched sampling affect the signal constellation at the receiver? Use again a 4-QAM signal constellation to study the effects.

Lab Exercise L-1.17

How does the additive white Gaussian noise affect the signal constellations? Use a 4-QAM constellation and varying noise powers to demonstrate the effect.

1.5.2.1 Measuring the Bit Error Ratio

— skipped —

Project 2

Implementation of Transmitter and Receiver in MATLAB

2.1 Introduction, Background, and Motivation

In this experiment, a complete digital communication system shall be implemented, where the whole system shall work on one computer using MATLAB without any additional hardware components. The script data '*./matlab_for_lab2/simulation.m*' will be the framework of this simulation, where all the following MATLAB functions will be called from. The missing MATLAB functions for transmitter, channel, and receiver shall be implemented step-by-step by the students. In order to evaluate the performance of the simulation setup, a function calculating the bit error rate (BER) shall be implemented. At the end of this project, the BER-$E_\mathrm{b}/\mathcal{N}_0$-curves of several transmission settings shall be visualized and compared to the well-known curves from the literature (e. g. [Pro00, HS23]).

2.2 Purpose

The aim of this experiment is to get more into detail of a digital communication system. By building the code of the different steps of transmitter, channel and receiver, the students will get a better insight into their functionalities.

2.3 Lab Environment

```
%% Simulation Parameters
    PAM_type = '8ASKbipolar'; %'BPSK'; '4QAM';

    M = 8; %2; %4
    GrayMappingOn = 0; % 0 = off, 1 = on
    EbN0_dB = 10;
    numberOfBits = 12e4;
    f_b = 1e3; % symbol/baud rate
    oversamplingFactor = 4; % should be at least 4
    f_c = 5e3; % carrier frequency; should be ca. 5 times f_s
    f_s = f_c*oversamplingFactor;

%% Creation of a random bit stream
    traBits = round(rand(1,numberOfBits));

%% Transmitter
    traSignal = transmitter(traBits, PAM_type, ...
    GrayMappingOn, f_b, f_c, f_s);

%% Channel
    recSignal = channel(traSignal, EbN0_dB, M, f_s, f_b);

%% Receiver
    recBits = receiver(recSignal, PAM_type, ...
    GrayMappingOn, f_b, f_c, f_s);

%% Calculate BER
    BER = calculateBER(traBits, recBits)
```

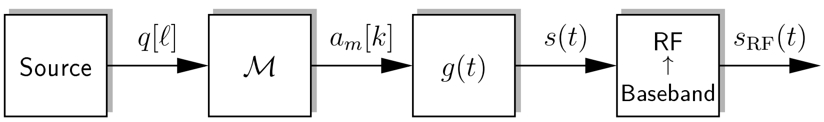


Figure 2.1: transmitter structure of a digital transmission system.

The code of the simulation script ('*./matlab_for_lab2/simulation.m*') is listed in this section. The code starts with setting the parameters. Especially the parameter *oversamplingFactor* is important. It will be described in Sec. 2.3.1. Then, a random bit vector is created with the bit length set before. Subsequently the function calls of transmitter, channel and receiver are listed. The last part calculates the BER.

2.3.1 Oversampling factor

Transmitting data on a physical channel means to make use of D/A and A/D converter. So, the signals are not only time discrete but also time continuous. MATLAB works on vectors and matrices and cannot handle time continous signals. Even so, simulating a time continous signal is also possible in MATLAB: The continuous signal is simply approximated by working with an oversampled signal. By default, the parameter *oversamplingFactor* is set to 4.

Remark: In the following, the time continous signal is described as a function of the time continous value t, i.e. $s_{\mathrm{RF}}(t)$. In the context of MATLAB the signal should be a function of a time discrete value, i.e. $s_{\mathrm{RF}}[t]$. Using a *oversamplingFactor* of 4 means, that the system sampling rate f_{s} is $4 \cdot f_{\mathrm{c}}$. Thus, the values $s_{\mathrm{RF}}[t]$ can be seen as sampled values at sampling rate f_{s}.

2.3.2 Transmitter

The block diagram of the transmitter is the same as in experiment 1, see Fig. 2.1. Please have a closer look at the sections describing the transmitter units in experiment 1. In the following, *Bit Mapping* using Gray Mapping will be described in detail.

The job of *Bit Mapping* (block $\mathcal{M}$ in Fig. 2.1) is to map a M binary m-tuple to (complex) amplitude coefficients $a_m[k]$. This can be done in a natural way or in a more sophisticated

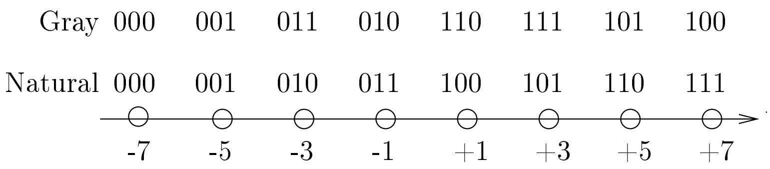
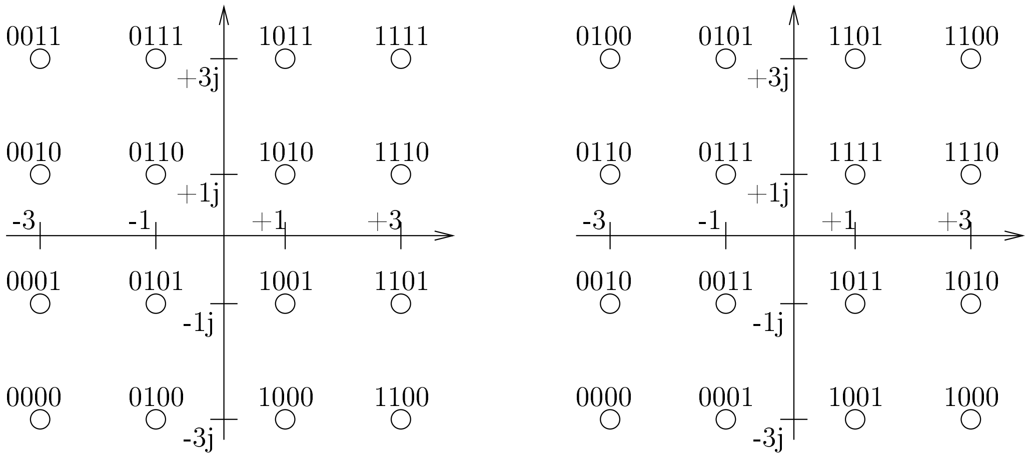


Figure 2.2: Constellation diagram for bipolar 8-ASK



(a) Natural Mapping

(b) Gray Code

Figure 2.3: Constellation diagram for 16-QAM

way: the Gray Code. The resulting constellation diagrams are shown in Fig. 2.2 for (bipolar) 8-ASK and in Fig. 2.3 for 16-QAM.

If a symbol was detected at receiver-side which was not the transmitted one, the probability of detecting the neighbor of the transmitted symbol is much higher than detecting the others. By using the Natural Mapping neighboring signal points might differ in more than one bit value. This is not the case by using the Gray Code. There, detecting the direct neighbor yields to only one bit error in the whole m tuple.

2.3.3 Channel

The channel is kept simple in this experiment. The signal is just degraded by additive white Gaussian noise (AWGN) which has a constant power spectral density with the two-sided power spetral density denoted as $\mathcal{N}_0/2$. So, the block diagram is also simple, see Fig. 2.4. Adding the noise can be described with:

$$e_{\mathrm{RF}}(t) = s_{\mathrm{RF}}(t) + n(t) \tag{2.1}$$

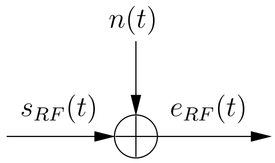


Figure 2.4: AWGN channel

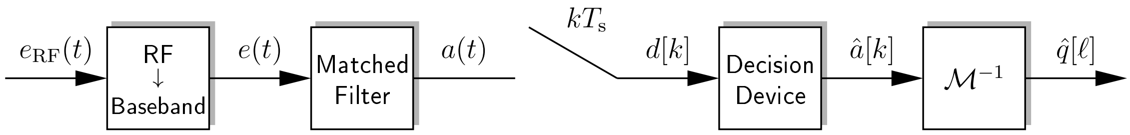


Figure 2.5: receiver structure of a digital transmission system.

2.3.4 Receiver

In a consistent way, the block diagram of the receiver is also the same as in experiment 1, see Fig. 2.5. Please have a closer look at the sections describing the receiver units in experiment 1.

Readings for Lab 2

[HS23] Johannes B. Huber and Robert Schober, *Digital Communications*, Lecture Notes, Erlangen, October 2022.

[HS22] ______, *Digitale Übertragung*, Lecture Notes, Erlangen, April 2022.

[Kam08] Karl-Dirk Kammeyer, *Nachrichtenübertragung*, 4 ed., B. G. Teubner, Stuttgart, March 2008.

[Pro00] John G. Proakis, *Digital communications.*, 4th ed., McGraw-Hill, New York, NY, USA, 2000.

[Hay00] Simon Haykin, *Communication systems*, 4th ed., John Wiley & Sons, New York, NY, USA, 2000.

2.4 Homework and Lab Exercises

2.4.1 Transmitter

2.4.1.1 Bit Mapping

Homework H-2.1

The bit stream 11 00 10 00 11 11 is given. Map these bits onto complex PAM constellation symbols using bipolar 8-ASK and 16-QAM, first with Natural Mapping, then using Gray Code (see Fig. 2.2 and 2.3).

Homework H-2.2

Create the constellation diagram for 4-QAM for both, natural mapping and Gray mapping equivalent to Fig. 2.3. Finally, map the bitstream 11 00 10 00 11 11 to the corresponding 4-QAM symbols.

Homework H-2.3

Depending on the number of elements inside a bitstream, it might happen that there are some bits remaining, which do not represent a valid signal constellation point. For example 01 010 can not be mapped fully onto 4-QAM, 8-ASK nor 16-QAM. To cover those cases, the following tasks require Zero-Padding of the bitstream if necessary. Briefly give an example of the procedure using the bitstream 01 010 and 4-QAM, 8-ASK and 16-QAM!

Lab Exercise L-2.1

Edit the MATLAB file *mapBitsToSymbols.m* in the working folder. The function has the following head:

```
function PAM_symbols = mapBitsToSymbols(bitvector, PAM_type)
```

Write the code which maps the bits inside the vector `bitvector` onto complex symbols according to input string `PAM_type` using Natural Mapping. Possible values for the input variable `PAM_type` can be 'BPSK', '8ASKbipolar' and '4QAM' .

Add the function call into the MATLAB file *transmitter.m*.

Lab Exercise L-2.2

At the receiver side, the incoming symbols might be affected by noise. Edit the Matlab file *mapSymbolsToBits.m* in the working folder with the following head:

```
function bitvector = mapSymbolsToBits(PAM_symbols, PAM_type)
```

Write the code which first maps the estimated PAM points $\hat{a}[k]$ to the nearest possibly transmitted point and second map these values to bits again according to input string `PAM_type` and with Natural Mapping. Possible values for the `PAM_type` can be 'BPSK', '4QAM' and '8ASKbipolar' .

Lab Exercise L-2.3

Edit the MATLAB file *mapBitsToSymbols_ Gray.m* in the working folder with the following head:

```
function PAM_symbols = mapBitsToSymbols_Gray(bitvector, PAM_type)
```

Write the code equivalent to the function *mapBitsToSymbols.m*. It shall map the bits onto complex symbols, but with Gray Code. Add the call of this function to the function `transmitter`. Do not forget to implement an if-else or switch-case which selects one of both functions according to the value of `GrayMappingOn`.

Lab Exercise L-2.4

Edit the Matlab file *mapSymbolsToBits_ Gray.m* in the working folder with the following head:

```
function bitvector = mapSymbolsToBits_Gray(PAM_symbols, PAM_type)
```

Write the code equivalent to the demapping without Gray Code. It shall map the received PAM points to the nearest possibly transmitted point and map these values to bits again according to input string `PAM_type`.

Lab Exercise L-2.5

Assure that there is no error if you create a random bitstream, select the signal constellation type and call `mapBitsToSymbols_Gray` and insert the output into `mapSymbolsToBits_Gray`. Also check for the case of Natural Mapping!

2.4.1.2 Pulse Shaping

Lab Exercise L-2.6

The MATLAB file *pulseShape.m* is already in the working folder and ready to use. Add the following function call into the MATLAB file *transmitter.m*.

```
ecb_signal = pulseShape(PAM_symbols, f_b, f_s)
```

2.4.1.3 Modulation

Homework H-2.4

Write down the block diagram of the modulation of the complex PAM symbols to get the high frequency signal with carrier frequency f_c (last block in Fig. 2.1). A simple filter with an rectangular impulse answer is used for pulse shaping. Complete the following equation:

$$s_\mathrm{RF}(t) =$$

Homework H-2.5

Given the following ECB signal:

$$s(t) = \begin{cases} +3 - 5i & \text{for } 0 \leq t < T \\ -1 - 1i & \text{for } T \leq t < 2T \\ +7 + 3i & \text{for } 2T \leq t < 3T \\ 0 & \text{else} \end{cases}$$

Calculate the RF signal after modulation for $0 \leq t < 3T$

Lab Exercise L-2.7

Edit the MATLAB file *modulate.m* in the working folder with the following head:

```
function rf_signal = modulate(ecb_signal, f_c, f_s)
```

Write the code which takes the ECB signal and modulates it onto the carrier frequency f_c. Add the function call into the MATLAB file *transmitter.m*.

2.4.2 Channel

Homework H-2.6

Given a signal vector $\mathbf{x}$ in MATLAB, write a (pseudo) code calculating the mean power of this signal.

Homework H-2.7

Given the calculated mean power $\bar{S}$ of the transmit signal $s_\mathrm{RF}(t)$, the symbol rate f_b and the modulation constellation size M. How can the energy per information bit E_b be calculated?

Homework H-2.8

A MATLAB Gaussian noise vector shall be added to the analogue transmit signal given these the two-sided power spectral density $\mathcal{N}_0/2$ and the system sampling rate f_s. How is the mean power of this noise vector calculated given these parameters

---

Lab Exercise L-2.8

Edit the MATLAB file *channel.m* in the working folder with the following head

```
function recSignal = channel(traSignal, EbN0_dB, M, f_s, f_b)
```

Write the code which adds the Gaussian noise given the parameters of the function head.

2.4.3 Receiver

2.4.3.1 Demodulation

Homework H-2.9

Write down the block diagram of the demodulation of the received signal to get the complex baseband signal $e(t)$ (first block in Fig. 2.5).

Lab Exercise L-2.9

Edit the Matlab file *demodulate.m* in the working folder with the following head:

```
function demodSignal = demodulate(recSignal, f_c, f_s)
```

Write the code which takes the RF signal and demodulates it into ECB domain. Take care that both signal parts (real and imaginary part) are taken into account. Add the call of this function inside the Matlab file *receiver.m*.

2.4.3.2 Matched Filter / Downsampling

Lab Exercise L-2.10

Add the following function call into the Matlab file *receiver.m*.

```
 filtered_signal = MatchedFilter(ecb_signal, f_b, f_s);
PAM_symbols = Downsample(filtered_signal, f_s/f_b);
```

2.4.3.3 Decision Device / Demapping

Homework H-2.10

The received PAM symbols are degraded by noise. What are the decision boundaries for 16-QAM in the AWGN case? Fill them into Fig. 2.3 using dotted lines.

In the following, the transmitted bit stream is 00 01 11 10 10 11. Consequently, the transmitted 16-QAM symbol sequence using Natural Mapping is $\{-3-1\mathrm{j}; 3+1\mathrm{j}; 1+3\mathrm{j}\}$. The transmitted 16-QAM symbol sequence with Gray Code is $\{-1-3\mathrm{j}; 3+1\mathrm{j}; 1-1\mathrm{j}\}$.

At the receiver-side, the received 16-QAM symbol sequence is degraded by additive white Gaussian noise. The complex noise part can be described by the sequence {+1.5-1.7j; 0.2+1.1j; -1.1+0.5j} which is added to the transmitted symbols.

Homework H-2.11

Calculate the estimated 16-QAM symbols $\hat{a}[k]$ for both received signal sequences!

Homework H-2.12

Demap both sequences to bit streams using the decision boundaries defined in the homework and calculate the bit error rate! What is better in this case: Natural Mapping or Gray Code? Why is it better?

Homework H-2.13

Assuming a symbol error occured for 8-ASK transmission and the signal-to-noise-ratio is rather large. How many bit errors are made in average for 8-ASK using natural and Gray mapping.

Lab Exercise L-2.11

The demapping and decision functions were already created in the earlier part of the project. Now, add an if-else-statement inside the receiver which calls the correct demapping function (*mapSymbolsToBits* or *mapSymbolsToBits_ Gray*) depending on the value of `GrayMappingOn`.

Lab Exercise L-2.12

Now run the given simulation script with the initial parameters and check also for the two other implemented constellation types. Please also adjust the parameter `M` when you apply changes. Since the value of `EbN0` is not required to be changed, the main goal of this task is to look out for matlab errors.

Lab Exercise L-2.13

In this task the the two sided power spectral density can be visualized by executing the following code:

```
 figure;
pwelch(traSignal, [], [], [], f_s, 'centered')
figure;
pwelch(recSignal, [], [], [], f_s, 'centered')
```

It is recommended to analyze for different values of the parameter `EbN0`.

2.4.4 BER calculation

Homework H-2.14

Having the transmitted bit stream and the received bit stream. How can the bit error rate be calculated?

Homework H-2.15

How many bits have to be simulated to calculate reliable value for a bit error rate of less than 10^{-6}?

Lab Exercise L-2.14

Edit the Matlab file *calculateBER.m* in the working folder with the following head:

```
function BER = calculateBER(traBits, recBits)
```

Write the code which calculates the bit error rate of the communication system.

Lab Exercise L-2.15

Run the simulation script with different parameters such as number of bits, PAM type, etc. and check if everything works as you expect.

Up to now, the simulation script is ready to run one simulation. However, to compare different settings, it is common to simulate with different $E_{\mathrm{b}}/\mathcal{N}_0$ values and save the BER in a vector. Then, BER / $E_{\mathrm{b}}/\mathcal{N}_0$ curves can be plotted for the selected settings. Therefore, the simulation script has to be adapted to the form on the following page.

```
%% LOOP PARAMETERS
EbN0_dB_Vector = 0:1:15;
BER_Vector = zeros(1,length(EbN0_dB_Vector));

%% Simulation Parameters
    PAM_type = '8ASKbipolar'; % 'BPSK'; '4QAM';

    M = 8; % 2; 4;
    GrayMappingOn = 0; % 0 = off, 1 = on
    numberOfBits = 12e4;
    f_b = 1e3; % symbol/baud rate
    oversamplingFactor = 4; % should be at least 4
    f_c = 5e3; % carrier frequency; should be ca. 5 times f_s
    f_s = f_c*oversamplingFactor;

%% Creation of a random bit stream
    traBits = round(rand(1,numberOfBits));

%% LOOP START
curLoop = 0;
for EbN0_dB = EbN0_dB_Vector
    curLoop = curLoop + 1;

%% Transmitter
    traSignal = transmitter(traBits, PAM_type, ...
    GrayMappingOn, f_b, f_c, f_s);

%% Channel
    recSignal = channel(traSignal, EbN0_dB, M, f_s, f_b);

%% Receiver
    recBits = receiver(recSignal, PAM_type, ...
    GrayMappingOn, f_b, f_c, f_s);

%% Calculate BER
    BER = calculateBER(traBits, recBits)

%% LOOP END
    BER_Vector(curLoop) = BER;
end
```

Lab Exercise L-2.16

Add the simulation loop lines in your simulation script and run the simulation for the three PAM types and the configuration with and without Gray mapping. Do not forget to change the name the specific BER vector at the end of the code for the plots.

Lab Exercise L-2.17

Plot the six BER curves in Matlab using the functions `figure`, `semilogy(x,y)` and `hold on`. You can label the figure with the function `xlabel`, `ylabel`, `legend`, `title`.

The main goal is to have two figures (either Natural or Gray mapping with three curves in each).

Lab Exercise L-2.18

Compare the visualized behaviour of the BER vs SNR of your curves with the literature (e.g. lecture notes).

Project 3

OFDM

3.1 Introduction, Background, and Motivation

In many (digital) transmission schemes the channel cannot be modeled by a simple AWGN channel but the channel is also dispersive. This means that the transfer function of the channel is not only a scalar as for the AWGN channel, but depends on the frequency. The channel introduces intersymbol interference. Orthogonal frequency-division multiplexing (OFDM) is a well-known and popular technique to cope with these frequency-selective channels.

3.1.1 Orthogonal Frequency-Division Multiplexing

For this project, we follow the description and definition of OFDM as given in [HS23]. The basic idea of OFDM can be visualized by splitting the frequency-selective transfer function of the dispersive channel into a number of quasi-constant subbands. These subchannels are then non-dispersive and can be used in parallel. They do not introduce any intersymbol interference.

The basic pulse shapes used for the parallel transmission over these subchannels have to be orthogonal in the frequency domain. At the receiver a bank of matched filters prevents interference between the individual subchannels. This general approach to OFDM is is shown in Fig. 3.1.

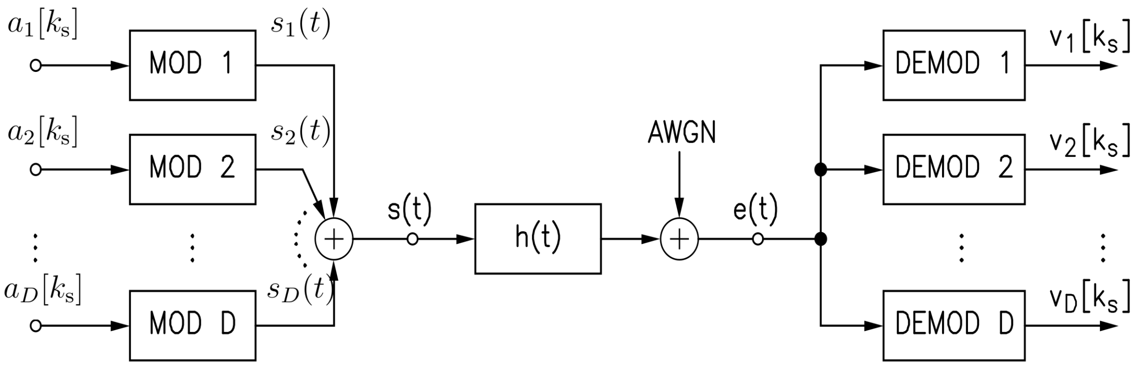


Figure 3.1: Basic principle of OFDM

For the practical implementation of an OFDM system, a different approach is preferred over the individual parallel modulators. Instead, the inverse discrete Fourier transform (IDFT) is used to transform blocks of D symbols from the frequency domain to the time domain. These symbols are then modulated by a single basic pulse shape $g(t)$. At the receiver the matched filter wrt. $g(t)$ is applied and then the block of D receive symbols is transformed again into the frequency domain using a DFT.

The individual blocks generated at the transmitter are separated by a cyclic prefix to avoid interblock interference. In Fig. 3.2 this block-based description of OFDM is illustrated. Finally, the OFDM transmission over independent parallel subchannels can be described by just using individual scaling factors per channel.

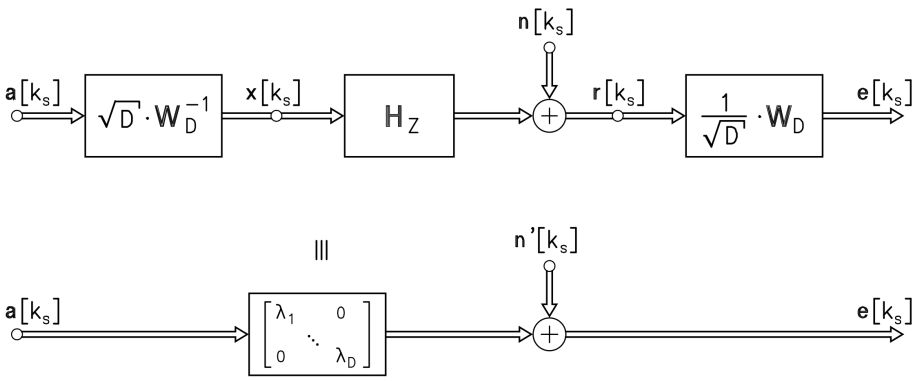


Figure 3.2: Transmission over independent subchannels

3.2 Lab Environment

The lab environment is identical to that of Project 2.

Readings for Lab 3

[HS23] Johannes B. Huber and Robert Schober, *Digital Communications*, Lecture Notes, Erlangen, October 2022.

3.3 Lab Exercises

In this project we aim to simulate an OFDM system. The first modification is the introduction of a dispersive channel. Inside the transmission chain, it is modeled as a new box between the Tranmitter and the AWGN channel.

Homework H-3.1

Even without knowing the parameters of the dispersive channel(s), give a brief statement, how the bit error curves will change.

Lab Exercise L-3.1

There is a function `dispersive_channel()` which models the dispersive channel. Add one of the three given channel types to your transmission chain of Project 2. How does the bit error ratio change?

Hint: First run the simulation without a dispersive channel, then compare this BER curve to the plots with an active dispersive channel.

Homework H-3.2

How can you observe the influence length of the dispersive channel?

In this Project we use an all-digital representation of the transmission scenario. Thus, the influence length of the channel can not be measured in seconds, but in samples only.

Lab Exercise L-3.2

Give the length of the impulse response (samples) of the given dispersive channel.

3.3.1 OFDM Transmitter

We now want to transmit the data using an OFDM system.

Homework H-3.3

Draw the block diagram of a general OFDM transmitter.

We now switch to the folder of Lab 3. In order to construct the OFDM transmitter, it is reasonable to split it into smaller blocks.

Lab Exercise L-3.3

Edit the file *ofdm_transmitter.m*. The bitstream has to be split in $D = 16$ substreams. Use the given function `map_bits_qam(bitvector, M)` for the mapping for each substream on symbols.

Lab Exercise L-3.4

Use the inverse discrete Fourier transform to generate the OFDM signal out of the symbols for each subchannel.

Homework H-3.4

How many samples are required at least for the length of the cyclic prefix in order to avoid interblock interferenece?

Lab Exercise L-3.5

Add the cyclic prefix with the optimized length fitting to the given implementation.

Next, we analyze the transmit signal.

Lab Exercise L-3.6

In order to analyze the OFDM transmit signal, we generate a signal resulting from 1,200,000 Bits. Analyze the OFDM signal in time domain and frequency domain (use the given function `analyze_signal(block)`).

Lab Exercise L-3.7

Study the distorted signal after the dispersive channel and the AWGN (use the given function `analyze_signal(block)`).

3.3.2 OFDM Receiver

Homework H-3.5

Draw the block diagram of a generic OFDM receiver.

Lab Exercise L-3.8

Edit the given file *ofdm_receiver.m* with the following head:

```
function receivedBits = ofdm_receiver(receivedSignal, M)
```

Remove the cyclic prefix from the signal.

```
 OFDM_symbols = reshape(recSignal,[], D+(cp_length));
OFDM_symbols = OFDM_symbols(:,(cp_length+1):end);
```

Lab Exercise L-3.9

Use the Fourier transform to generate the OFDM signal out of the symbols for each subchannel.

In order to equalize the effect of the scaling factors in the subchannels, we apply a so-called frequency domain equalization (FEQ).

Homework H-3.6

Assume we archieve a perfect meassurement of the dispersive channel and insert those coefficients into the FEQ-fuction. One of the effects of the equalization is seen by the behaviour of the bit error. Given a perfect meassurement, can it happen that there are no errors at the end of our transmission chain?

Lab Exercise L-3.10

The FEQ is provided in the function `feq(PAM_Symbols)`.

Note: Remember to use equivalent channel coefficients inside the files *dispersiveChannel.m* and *feq.m*.

Lab Exercise L-3.11

The bitstream has to be split up for the 16 individual subchannels. Use the given function `demap_bits_qam(PAM_Symbols, M)` for the mapping of individual stream.

After the transmission the BER can be calculated.

Lab Exercise L-3.12

Run the file *Simulation_ OFDM.m* and modify the types of `dispersiveChannel`. Study the resulting curves of BER/SNR and prove your implementation works correctly.

Project 4

Signal Space Representation

4.1 Introduction

In project 4, we implement a general transmission scheme which is based on a set of signal elements. We exploit a set of matched filters (MF) at the receiver. In this project, we show that the complexity of the receiver can be reduced by representing the signal elements as linear orthonormal combinations of basis functions.

4.2 Lab Environment

The project environment is identical to project 3. All the function templates are stored in folder ('*./matlab_for_lab4*').
Note that some functions are reused from the previous projects.

4.3 Signal Space Representation

A transmitter consists of several signal processing blocks such as an encoder, a mapper, and a modulator. Here, we consider channel code-free transmission. We assume a memoryless modulation, which consists of a set of non-orthogonal signal elements (Fig. 4.1).

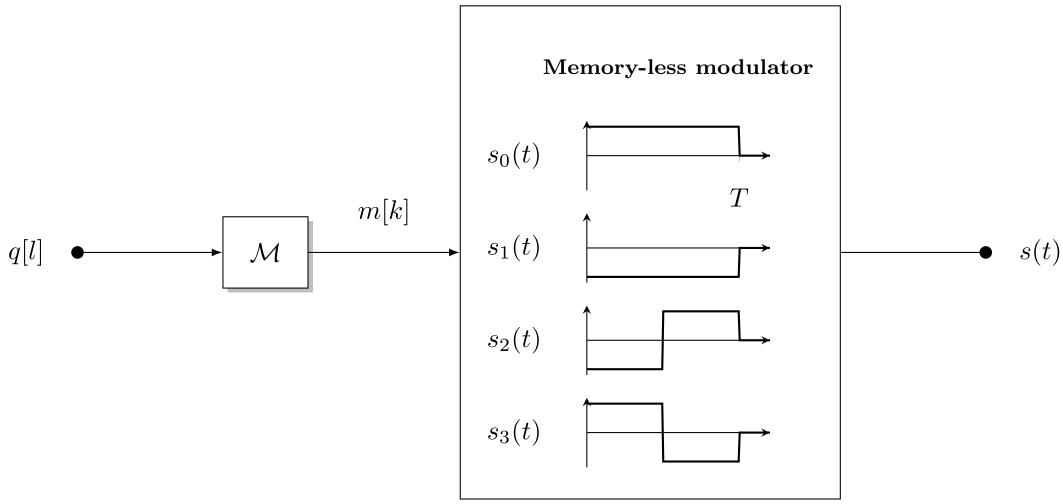


Figure 4.1: Block diagram of the considered transmitter. The binary data sequence $q[l]$ is mapped to the signal number $m[k] \in \{0, 1, 2, 3\}$, which selects the signal element.

In contrast to multiplying a basis function $g(t)$ with ± 1 as it is done for binary bipolar PAM transmission, we now consider a signal space $\mathcal{S}$ with a general set of signal elements. For each symbol interval of duration T one signal element according to m is selected. The resulting transmit signal is obtained as

$$s(t) = \sum_{k=-\infty}^{+\infty} s_{m[k]}(t - kT), \tag{4.1}$$

where $m \in \{0, \cdots, (M-1)\}$ and M denote the signal element number in symbol interval k with $s_{m[k]}(t)$ as the corresponding signal element and the number of signal elements, respectively.

4.3.1 Orthogonality

In general, the signal elements are not limited in time. Here, we focus on time-limited signals which, fulfill the temporal orthogonality condition [HS21]

$$\frac{1}{E_{i\ell}} \int_{-\infty}^{+\infty} s_i(t + kT) \cdot s_\ell^*(t)\, \mathrm{d}t = \underline{\underline{\delta_{0k}}}\,, \quad \forall i, \ell \in \{0,\ 1, \ldots,\ (M-1)\}, \tag{4.2}$$

where $E_{i\ell}$ and δ_{ij} denote the energy of the cross correlation of the signal elements $s_i(t)$ and $s_\ell(t)$ and the Kronecker-symbol, respectively, with

$$\delta_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j. \end{cases} \tag{4.3}$$

However, the signal elements might still not be mutually orthogonal. If the latter does also hold, the so-called *double orthogonality condition* is fulfilled.

$$\frac{1}{E_{g_i}} \int_{-\infty}^{+\infty} g_l(t + kT) \cdot g_\ell^*(t)\, \mathrm{d}t = \underline{\underline{\delta_{l\ell} \cdot \delta_{0k}}}\,, \quad \forall i, \ell \in \{0,\ 1, \ldots,\ (M-1)\}. \tag{4.4}$$

Throughout this project the signal elements in general are described by $s_i(t)$ with $i \in \{0, \ldots, (M-1)\}$. Signal elements which fulfill the double orthogonality condition are denoted as basis functions $g_i(t)$. They constitute the orthonormal basis of the signal space.

The continuous-time signals $s_i(t)$ and $g_i(t)$ have to be transformed to discrete-time vectors with $s[\kappa] = s(kT)$ for the implementation in MATLAB . Each signal element can be represented by a vector with T/T_s elements (T/T_s must be an integer) and contains one symbol. T_s denotes the duration between two consecutive samples.

The double orthogonality condition in discrete-time is given as follows

$$\frac{1}{E_g} \sum_{k=-\infty}^{+\infty} g_l[k] \cdot g_\ell^*[k] = \underline{\underline{\delta_{l\ell} \cdot \delta_{0k}}}\,, \quad \forall i, \ell \in \{0,\ 1, \ldots,\ (M-1)\}. \tag{4.5}$$

All signal elements and basis functions are collected in a vector, respectively, as follows

$$\boldsymbol{g}(t) \stackrel{\text{def}}{=} [g_0(t), g_1(t), \dots, g_{M-1}(t)] \qquad \text{(continuous-time)} \tag{4.6}$$

$$\boldsymbol{s}(t) \stackrel{\text{def}}{=} [s_0(t), s_1(t), \dots, s_{M-1}(t)] \qquad \text{(continuous-time)} \tag{4.7}$$

$$\boldsymbol{g}[k] \stackrel{\text{def}}{=} [g_0[k], g_1[k], \dots, g_{M-1}[k]]\,. \qquad \text{(dicrete-time)} \tag{4.8}$$

$$\boldsymbol{s}[k] \stackrel{\text{def}}{=} [s_0[k], s_1[k], \dots, s_{M-1}[k]]\,. \qquad \text{(dicrete-time)} \tag{4.9}$$

In discrete-time the resulting matrix $\boldsymbol{g}[k]$ or $\boldsymbol{s}[k]$ has the dimensions

$$\mathcal{M} \times (T/T_{\mathrm{s}}). \tag{4.10}$$

The double orthogonality condition can be tested with

$$\frac{1}{E_g} \sum_{\kappa=-\infty}^{+\infty} \boldsymbol{g}^\top[\kappa] \cdot \boldsymbol{g}^*[\kappa] = \underline{\underline{\boldsymbol{I}_{M\times M}\delta_{0k}}} \tag{4.11}$$

4.3.2 Orthogonalization

Although in most digital transmission schemes the signal elements are not mutually orthogonal, that signal space defined by the signal elements can be equivalently represented by orthonormal basis functions.

Therefore we derive the basis functions $g_i(t)$ and $g_i[k]$, which are in most cases not unique for a given set of signal elements. Furthermore, we derrive the linear factors $s_{i,l}$. Hence, each signal element can be represented as a *linear combinations* of D weighted basis functions [HS21]:

$$s_i[k] = \sum_{l=0}^{D-1} s_{l,i} \cdot g_l[k]\,. \tag{4.12}$$

Here, $D \leq M$ denotes the dimension of the signal space. Introducing a matrix-vector notation, we have

$$\boldsymbol{S} = \begin{bmatrix} s_{0,0} & \cdots & s_{0,(M-1)} \\ \vdots & s_{\ell,i} & \vdots \\ s_{(D-1),0} & \cdots & s_{1,(M-1)} \end{bmatrix} \tag{4.13}$$

which results in

$$\boldsymbol{s}[k] = \boldsymbol{g}[k]\boldsymbol{S}. \tag{4.14}$$

As stated before, this representation provides a mapping from linear combinations of basis functions to signal elements. This process can be reversed. The orthonormal basis functions can be represented as linear combinations of *orthogonalization factors* $g_{i,l}$ and the signal elements themselves

$$g_l[k] = \sum_{i=0}^{M-1} g_{i,l}\, s_i[k]. \tag{4.15}$$

To obtain a set of orthonormal basis functions we utilize on the so-called Gram-Schmidt Procedure (GSP) which is described by a recursive algorithm [HS21] and calculates:

- *Linear Factors* (linear factor matrix $\boldsymbol{S}$)
- *Orthonormal Basis Functions* $g_i(t)$ and $g_i[k]$

The set of M row vectors $s_0[k], s_1[k], \ldots, s_{M-1}[k]$ represents the signal elements over time k. The goal of the Gram-Schmidt Procedure is to find a set of orthonormal basis vectors $g_0[k], g_1[k], \ldots, g_{D-1}[k]$ with unknown dimensions $D \leq M$ and linear factors $s_{\ell,i}$ such that

$$s_i[k] = \sum_{\ell=0}^{D-1} s_{\ell,i} \cdot g_\ell[k]. \tag{4.16}$$

The recursive algorithm proceeds as follows:

1. Calculate the first basis function using the first signal element

   $$g_0[k] = \frac{1}{\sqrt{\sum_\kappa s_0[\kappa] \cdot s_0^*[\kappa]}} s_0[k]$$

   Number of basis functions determined so far: $l = 1$

   Linear factors for $\boldsymbol{s}_0$:

   $$s_{0,0} = \sqrt{\sum_\kappa s_0[\kappa] \cdot s_0^*[\kappa]}, \quad \text{(Here, } s_{\ell,0} = 0 \text{ for } \ell = 1, 2, \ldots, D-1\text{)} \tag{4.17}$$

   $$\boldsymbol{S} = \begin{bmatrix} \sqrt{\sum_\kappa s_0[\kappa] \cdot s_0^*[\kappa]} & \cdots \\ 0 & \\ \vdots & \ddots \\ 0 & \end{bmatrix} \tag{4.18}$$

   $$\boldsymbol{g}[k] = \begin{bmatrix} \frac{1}{s_{0,0}} s_0[\kappa], & \cdots \end{bmatrix} \tag{4.19}$$

   Dimension D still undetermined.

2. Continue with the next vector $s_i[k]$ ($i = 1, 2, M-1$):

   a) Linear factors regarding the l basis functions computed so far:

$$s_{\ell,i} = \sum_k s_i[k] \cdot g_\ell^*[k] \tag{4.20}$$

$$\boldsymbol{S} = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & \cdots \\ 0 & s_{1,1} & s_{1,2} & \ddots \\ 0 & 0 & s_{2,2} & \ddots \\ 0 & 0 & 0 & \ddots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{4.21}$$

$$\boldsymbol{g}[k] = \left[\left(\tfrac{1}{s_{0,0}} s_0[k]\right), \quad \left(\tfrac{1}{s_{0,1}} s_0[k] + \tfrac{1}{s_{1,1}} s_1[k]\right), \quad \cdots \right] \tag{4.22}$$

   b) Now, checkt if the current signal element can already be represented by the existing basis functions:

$$\Delta_i = \underbrace{\sum_k s_i[k] \cdot s_i^*[k]}_{\text{energy of the signal element}} - \underbrace{\sum_{\ell=0}^{l-1} |s_{\ell,i}|^2}_{\text{energy of the existing basis function}} \tag{4.23}$$

      i. $\Delta_i = 0$: The current signal element $s_l[k]$ can fully be represented by a linear combination of existing basis functions. No new basis function is needed.

$$s_{\ell,i} = 0 \text{ für } \ell = l, l+1, \ldots, D-1$$

      ii. $\Delta_i > 0$: There is energy left, that cannot be represented by a linear combination of basis functions. Thus a new basis function has to be introduced.

$$g_l[k] = \frac{1}{\sqrt{\Delta_i}} \left(s_i[k] - \sum_{\ell=0}^{l-1} s_{\ell,i} g_\ell[k] \right) . \tag{4.24}$$

      Determine the new linear weight for $s_i[k]$:

$$s_{l,i} = \sqrt{\Delta_i}, \quad s_{\ell,i} = 0 \text{ for } \ell = l+1, l+2, \ldots, D-1 . \tag{4.25}$$

      Increment number of basis functions: $l \to l+1$

3. As long as $i < M-1$, repeat step 2 with element $s_{i+1}[k]$ (recursion)
4. Orthogonalization finished. Dimensionality: $D = l$

The Gram-Schmidt Procedure results in a set of orthonormal basis functions which describes the same signal space as the original set of signal elements, and a set of linear weights for each basis.

The above description uses an "incremental" order to select the next signal element in the progress. Alternatively one can choose the signal element which has the maximum energy Δ_i that cannot be represented by the linear combination of basis functions. This modification denotes as "sorted-by-energy".

Readings for Lab 4

[HS21] Johannes B. Huber and Robert Schober, *Digital Communications*, Lecture Notes, Erlangen, October 2021.

[Kam08] Karl-Dirk Kammeyer, *Nachrichtenübertragung*, 4 ed., B. G. Teubner, Stuttgart, March 2008.

[Pro00] John G. Proakis, *Digital communications.*, 4th ed., McGraw-Hill, New York, NY, USA, 2000.

4.4 Homework and Lab Exercises

4.4.1 Information Transmission Using Signal Elements

We define the signal elements as depicted in Fig. 4.2. We show three different transmission schemes with signal elements $s_{1,i}(t)$, $s_{2,i}(t)$, and $s_{3,i}(t)$, $i \in \{0, \ldots, 3\}$, respectively. The sets differ due to different modulation schemes, which are employed to generate the signal elements.

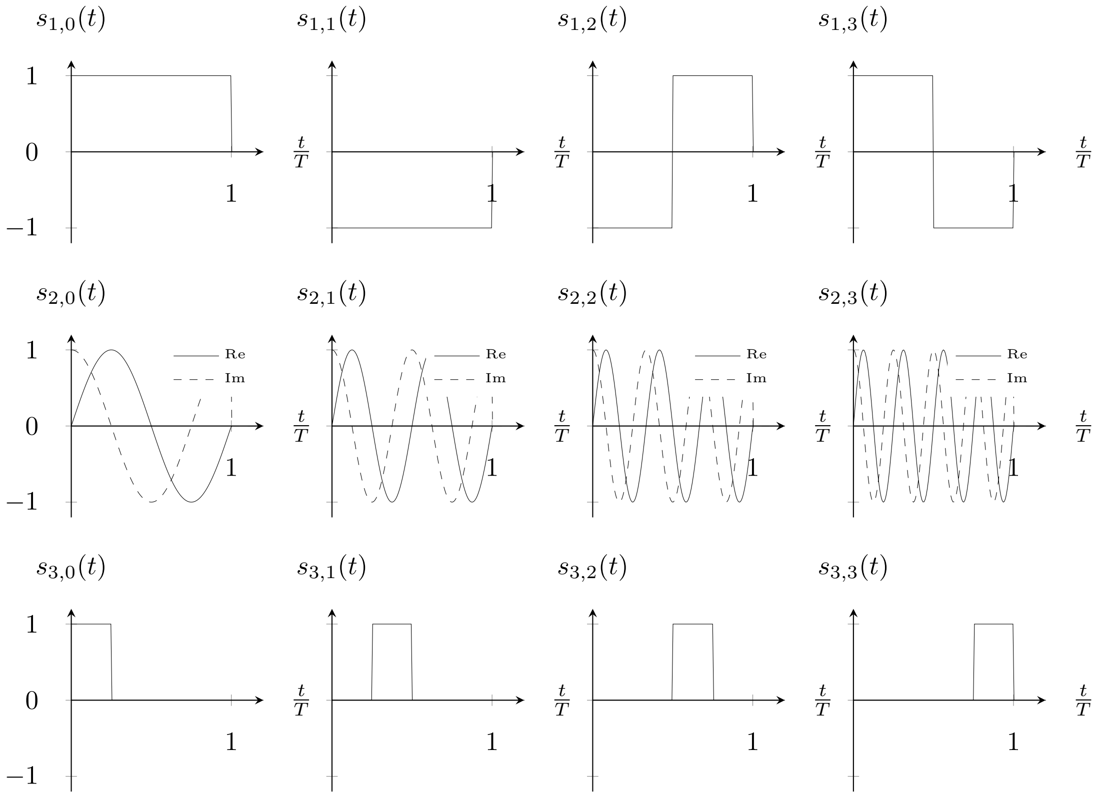


Figure 4.2: The considered transmissions schemes and their corresponding signal elements

Homework H-4.1

Explain how the sets of signal elements shown in Fig. 4.2 are assigned to the following multiplexing schemes:

- Time Division Multiplex (TDM)
- Code Division Multiplex (CDM)
- Frequency Division Multiplex (FDM)

Lab Exercise L-4.1

Edit the file `generateSignalElements.m` which contains the function with the following head:

```
function signalElements = generateSignalElements(type, ovs)
```

Generate sets of signal elements depending on the parameter `type`. We refer to Figure 4.2, which provides an overview of the three considered possibilities.

Lab Exercise L-4.2

Edit the file *transmitter_SE.m* which contains a function with the following head:

```
transmitSignal = transmitter_SE(bitstream, type, ovs)
```

The transmitter calls the function `generateSignalElements` to generate the possible signal elements for the specified type. The transmit signal should be created by concatenating the signal elements which are coded by the bitstream.

Hint: Use the MATLAB-function `bi2de` to get the number of the signal elements.

Homework H-4.2

Determine the sequence of signal numbers and the respective bitstream for the following transmit signal s(t) which uses the signal numbers $s_{1,0}(t), \ldots s_{1,3}(t)$! Refer to Table 4.1 for the corresponding bits.

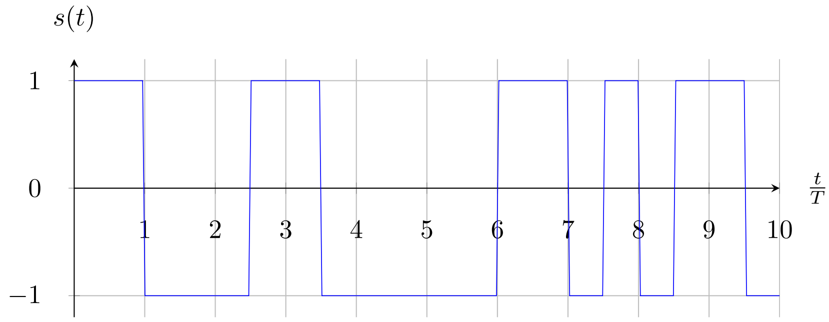


Figure 4.3: Transmit signal s(t)

Bits	0 0	0 1	1 0	1 1
Signal Element	$s_{1,0}(t)$	$s_{1,1}(t)$	$s_{1,2}(t)$	$s_{1,3}(t)$

Table 4.1: Mapping of bits to signal elements

Lab Exercise L-4.3

Implement the Homework 4.2 as a Matlab script.

Homework H-4.3

Using your knowledge in signal space representation:

- Which signal elements exist for a 4-ary PAM with a rectangular pulse shape?
- How many possibilities exist for the mapping from binary data streams to signal numbers?

Furthermore

- Sketch all signal elements!
- Determine the number of orthonormal basis functions which are needed to represent all signal elements?

4.4.1.1 Correlation Receiver

At the receiver we employ coherent maximum likelihood (ML) detection, which is realized by a correlation receiver as depicted in Fig. 4.4. Here, the received signal $r(t)$ denotes the input signal. The correlation receiver consists of all matched filters needed according to the signal elements in the signal space $\mathcal{S}$. The correlation receiver filters the received signal $r(t)$ by each signal element $s_i(t)$. After sampling, we archieve a correlation vector $\boldsymbol{d}[k]$ which provides *sufficient statistics* for the receiver input signal $r(t)$ with respect to data estimation. The final decision for a signal number is then simply taken by picking the *maximum in the correlation vector*.

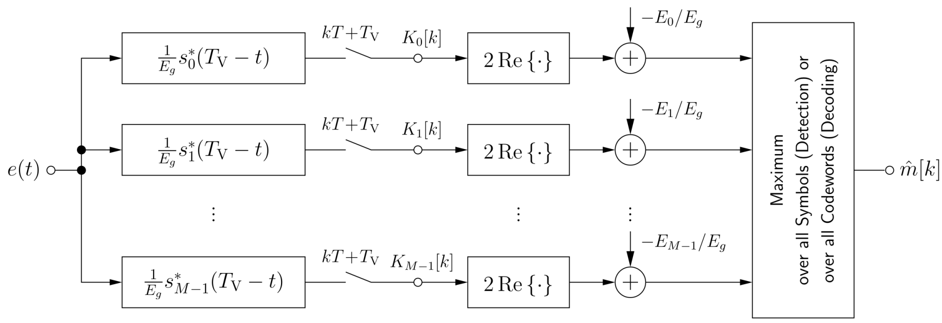


Figure 4.4: Correlation Receiver [HS21]

Lab Exercise L-4.4

Edit the file *receiver_SE.m* using the signal elements defined in exercise **L-4.1**! We recommend to first implement the filter bank in the correlation receiver and the correlation function using the MATLAB table with the signal elements. Take a look at the results of the correlations before implementing the sampling. After locating the maximum value in the correlation vector, compare the sequence of signal numbers to those at the transmitter!

Lab Exercise L-4.5

Modify the parameters inside the script *simulate_SE.m*. The simulation uses your recently implemented functions `generateSignalElements`, `transmitter_SE` and `receiver_SE`. Together with the provided functions `channel` and `calculateBER` a transmission chain is created.

Run the simulation script seperately for every single of the three possible signal constellation types and plot the resulting curves of the error probabilities over the SNR into the same figure.

4.4.2 Gram-Schmidt Procedure

We now consider the Gram-Schmidt Procedure to orthogonalize the signal elements.

Homework H-4.4

Use the Gram-Schmidt Procedure according to the lecture notes to determine a set of orthonormal basis functions for the signal space defined by the signal elements $s_0(t)$, $s_1(t)$ and $s_2(t)$ shown below and skecth them. Also express $s_0(t)$, $s_1(t)$ and $s_2(t)$ in terms of these basis functions. Summarize your results and provide the matrix of the orthogonalization factors and the matrix of the linear weights.

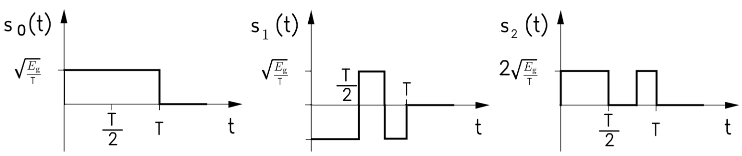


Figure 4.5: Signal elements for homework **H-4.4**

Lab Exercise L-4.6

Implement the Gram-Schmidt Procedure in MATLAB!

Lab Exercise L-4.7

Compare the result of your homework with the result of your implementation!

Lab Exercise L-4.8

Orthogonalize the signal elements $s_{1,0}(t)$, $s_{1,1}(t)$, $s_{1,2}(t)$ and $s_{1,3}(t)$ given in exercise **H-4.1**! Sketch the resulting orthonormal signal elements $g_i(t)$! How many basis functions are necessary?

Lab Exercise L-4.9

Discuss how to describe the original signal elements $s_i(t)$ using the orthonormal basis vectors $g_i(t)$! Rebuild the original signal elements in MATLAB.

Homework H-4.5

Describe in your own words how the receiver in task **L-4.4** has to be modified so that a matched filter bank with less filters can be used!

Homework H-4.6

Assume that you need to further reduce the complexity at the receiver and therefore you decide to neglect one basis function. Which basis function would you neglect. Does the option “incremental” and “sorted-by-energy” influence the decisions?

4.4.3 Frequency Shift Keying

In a M-ary frequency shift keying modulation scheme (FSK) [Kam08, Pro00] the signal elements are defined as

$$s_i(t) = \begin{cases} \sqrt{E_g/T}\ \mathrm{e}^{\mathrm{j}(2\pi i\, h\, t/T + \varphi_0)} & \text{for } t \in [0,\ T) \\ 0 & \text{for } t \notin [0,\ T) \end{cases} \tag{4.26}$$

For the discrete time version it holds $t = nT_\mathrm{s}$:

$$s_i[n] = \begin{cases} \sqrt{E_g/T}\ \mathrm{e}^{\mathrm{j}(2\pi i\, h\, (nT_\mathrm{s})/T + \varphi_0)} & \text{for } n \in \{0, 1, \ldots, (T/T_\mathrm{s} - 1)\} \\ 0 & \text{else} \end{cases} \tag{4.27}$$

with a modulation index $h \in \mathbb{R}$ which describes the maximum phase shift and the symbol numbers $i = 1,\ 2, \ldots,\ M$. The transmit signal $s(t)$ can then again be described as a concatenation of signal elements.

Homework H-4.7

Determine the modulation index h and the alphabet size M of an FSK scheme which uses the signal elements $s_{2,0}(t)$,...,$s_{2,3}(t)$ in Fig. 4.2 (real value of $s_{2,i}(t)$ plotted)!

Lab Exercise L-4.10

Generate a FSK signal elements for $M = 4$ and $h = \frac{1}{4}$! Plot them and describe the result!

Lab Exercise L-4.11

Determine the orthonormal basis functions for the signal space in exercise **H-4.10**! Describe the resulting basis vectors!

Homework H-4.8

How are the FSK parameters correlated that result in orthogonal signal elements?

Project 5

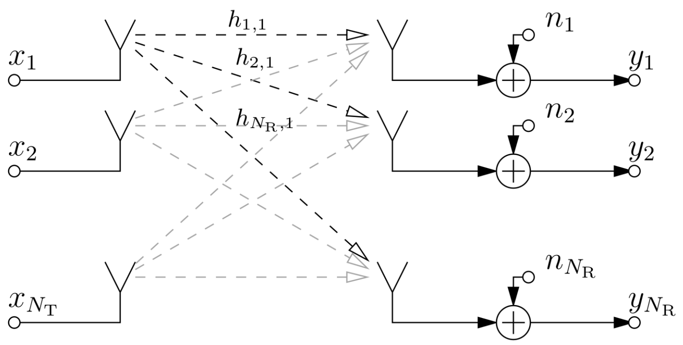
Signal Processing in MIMO Systems

5.1 Introduction, Background, and Motivation

Modern digital communication systems employ multiple antennas at the transmitter and the receiver side. Two basic principles can be distinguished which constitute the benefits of multiple-input/multiple-output (MIMO) systems compared to single antenna systems: on the one hand the multiplexing gain, i. e., the increased data rate due to transmission of independent data streams over multiple transmit antennas. On the other hand, the possibility to observe several independent copies of the transmit signal, e. g., through the use of multiple receive antennas, and thus increased robustness to shadowing and noise due to diversity.

In this project, an introduction to signal processing for MIMO communications is given, motivating the potentials and the benefits of the techniques enabled through the use of multiple antennas at transmitter and/or receiver. To this end, we consider the discrete-time equivalent model of digital pulse-amplitude modulation transmitted over a flat-fading MIMO channel with AWGN as depicted below.

Aspects such as pulse-shaping, up-/down-conversion to and from the carrier frequency, as well as matched filtering are hidden in this equivalent model, and implicitly assumed to op-



erate perfectly synchronized. With these assumptions, adopting a compact vector/matrix-notation we define the transmit signal as

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_{N_\mathrm{T}} \end{bmatrix} \tag{5.1}$$

and the received signal as

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_{N_\mathrm{R}} \end{bmatrix} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{5.2}$$

where $\boldsymbol{n}$ is a $(N_\mathrm{R} \times 1)$-vector collecting the additive white Gaussian noise samples on each receive antenna and $\boldsymbol{H}$ denotes the $(N_\mathrm{R} \times N_\mathrm{T})$-dimensional MIMO channel matrix given by

$$\boldsymbol{H} = \begin{bmatrix} h_{1,1} & \dots & h_{1,N_\mathrm{T}} \\ \vdots & \ddots & \vdots \\ h_{N_\mathrm{R},1} & \dots & h_{N_\mathrm{R},N_\mathrm{T}} \end{bmatrix} \tag{5.3}$$

The transmit symbols x_i, $i = 1, \dots, N_\mathrm{T}$, are taken from an arbitrary complex-valued signal constellation (here, we will only consider square QAM constellations).

We begin with studying the special case of single-input/multiple-output (SIMO) transmission, i. e., the transmitter communicates a single data stream using a single transmit antenna ($N_\mathrm{T} = 1$) to a receiver which is equiped with multiple receive antennas (N_R). The results are compared to a conventional single-input/single-output (SISO) system. Next, we consider the dual case of multiple-input/single-output (MISO) transmission ($N_\mathrm{T} \geq 1$, $N_\mathrm{R} = 1$) and briefly review a technique called space-time codes. We will then extend our considerations to include multiple antennas at transmitter and receiver side and implement and discuss two different (receiver-side) equalization strategies for this MIMO system. To this end, we first implement a generic system model of MIMO transmission using QAM.

Noteworthy, multi-antenna systems are only one, but probably the most intuitive example for MIMO systems; other communication systems, as, e. g., code-division multiple-access (CDMA) systems, can be modeled in a similar way. The studied principles of this project are thus applicable in a broad class of digital communication systems.

5.2 Lab Environment

The MIMO transmission system considered in this lab is implemented in MATLAB. Different from previous labs, all simulations operate in the discrete-time equivalent system of digital transmission. Pulse shaping and modulation is not required; functions of previous labs are not reused.

5.3 Homework and Lab Exercises

5.3.1 Signal Constellations

Homework H-5.1

Give an equation for the variance of the transmit symbols for the case of a M-ary square-QAM constellation. Assume that the signal points are taken from the grid

$$\mathcal{A} \subseteq \{\pm 1 \pm \mathrm{j},\ \pm 3 \pm \mathrm{j},\ \pm 1 \pm 3\mathrm{j},\ \pm 3 \pm 3\mathrm{j}, \dots\}\ . \tag{5.4}$$

Lab Exercise L-5.1

Implement a function

```
x = GetQAM(r, c, M)
```

which generates an $r \times c$ matrix $\boldsymbol{x}$ containing M-ary QAM symbols randomly drawn from $\mathcal{A}$. Generate a matrix $\boldsymbol{x}$ of reasonable length and determine the variance of the entries. Compare this result with the analytical one from the lecture notes.

For the following tasks it should at least be working with 4-QAM and 16-QAM.

Hint: The MATLAB-functions `randi` and `qammod` can be helpful.

Lab Exercise L-5.2

Implement a function

```
a_hat = QuantQAM(z, M)
```

which quantizes the elements of the matrix $\boldsymbol{z}$ to M-ary QAM symbols.

For the following tasks it should at least be working with 4-QAM and 16-QAM.

Hint: Be aware of a possible amplification on all signal elements while passing the channel and scale the input matrix z.

5.3.2 Channel Models

In this project we will only consider the following channel model:

- the elements $h_{m,n}$ of the MIMO channel matrix $\boldsymbol{H}$ are independent zero-mean unit-variance complex-Gaussian distributed
- the elements of the additive noise vector are independent zero-mean complex-Gaussian distributed with equal variance σ_n^2.

Homework H-5.2

Interprete the assumptions of this channel model and briefly explain the reasoning behind.

Homework H-5.3

Give simple methods, which generate a random channel realization $\boldsymbol{H}$ and a realization of the noise vector $\boldsymbol{n}$ according to this model in MATLAB.

In this lab, we restrict ourselves to the performance measure symbol error rate (SER).

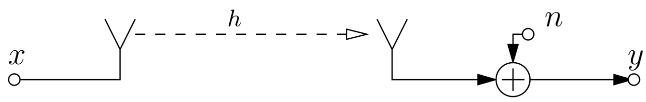
Homework H-5.4

Give a simple method to calculate the SER in MATLAB given the QAM transmit symbols and its estimates.

We now have implemented three major building blocks of a generic MIMO transmission: the transmitter generating a sequence of QAM symbols transmitted over multiple antennas, the MIMO AWGN channel, and the decision device quantizing the equalized symbols into QAM symbols.

5.3.3 SISO

As a reference, we first implement a conventional SISO system ($N_{\mathrm{R}} = N_{\mathrm{T}} = 1$), as depicted below.



As in the generic MIMO scenario we assume the fading coefficient h to be complex-Gaussian distributed with zero mean and unit variance.

Homework H-5.5

State the name of the distribution of the magnitude of the fading coefficient, i.e., the channel gain, and give a formula for its probability density function (pdf).

Lab Exercise L-5.3

Write a simulation script which generates and transmits a sequence of QAM symbols according to this setup.
The results shall be averaged over a large number of channel realizations. Include a `for`-loop in order to simulate the SER for different SNRs and plot the resulting SER-vs.-SNR curve. Take care that the SNR is correctly set. Measure the SER for 4-QAM and an SNR range of $-10, 5, \ldots, 50$.

Note: Several (L) symbols can be transmitted over a single channel realization simultaneously by constructing a $(1 \times L)$-dimensional transmit vector.

Lab Exercise L-5.4

Estimate the pdf of the channel gain (using the built-in MATLAB-function `histogram`) and compare it to the analytical results from **H-5.5**.

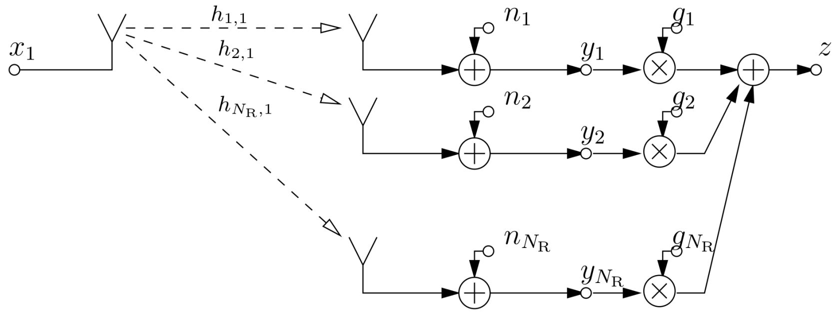
5.3.4 SIMO

We now consider the case of transmission of a single data stream (transmit symbols x with variance σ_x^2) using a single transmit antenna ($N_\mathrm{T} = 1$) to a receiver equiped with multiple (N_R) receive antennas.

Grouping the receive symbols into a vector, we have

$$\boldsymbol{y} = \boldsymbol{h}x + \boldsymbol{n} \tag{5.5}$$

Linearly combining the signals observed at the antennas as depicted below (combining coefficients g_n) offers to significantly increase robustness to noise (diversity gain); there is no multiplexing gain compared to the SISO system.



Assuming channel state information at the receiver side, the optimum (w.r.t end-to-end SNR) combining strategy for this SIMO setup is maximum ratio combining (MRC) which sets $g_n = h_n^*$.

Homework H-5.6

Write the signal processing of MRC compactly in vector/matrix-notation and give an interpretation of the underlying principle.

Lab Exercise L-5.5

Extend your simulation to incorporate multiple antennas at the RX side and implement MRC. Measure the SER for 4-QAM, $N_\mathrm{R} = 2$, and an SNR range of $-10, 5, \ldots, 50$. For comparison, the SER of both schemes SISO and SIMO shall be recorded.

Note that the combined receive signal has to be scaled to match the expected input range of the decision device.

Lab Exercise L-5.6

Which effect is observed for an increasing number of receive antennas.
Run simulations for $N_{\mathrm{R}} = 4$, and 8 and also for different QAM constellations to verify your observations.

Using MRC, the equivalent effective channel $z = \|\boldsymbol{h}\|x$ can be introduced, i.e., the effective channel gain calculates to $\|\boldsymbol{h}\|^2$.

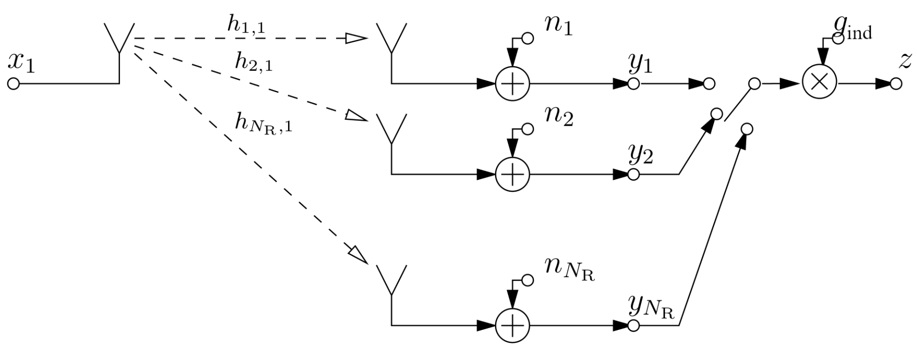
Homework H-5.7

State the name of the distribution of the effective the channel gain and give its mean and variance.

Lab Exercise L-5.7

Estimate the pdf of the channel gain (using the built-in MATLAB-function `histogram`) and compare it to the analytical results and the pdf of the channel gain of SISO transmission.

A simpler scheme to exploit the benefits of multiple antennas at the receiver is so-called antenna selection (AS), as depicted below.



In AS, for each time step, the best antenna is selected; its receive symbol is used as a decision variable.

Homework H-5.8

Explain the advantages of AS over MRC with respect to hardware implementation complexity.
Which antenna should be selected in each time step?

Lab Exercise L-5.8

Extend your simulation script to incorporate AS and compare the resulting performance with MRC and the SISO system for different values of N_R.
Which effect do you observe?

5.3.5 MIMO

In the final part of this project, we investigate systems employing multiple antennas at both transmitter and receiver side. Compared to the previous systems, such schemes offer the potential to have both, a multiplexing gain (increased data rate through the transmission of independent data symbols on the antennas for each time step) and a diversity gain (increased robustness to noise).

In vector/matrix-notation, the MIMO receive signal can be written as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{5.6}$$

Similar to equalization of inter-symbol interference in digital communication over dispersive channels, the straight-forward approach for equalization of the interference induced by the use of the same transmit medium (time/frequency) is so-called zero-forcing linear equalization (ZF-LE). In case of square systems ($N_\mathrm{T} = N_\mathrm{R}$) the receive signal is processed with the inverse of the MIMO channel matrix, yielding

$$\boldsymbol{z} = \boldsymbol{H}^{-1}\boldsymbol{y} \tag{5.7}$$

In case of non-square systems, the left-pseudo inverse (Moore-Penrose inverse) has to be employed, yielding

$$\boldsymbol{z} = \left(\boldsymbol{H}^\mathsf{H}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^\mathsf{H}\boldsymbol{y} \tag{5.8}$$

Lab Exercise L-5.9

Write a new simulation script which implements MIMO transmission with ZF-LE and run simulations for 4-QAM using $N_\mathrm{T} = N_\mathrm{R} = 4$.

Which diversity order is obtained using this simple equalization strategy?

Note: Several symbols can be transmitted over a single channel realization simultaneously, by letting $\boldsymbol{x}$ represent a matrix of dimension $N_\mathrm{T} \times$ number of channel uses.

Similar to equalization of inter-symbol interference in digital communication over dispersive channels, linear equalization is also possible in a minimum mean-squared error (MMSE) sense. In this case, we have

$$\boldsymbol{z} = \left(\boldsymbol{H}^\mathsf{H}\boldsymbol{H} + \frac{\sigma_n^2}{\sigma_x^2}\boldsymbol{I}\right)^{-1} \boldsymbol{H}^\mathsf{H}\boldsymbol{y} \tag{5.9}$$

Lab Exercise L-5.10

Extend your simulation script incorporating MMSE-LE and compare the results with ZF-LE for different parameter settings.

The optimal detection scheme for MIMO transmission is maximum-likelihood detection. As the receive signal $\boldsymbol{y}$ is corrupted by AWGN, this is equivalent to finding the noise-free signal point with minimum Euclidean distance to the receive signal point, i. e.,

$$\boldsymbol{x}^\mathsf{ML} = \operatorname{argmin}_{\tilde{\boldsymbol{x}}} \|\boldsymbol{y} - \boldsymbol{H}\tilde{\boldsymbol{x}}\|^2 \tag{5.10}$$

There are various algorithms to solve the ML detection problem with moderate computational complexity. Here, we only consider the simple strategy of a full search over all candidate signal points.

Homework H-5.9

Calculate and discuss the number of candidate signal points for M-QAM transmitted over N_T transmit antennas.
