Technische Universität München
Lehrstuhl für Nachrichtentechnik
Prof. Dr. sc. techn. Gerhard Kramer

Bachelor's Thesis

Communication over unknown channel

Vorgelegt von:
Kevin Li

München, January 2018

Betreut von:
Marcin Pikus

Bachelor's Thesis am
Lehrstuhl für Nachrichtentechnik (LNT)
der Technischen Universität München (TUM)
Titel : Communication over unknown channel
Autor : Kevin Li

Kevin Li
Am Römerbrunnen 6
85586 Poing
kev.li1010@gmail.com

Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

München, 22.01.2018

Ort, Datum (Kevin Li)

Contents

List of Figures

List of Tables

1 Communication Chain

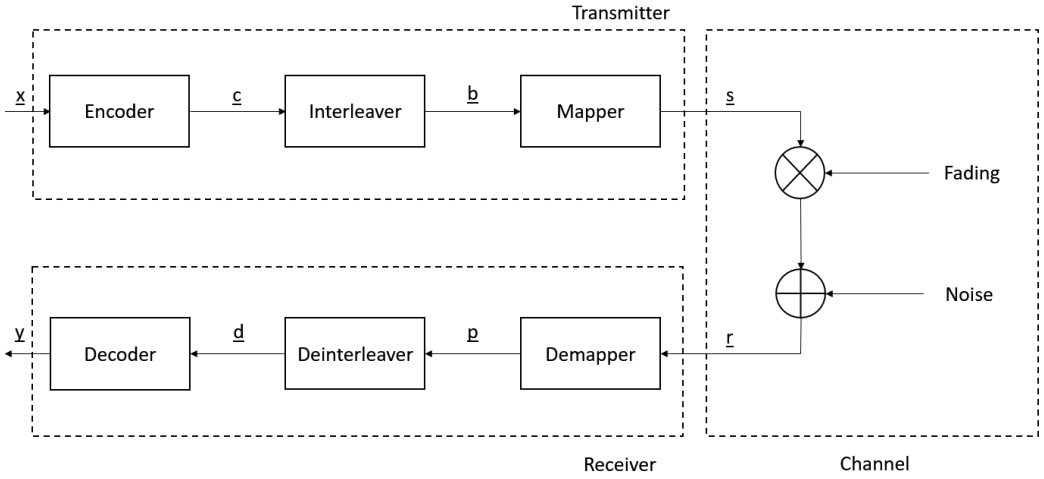

Figure 1.1: Communication chain for simulation

First, the communication chain for simulations is introduced (Figure 1.1). The link is built up of three main blocks: The transmitter, channel and receiver. The transmitter in turn contains the encoder, interleaver and mapper. We start with feeding a random generated bit stream, representing a message, $\underline{u}$ into the encoder. The resulting encoded code word $\underline{c}$ is next processed in the interleaver producing the shuffled code word $\underline{c}$'. The mapper can now modulate $\underline{c}$' into the desired modulation scheme with the symbol array $\underline{x}$. In the channel various kind of noises and fading can be added to the modulated signal,e.g., additive white gaussian noise (AWGN). In the end the receiver consisting of the counterparts build in the receiver will first demap the signal $\underline{y}$ to the estimated code word $\underline{\hat{c}}$. After de-interleaving and decoding the transmitted symbol to determine an estimate $\underline{\hat{u}}$. In the simulation we will compare the decoded message $\underline{\hat{u}}$ with the initially created input message $\underline{u}$ to determine the error rate in the system.

1.1 Encoder/Decoder

There are many ways to make the transmission more stable and less error prone. A major role in this protection plays the encoder and its counterpart the decoder. Encoder/decoder come in my different forms and shapes,e.g., as pre-built circuits in systems but more commonly today as software coder. They reach from simple linear block codes to more complex convolutional coding to the latest turbo codes. It is also important to note, that code working well in AWGN channel will often not the same performance in a fading channel.
A further look will be taken into low density parity check (LDPC) codes, in particular the WiMax code. !!citation!!. While LDPC was mainly ignored in the past, since the 1999's the introduction of turbo codes and a sharp increase in computing power helped the recognition of these forms of channel coding.
LDPC-codes are linear block codes with a particular structure for their parity check matrix **H**. In the case of LDPC-codes **H** has only a small amount of nonzero entries, which means that there is a low density in the parity check matrix. Another important difference in LDPC to turbo codes is the complexity of encoding and decoding. While turbo codes have low complexity in encoding they have high complexity in decoding. The total opposite can be said about LDPC with high complexity in encoding and low complexity in decoding.
WiMax IEEE 802.16e is a standard code model used in small and medium distances in urban areas, which fits the simulations quite well. It should be noted, that for WiMax there are predefined code length and code rates and encoding classes. Code lengths can range from 576 bits up to 2034 bits. Code rates !!footnote!! are divided into four rates: 1/2, 2/3, 3/4, 5/6. The simulation will also only be run with coding class A.

1.2 Bit interleaver/De-interleaver

While the above mentioned LDPC code (Chapter (1.1)) works really well for an AWGN channel this is not always the case in a fading channel. This is where the next important channel block comes into play. To guarantee a stable performance the method of interleaving will be introduced. Interleaving will handle a major problem in fading channels, the appearence of burst errors mainly caused by deep fading over a set time. LDPC coding suffers from loss of performance trying to correct these burst errors, deteriorating with the increase of the burst error length. With the interleaver the code word will be shuffled into a new random Gaussian distributed code word, which will be passed through the channel. At the receiver a restoration of the shuffled code word back into its initial state will take place. As clearly seen in Figure 1.2 the interleaver will not remove

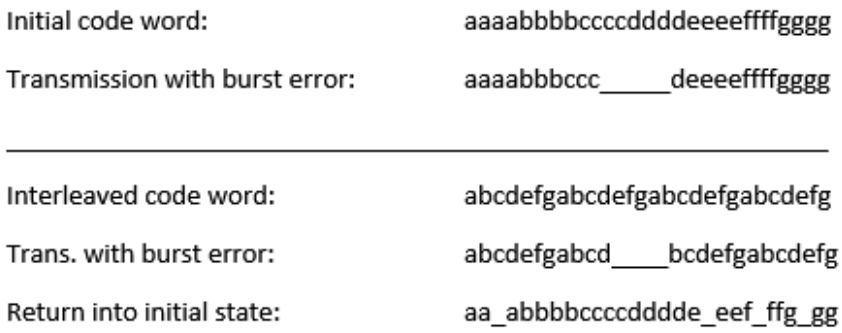

Figure 1.2: Example for interleaving

any errors but will prevent or at least mitigate the presence of burst errors. The LDPC decoder can correct single errors again. There are two main methods of interleaving today: symbol-interleaved coded modulation (SICM) will interleave the symbols after the modulator while bit-interleaved coded modulation (BICM) will interleave the single bits before the modulator block. BICM will be used in this thesis for having a more dominant position in practical communication systems. !!cite!!

1.3 Mapper/Demapper

In this block the mapper, also called the modulator, it is possible to assign the code word a specific symbol. Group of bits are taken from the bit stream to combine them to specific constellation points. The symbols are located in a real/imaginary plane, also called Inphase/Quadrature planes (I/Q-planes). With the distance from the nullpoint of the axis giving us the magnitude of the signal and the angle to the real axis the phase shift. !!I/Q plane with showing amplitude,phase shift in signal There are many forms of modulation schemes, with the most common ones being M-phase shift keying (PSK), M-frequency shift keying (FKS), M-amplitude modulation (AM) and M-quadrature amplitude modulation (QAM). For the simulation, a further look will be taken at quadrature phase shift keying (QPSK), 16-QAM and 64-QAM, which are all depicted below (Figure **??**).
With QPSK the symbols all share the same amplitude and only differ in their respective phase angle. To each symbol we can assign $log2(M)$ bits, with M being the number of symbols in the scheme. Therefore, for QPSK the number of bits per symbol amount to 2.
For QAM, signals which differ in their phase shift and also their amplitude, will be send. For 16-QAM a maximum of 4 bits per symbols and for 64-QAM 6 bits per symbol can be achieved.

1.4 Channel

The channel can be modeled in many different ways. Various sources of noise or fading can be applied, which will relate to real world interferences. Some interferences experienced in real life transmission are, e.g., thermal noise, distance fading, doppler effect and reflection of signals. To approach those kind of interferences there are many different channel models, like the AWGN channel or Rayleigh/Rician fading. A further look in the AWGN channel and Rayleigh fading will be given. A small graphic will further illustrate the usual culprits for degradation of signal power and resulting loss in communication performance (Figure 1.4).

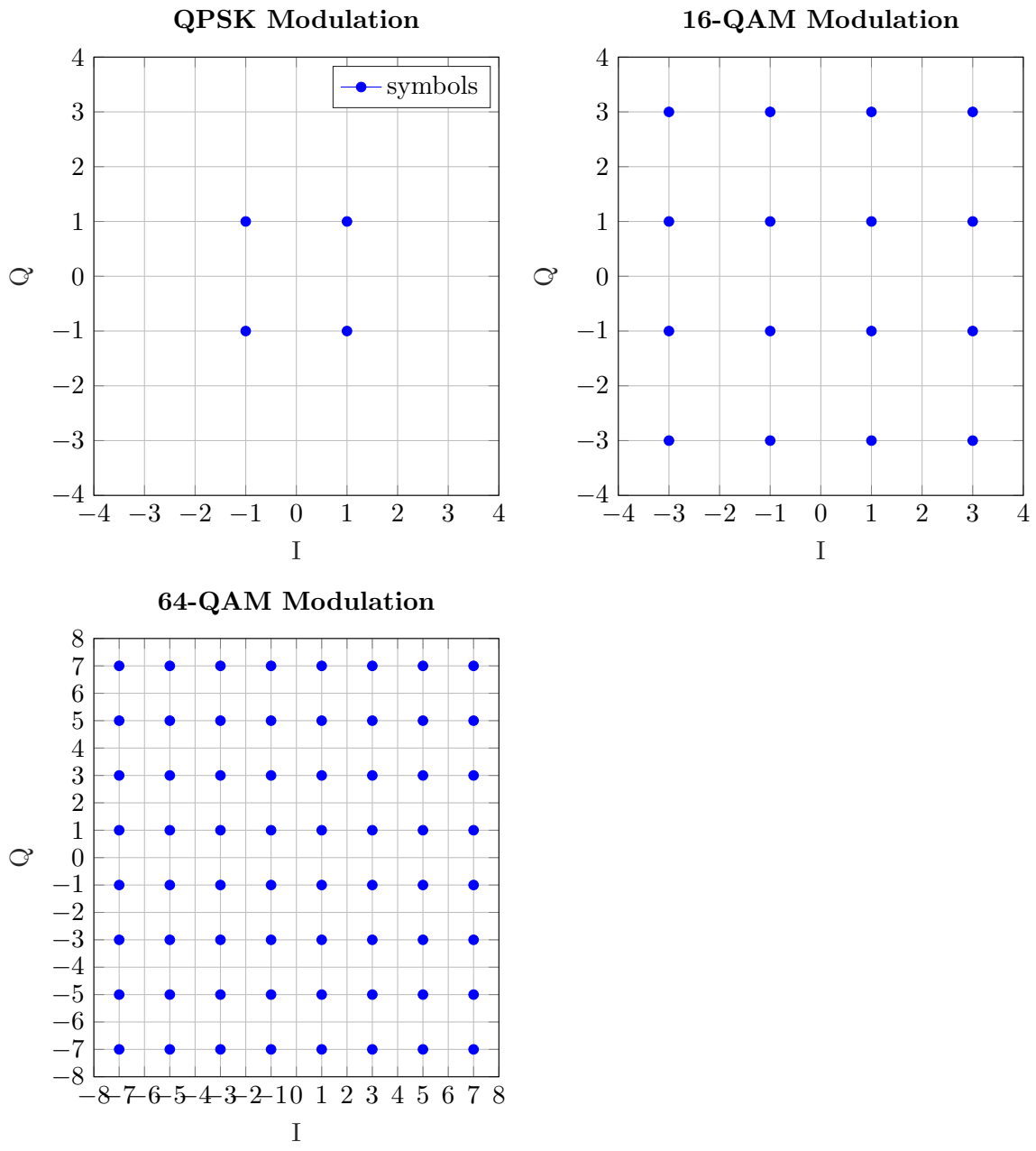

Figure 1.3: Modulation in I/Q planes for QPSK, 16-QAM and 64-QAM

1.4.1 AWGN Channel

The easiest kind of channel manipulation is to add random Gaussian noise to the channel, also commonly known as an AWGN channel.Like the name says we will add noise, which is a random Gaussian distribution with flat spectral density, to an existing transmitted

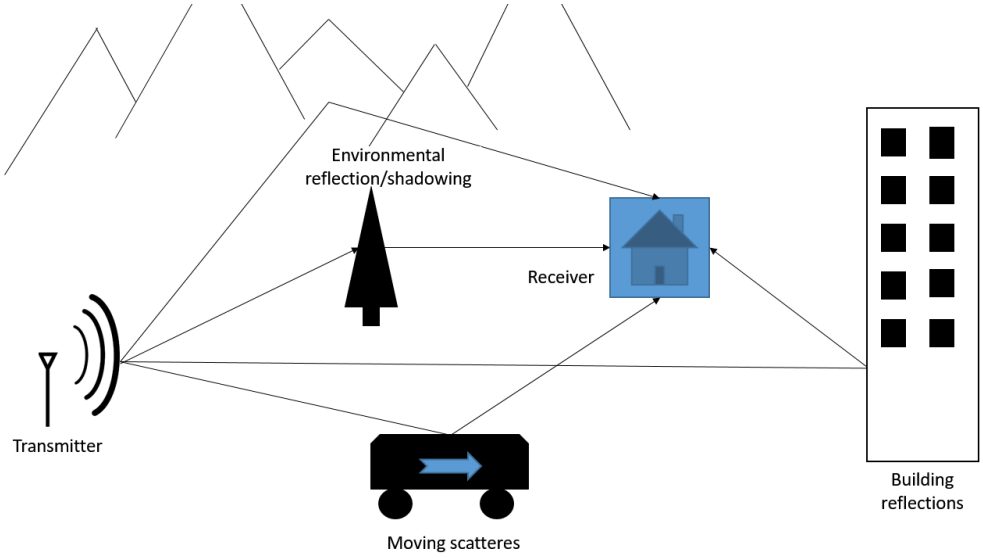

Figure 1.4: Interferences in a normal transmission between two devices

signal. Our receiver will receive a signal like this:

$$Y = X + N, \tag{1.1}$$

with Y being the received symbols, X the send symbol and N the complex AWGN noise. This can be applied for a full transmission of messages resulting in:

$$\underline{Y} = \underline{X} + \underline{N}, \tag{1.2}$$

which can be depicted more detailed like this:

$$[Y_1, Y_2, ..., Y_n] = [X_1, X_2, ..., X_n] + [N_1, N_2, ..., N_n] \tag{1.3}$$

The probability density function is defined as follows:

$$f(y|x) = \frac{1}{\pi\sigma^2} * e^{-\frac{(y-x)^2}{\sigma^2}}, \tag{1.4}$$

with y being the acquired point,x the send out symbol and σ^2 the variance of the distribution. Gaussian noise, representing thermal noise and overlay with multiple users in a wireless system, is therefore used in all the simulations run in this thesis. Below in Figure 1.5 a depiction of the spectral power distribution of AWGN. It ban be clearly seen that it is flat and spread evenly over the whole spectrum.

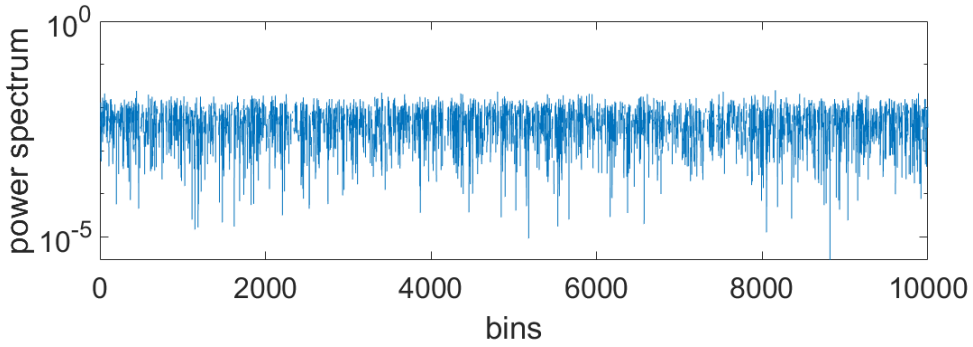

Figure 1.5: Power spectral density for a AWGN channel

1.4.2 Rayleigh Channel

Another common channel model used in communication theory is Rayleigh fading. Rayleigh fading simulates multipath reception, which means that for a receiver antenna in a wireless link there are many reflected and scattered signals reaching it (Figure 1.4). These kind of reflections are often seen in high-density urban areas. This results into construction or destruction of signal waves. The channel will now look like this:

$$Y = H * X + N, \tag{1.5}$$

which adds the new Rayleigh fading coefficient H. !!footnote explaining H!!
As done before for the AWGN channel the above equation can be expanded for a full transmission. Before that it needs to be clarified that not every symbol will be multiplied with a different fading coefficient H, but a block of symbols. This kind of transmission with Rayleigh fading is known as block fading:

$$[Y_1, Y_2, ..., Y_T] = H * [X_1, X_2, ..., X_T] + [N_1, N_2, ..., N_T], \tag{1.6}$$

with the subscript T indicating the length of the block. The block length can be chosen ranging from on single symbol up to the whole code word being one block. The pdf according to the calculations above is:

$$f(y\sigma) = \frac{1}{\sigma^2} e^{-\frac{y^2}{2\sigma^2}}, \tag{1.7}$$

The graphic (Figure 1.6) shows the power distribution over 12000 samples. Being Gaussian randomly distributed there are now these so called "deep fadings" where the power of the fading drops, which will also decrease tehe signal power of the received signal to

drop significantly. This results in the so-called burst errors, which were mentioned in Chapter (1.2).

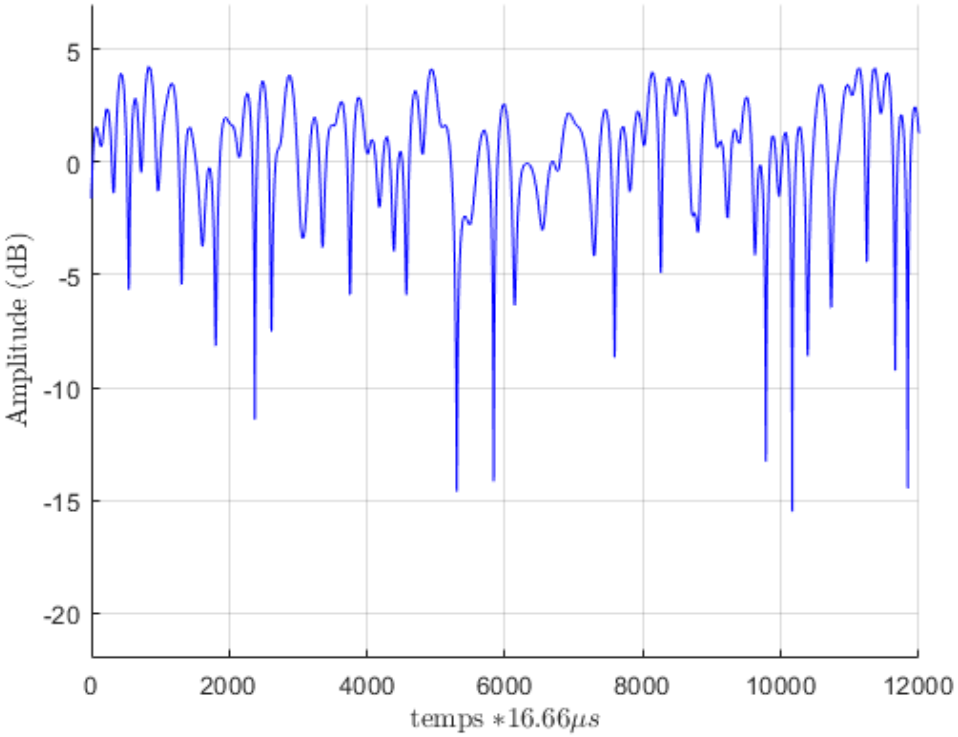

Figure 1.6: Power spectral density for a rayleigh channel

2 Capacity in AWGN Channel

In this section we will discuss the capacity of a wireless channel with AWGN noise interfering with the transmission between transmitter and receiver.
In general capacity C can be defined as the maximum data rate R at which information can be reliably transmitted over a channel, that means the highest rate of transmission with a very low error probability rate. It is proven that any rate exceeding the maximum capacity rate of the channel will result in error rates deviating from zero (!!cite!!). In modern technology with the help of smart modulation schemes and coding methods a rate close to the capacity can be achieved. All the capacities in the following simulations will be complex and also time-discrete. While time-continuous systems are analyzed for real world applications most systems can be converted into a time-discrete model with same capacity results. The transformation will also be shown in the following sections.

2.1 Capacity and Monte-Carlo-Simulation

For a AWGN-Channel the simple channel model already defined in Equation (1.1) will be used:

$$Y = X + N, \tag{2.1}$$

with X $\sim N(0,{\sigma_\mathrm{T}}^2)$ and N $\sim N(0,1)$. The received signal Y will have a distribution of Y $\sim N(0,{sigma}^2+1)$ under the condition that X and N being independently distributed. Now the capacity as a maximum of mutual information I between X and Y will be calculated:

$$C = max(I(X;Y)), \tag{2.2}$$

with X and Y being to independent randomly normal distributed variables.
For the mutual information further calculations will lead to the differential entropy:

$$I(X;Y) = h(Y) - h(Y|X) = h(Y) - h(N), \tag{2.3}$$

with N also being independent from X.
Which further simplifies to

$$h(Y) = h(X+N) = log(\pi * e^{\sigma^2+1}) \quad \text{and} \quad h(Y|X) = h(N) = log(\pi * e^1), \tag{2.4}$$

which will lead us to the final equation for the capacity in an AWGN-channel:

$$C = log(1+\sigma^2) \tag{2.5}$$

With this approach a good approximation of values for further calculations with added modulation schemes has been given. The above calculated data rate can be used as upper bound for any further capacity calculation done, this means there should be no capacity rate, especially for real world applications, exceeding this capacity rate (Equation (2.5)).

2.2 Capacity for QPSK and M-QAM

Now the capacity will be calculated for the three above mentioned modulation schemes (Chapter (**??**)). The schemes will be plotted with the capacity calculations for the AWGN channel to give us a overall comparison and overview.
Before the simulation or any calculation is done it can already be assumed how QPSK will behave for high SNR. As stated before QPSK (Chapter (**??**)) can transmit up to 2 bits per symbol, but no more without losing its reliability in transmission. So the plot will approach the 2 bit per symbol border for high SNR. The same assumption can be applied to both 16-QAM and 64-QAM. After creating a random codeword modulated with the fitting modulation scheme, noise is added to the signal, which is than received as the bit stream Y. The next step is to calculate the capacity.
Starting with the mutual information again we get:

$$I(X_Q;Y) = h(Y) - h(Y|X_Q) = h(Y) - h(N) \tag{2.6}$$

It is known that the signal is normal random distributed variable and now the differential entropy for h(Y) has to be calculated:

$$h(Y) = \int p(y) * [-log(p(y))]\mathrm{dx} \tag{2.7}$$

Applying the Monte-Carlo-Simulation will simplify the above equation so it can be solved numerically, turning the time-continuous system into a time-discrete form. The Monte-

Carlo-Simulation will be further explained in the following chapter (Ch. 2.2.1).

$$h(Y) = \sum_{n=0}^{N}(-log(p(y_{\mathrm{n}}))) * \frac{1}{N}, \tag{2.8}$$

with p(y) being the probability of y for a normal distributed variable and N the number of iterations. Now p(y) has to be further defined as it takes in consideration the possible input and output pairs $x \in X$ and $y \in Y$. The sum over all these pairs is taken:

$$p(y) = \frac{1}{n * \pi} * \sum_{i=1}^{n}(e^{y-x_{\mathrm{i}}}), \tag{2.9}$$

with x being the constellation points and y the received symbol. Here we only need to watch out for the number of symbols in the modulation scheme. For QPSK we have a n = 4, 16-QAM n = 16 and 64-QAM n = 64.

2.2.1 Monte-Carlo-Simulation

Monte Carlo Simulation is widely used in stochastics to get solutions for random experiments. It is applied to solve analytical unsolvable problems numerically. MC is based upon the law of large numbers, which says that a large number of performing the same experiment will lead the average of the results close to the expected value. We take this as a base to get reliable results. The Monte Carlo simulations will be used for two calculations, once already used above for calculating the differential entropy and later once to calculate a theoretical Rayleigh fading curve out of the AWGN channel.

2.2.2 Implementaion in MATLAB

We now implement all the above mentioned equations in MATLAB and plot the results. The code has to take into consideration the modulation scheme used, the range of calculation (SNR) and the sample size N. The code runs through the SNR range, here ranging from -10 to 30dB in step sizes of 0.5dB. The sample size N also has to be large enough to apply the Monte-Carlo simulation reliably. In our simulation N has to be higher than 10000 samples to receive dependable results. First a random bit stream, functioning as the data, is initialized. The bit stream is passed through a modulator which assigns the corresponding symbol. We scale the created constellation point with the SNR value. After passing $\underline{X}$ through the complex AWGN channel the resulting symbols $\underline{Y}$ and $\underline{X}$ are put into equation (2.8) and equation (2.9). Done for every stepsize a resulting plot for

the capacity over SNR in decibel is created.

2.3 Results

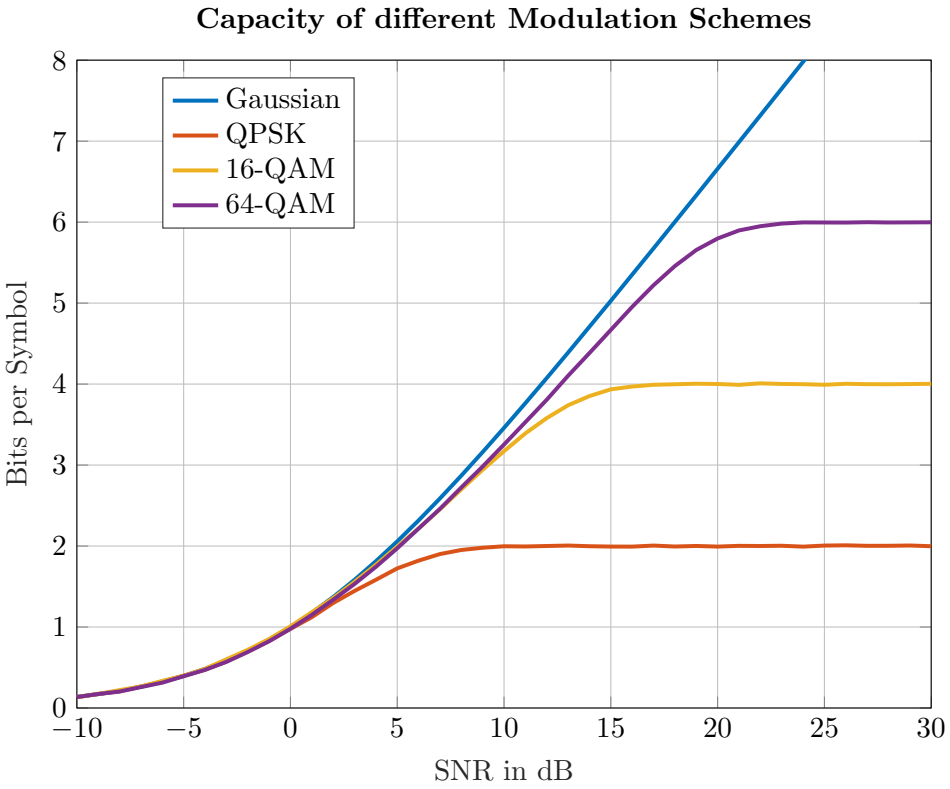

Figure 2.1: Capacity plot for general AWGN-channel, QPSK, 16-QAM and 64-QAM

The results of the calculation in MATLAB can be seen above. The plot shows the capacity for a simple AWGN channel and the three modulation schemes mentioned before (Chapter (1.3)). We plot the resulting capacity calculated over the corresponding SNR value. It can be seen that for all three modulation schemes that for higher signal-to-noise ratio (SNR) they approach the desired maximum number of bits per symbol. Also the Gaussian channel clearly outperforms the modulated channels, especially seen after 0 dB SNR. Noted before that the Gaussian capacity is the upper bound with no modulation exceeding it the first hurdle of confirming our simulation is overcome.

3 Frame Error Rate for AWGN Channel

This section will further built upon the topics discussed in the previous chapters. We will now build a communication chain using the blocks discussed in chapter (**??**). After the transmission the frame error rate (FER) will be determined. The FER is the estimated rate of faulty transmissions in one whole simulation. We will be sending frames of code words each consisting of 256 up to 2048 bits. After comparison between the sent code word and the decoded code word at the receiver we determine a frame error. If the decoded codeword has any wrong bit the whole frame will be marked as a faulty frame. This will be done for a certain amount of frames.

3.1 LDPC and the Coded Modulation Library

Already explained in Chapter 1 the whole transmission will be done with the coding method of LDPC. For the coding and decoding we will be using the Coded Modulation Library. This library and the functions used in this simulation will be further explained now.
The Iterative Solutions Coded Modulation Library (ISCML) is an open source toolbox for simulating capacity approaching codes in Matlab. (!cite!) The toolbox supports many different standard linear block codes and turbo codes. With many of complex and computational heavy codes implemented in C and ported back to MATLAB as so called C-mex functions. (!cite!)
For this thesis a further look will be taken at the WiMax LDPC code. The coder function will create the parity-check-matrix **H** with a given code word length n and rate r. The code word which will be sent will be created like this:

$$\boldsymbol{H} * \boldsymbol{a}^{\boldsymbol{T}} = \mathbf{0}, \tag{3.1}$$

with a being the code word transmitted. Further noted a consists of the relevant data a_d, which is known, and the unknown check nodes a_r.

$$\boldsymbol{a_r} = (\boldsymbol{H_k})^{-\mathbf{1}} * \boldsymbol{H_l} * \boldsymbol{a_n} \tag{3.2}$$

The decoding will be also be done by the Coded Modulation Library (CML) which again uses the parity-check matrix **H**. The received code word has to fulfill this condition:

$$\boldsymbol{H} * \boldsymbol{b^T} = \mathbf{0}, \tag{3.3}$$

with b^T being the received code word. This will be checked with typical graph solving algorithms. For LDPC the commonly used algorithm is the sum-product algorithm. (explain? a lot of computation and explaining...) With the CML the first and last block of the full communication chain (Figure 1.1) is implemented.

3.2 Demapping after AWGN

Another big part of a functioning communication chain is the estimation of the code word from the received symbol stream **Y**, which passed the channel experiencing various kind of noises and fadings. There are two main forms a restoring the code word, namely hard-decision demapping and soft-decision demapping.

3.2.1 Hard-Decision Demapping vs. Soft-Decision Demapping

We will now discuss the benefits between hard-decision and soft-decision demapping. Hard-decision demapping makes a decision based on the decision boundary of the received symbol. While soft demapping will take into consideration all symbol constellations in the modulation scheme. It is proven that soft demapping will achieve better demapping results while hard demapping is not as complex as soft demapping. (!!cite!!)In our case we do not mind a complex system which takes more time but are more focused on achieving the maximum rate of successful transmissions.

3.2.2 Log-Likelihood Ratio

A soft demapper will now be implemented in this system. The soft demapper will result in turning the symbols in the corresponding bit block.

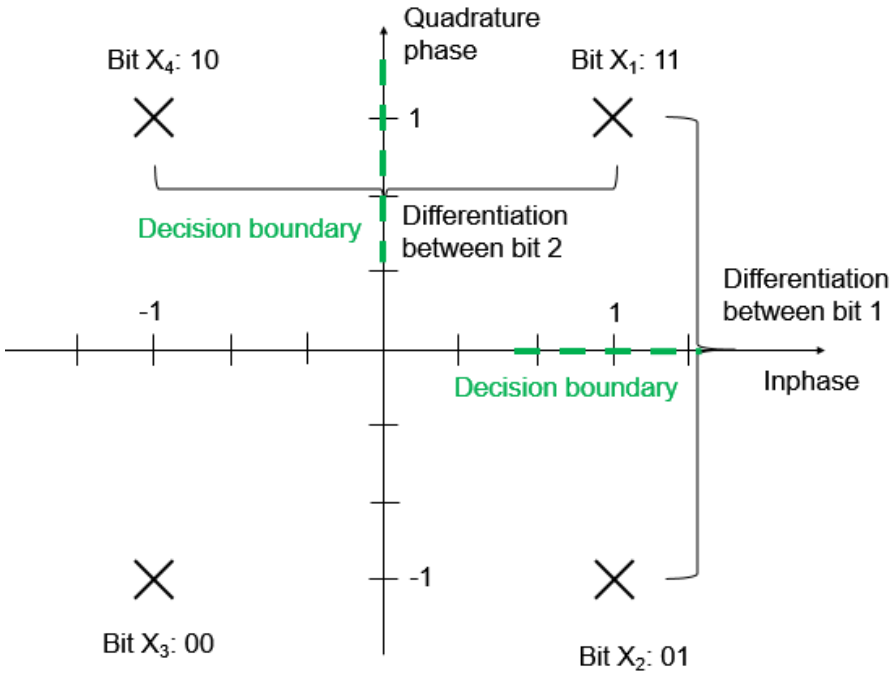

Figure 3.1: PDF $p_N(N)$ of the number N of times that the head side is up.

As seen in the figure above we have a QPSK modulated codeword. Symbols $\underline{\mathbf{Y}}$ received can be located anywhere on the I/Q-plane distorted by AWGN. We will now assume a single symbol Y received at the demapper. With QPSK consisting of two bit blocks $[B_1 B_2]$ first a differentiation for the first bit is done.

$$L^n = log \frac{P(B_1 = 0)|Y)}{P(B_1 = 0)|Y)} \tag{3.4}$$

With Baye's rule:

$$P(B_1 = 0|Y) = \frac{P(Y|B_1 = 0)}{p(Y)} * P(Y = 0), \tag{3.5}$$

the equation simplifies to

$$L^n = log \frac{P(Y|B_1 = 0)}{P(Y|B_1 = 1)}, \tag{3.6}$$

with $P(Y = 0) = P(Y = 1) = 0.5$. Getting back to Figure 3.4 it can be determined that X_1 and X_4 have $B_1 = 1$ while X_2 and X_3 result in $B_1 = 0$.

$$L^n = log\frac{P(Y|X_2) + P(Y|X_3)}{P(Y|X_1) + P(Y|X_4)} \tag{3.7}$$

With Equation (1.4) the log likelihood for bit B_1 can now be calculated. The corresponding bit B_2 will also be determined like this. In the end a code word is received as log likelihood ratios.

3.3 FER

The FER can now be determined with the help of the two previous sections. By dividing the number of faulty frames from the total number of frames we get the FER. It should be noted that not every FER calculated is a reliable result for our simulation. For a whole simulation run a certain amount of faulty frames is needed, usually at least 50 faulty frames need to be detected. It can be seen like this: A simulation run for 100 frames with 1 faulty frame does not give a confident result of a FER of 0.01. While a simulation ran for 10000 frames with 100 faulty frames will be seen as a more reliable result of a FER of 0.01.
With the proposition above for a FER of 10^{-4} we would need to run a simulation with 10^6 frames. Now the problem of simulation time length arises. If lower FER need to be calculated the number of frames will rise. Also while maybe feasible for short SNR-ranges some calculations will need longer SNR-ranges, which becomes a problem in the further chapters for fading channels. Both combined will make the simulation run rather long.
Two methods to reduce the simulation time are now implemented: The first one being a premature break of the simulation after reaching 100 faulty frames and dividing by the number of iterations ran. Even if running the whole simulation resulting in more precise FER, 100 faulty frames are enough for plotting a reliable result. Another technique is to increase the step size of the SNR between simulations, which will allow us to keep the SNR range at the expense of plot resolution. This can be mitigated by interpolating the results, which means we numerically create more data samples to increase the plot resolution.

3.4 Results

All the above sections will again be combined in a MATLAB script. (Maybe add the Algo Flow???) The simulation will be run for an SNR range of 0-10 SNR. Furthermore a FER of at least 10^{-3} should be calculated, which is a total of 10000 frames run. The following plot shows the FER plots for all three modulation schemes.
!!! find the capacity plots!!!!
Now with the FER calculated the capacity plots from chapter (2) can be used again for comparison. In this part the FER calculation are used by examining the interpolated data. Setting a threshhold at 10^{-3} FER the corresponding SNR value is noted down. The value is then plotted into the capacity plot. It has to be taken into account the rate of LDPC coding. The rate defines the amount of relevant data in a whole frame/code word. That means for a rate of 1/2 the transmission of relevant data is halved. For QPSK with a maximum of 2 bits per symbols and a rate of 1/2 a information value of only 1 bit per symbol can be achieved.
Below are the plots for the comparison between capacity and FER.

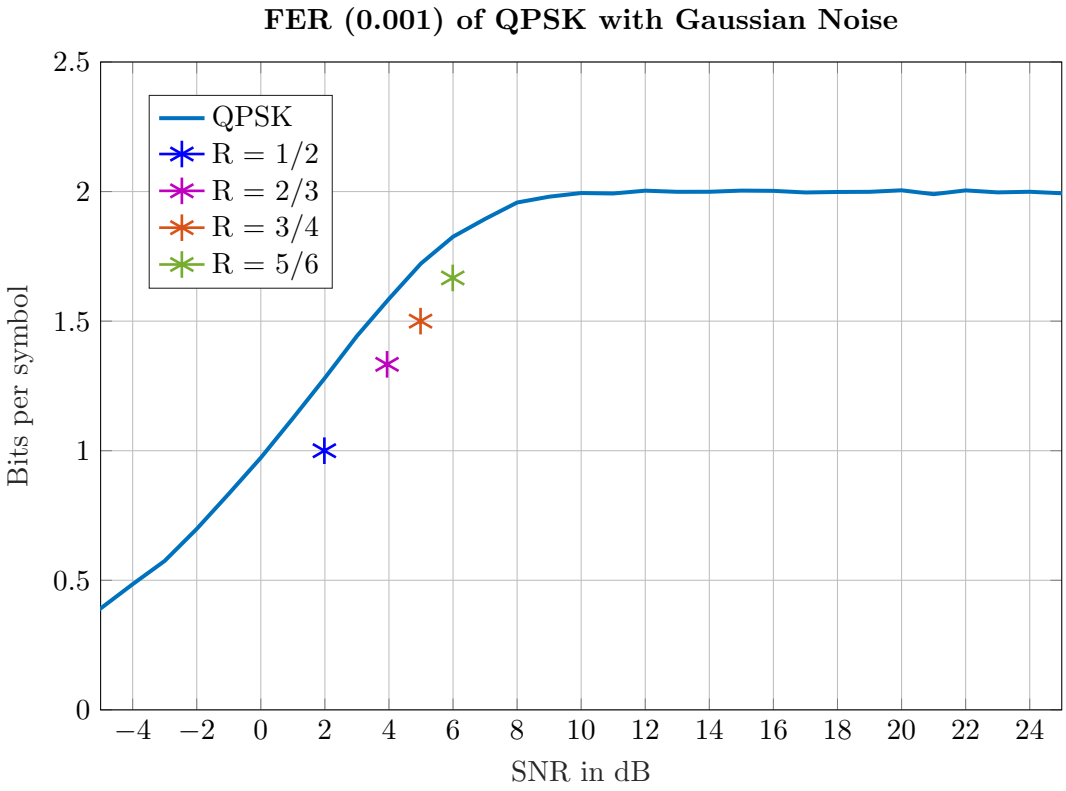

Figure 3.2: Capacity to FER comparison for QPSK

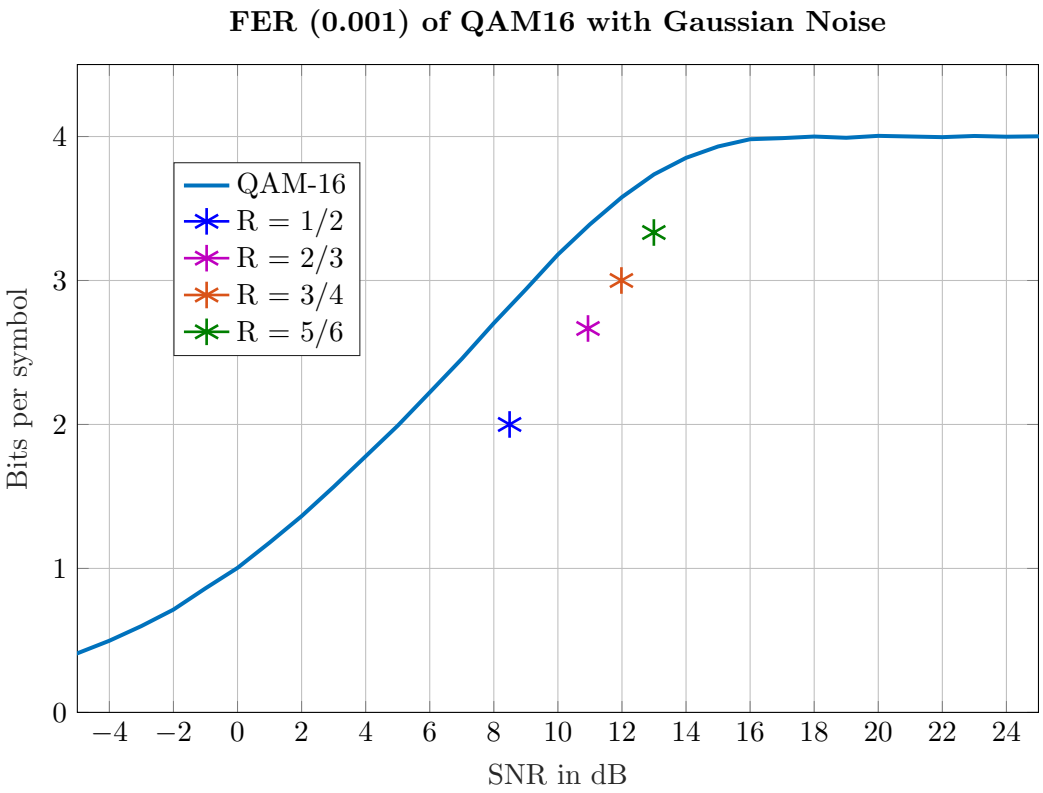

Figure 3.3: Capacity to FER comparison for 16-QAM

For all three plots the same procedure has been done. For all four available rates in WiMax coding the corresponding SNR value for the FER of 10^{-3} is plotted. It can be observed for all three plots that the for every transmission to achieve our desired FER we have an increase in SNR, which means that the transmission power needs to be higher. This is to be expected for a real communication system simulation with the simulated capacity being the upper bound of data rate that can be achieved.

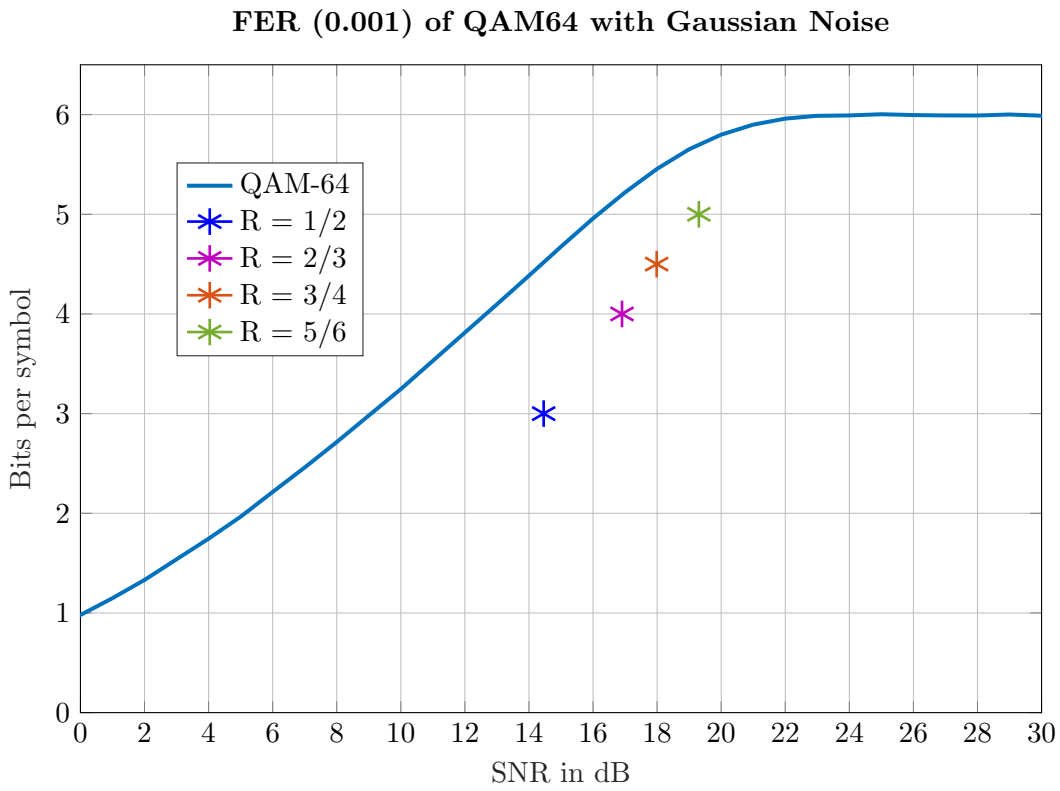

Figure 3.4: Capacity to FER comparison for 64-QAM

4 Capacity for a Rayleigh Channel

We will now discuss the above implemented simulation in AWGN with the addition of Rayleigh fading. There are many different fading processes that can be considered for simulation. In this thesis we will look at the so called block fading channel. In a block fading channel the fading coefficient H is constant over the block length $\mathbf{T}$. After every block the fading coefficient will change to a new independent value based on the distribution used. In our case the fading used is called Rayleigh fading, also mentionend before in chapter (**??**). For the whole block fading simulation 2 different scenarios will be considered:
First of all for both scenarios "Channel distribution information" (CDI) will be applied, which means that both for the transmitter and the receiver the distribution of the fading coefficient is known. Now for the first scenario, additionally to CDI, the knowledge of the fading coefficient power will be given to the receiver. This scenario is also known as "Receiver CSI", with CSI standing for "channel side information".
In the second scenario the information of the fading coefficient power will be unknown, which means a method is needed to try to estimate the coefficient as accurate as possible.

4.1 Fading Channel

(!! add fading channel picture!!) Different to the AWGN channel where only additive noise is added to the sent information signal in the fading channel before addition of noise the signal is scaled with the fading coefficient. Also mentioned in chapter (1.4.2) and the corresponding (1.5). (!!add figure with fading!! scatterplot) As seen in the figure above with various fading coefficients the overall power of the signal can increase or decrease. Which also results in a bigger/lesser interference of AWGN noise. Still it is important to know, while higher "power" is beneficial for the error probability in the system it is not needed to maintain stable information rate. At the same time a decrease in "power" will make the system suffer a loss in information. So in the end it is from utmost importance to try and reduce the fading from the received signal $\mathbf{Y}$.

4.2 Receiver CSI

We will now discuss recovering the signal with perfect channel knowledge.

$$\underline{Y} = H * \underline{X} + \underline{N}, \tag{4.1}$$

With the knowledge of the fading coefficient a simple division of the equation will solve the problem.

$$\hat{\underline{Y}} = \underline{X} + \hat{\underline{N}}, \tag{4.2}$$

with $\hat{\underline{Y}}$ being the new estimation of the received code word and $\hat{\underline{N}}$ the division of the noise with the fading coefficient. Although the fading has been removed from the initial sent code word the restored code word $\hat{\underline{Y}}$ has still to be considered a fading channel because of $\hat{\underline{N}}$. It is expected to have a decrease in performance in comparison to a normal AWGN channel.

4.3 Fading estimation with pilot symbol

In this section the scenario is taken into consideration that the receiver does not know the fading coefficient, but still has the information of fading block length. Now a method of restoring the code word $\underline{\mathbf{Y}}$ has to be found.
One common and simple method used is the addition of pilot symbols into the codeword. To specify this procedure look at the code word of length N. To this code word additional pilot symbols will be added. In this simulation only one pilot symbol per block will be added.

$$\underline{X} = [X_T, X_1, ..., X_N] \tag{4.3}$$

The pilot symbol X_T has a set constant value both known at the transmitter and receiver side. With this knowledge an easy estimate can be given for the fading coefficient for each block.

$$Y_T = H * X_T + N \tag{4.4}$$

$$\hat{H} = \frac{Y_T}{X_T}, \tag{4.5}$$

being an estimate because of the unknown portion of noise. With an increase in transmission power an increase of accuracy in the estimation of H can be given. A recovery of the received signal $\mathbf{Y}$ with the estimate $\hat{H}$ will be done. It is to be expected, that this method will result in a worse performance than the recovery with the perfect channel knowledge.

4.4 Results

This section will cover some cases simulated in this thesis. We will differentiate between different block sizes, (!!Add figure with perfect channel knowledge and pilot symbol!!)

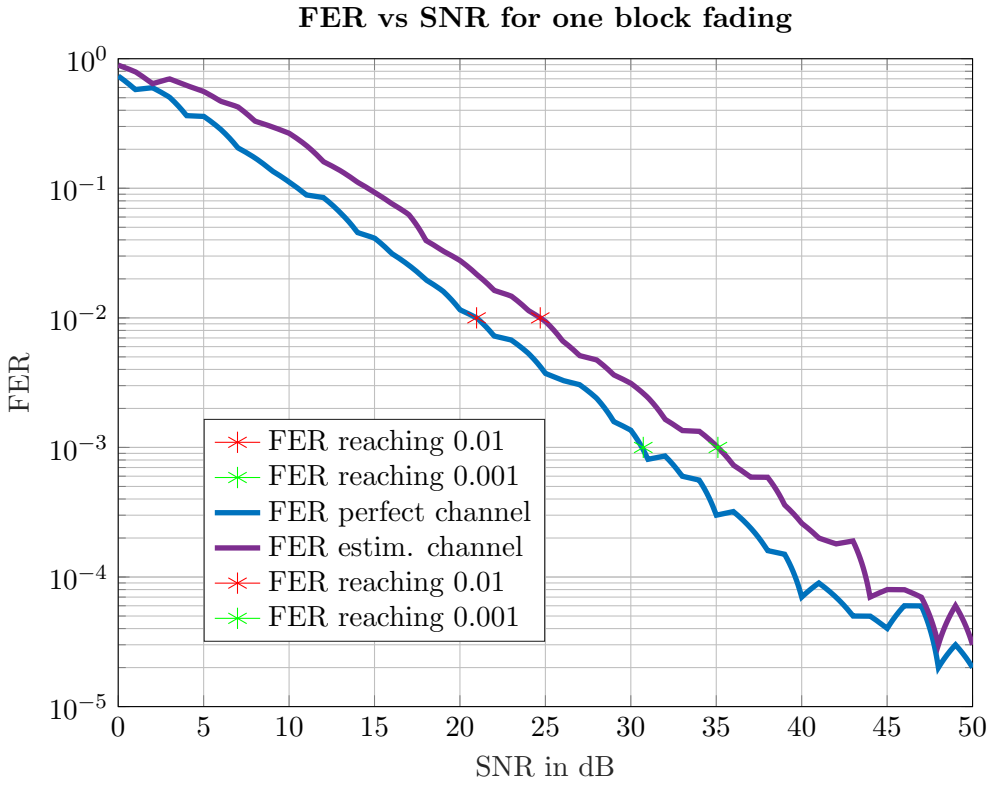

Figure 4.1: Simulation for rayleigh channel with known and estimated fading coefficient

As seen above the simulation with perfect channel knowledge clearly outperforms the one with unknown fading coefficient. To be exact for both the FER of 10^{-2} and 10^{-3} a improvement of about 3.5dB to 4 dB can be seen. This corresponds to more than the power consumption in the estimated channel to get the same performance than the perfect channel.

In the second figure different block lengths $\mathbf{T}$ are simulated for the channel with unknown fading coefficient. The block lengths here consists of the previous two plots and the new additional block lengths of two blocks and sixteen blocks per transmission. An observation can be given for low and high SNR behavior. For shorter block lengths, which means more blocks per transmission, the performance in low SNR is slightly worse while the performance in high SNR is slightly improved. For sixteen blocks per transmission this decline/improvement is not very noticeable.

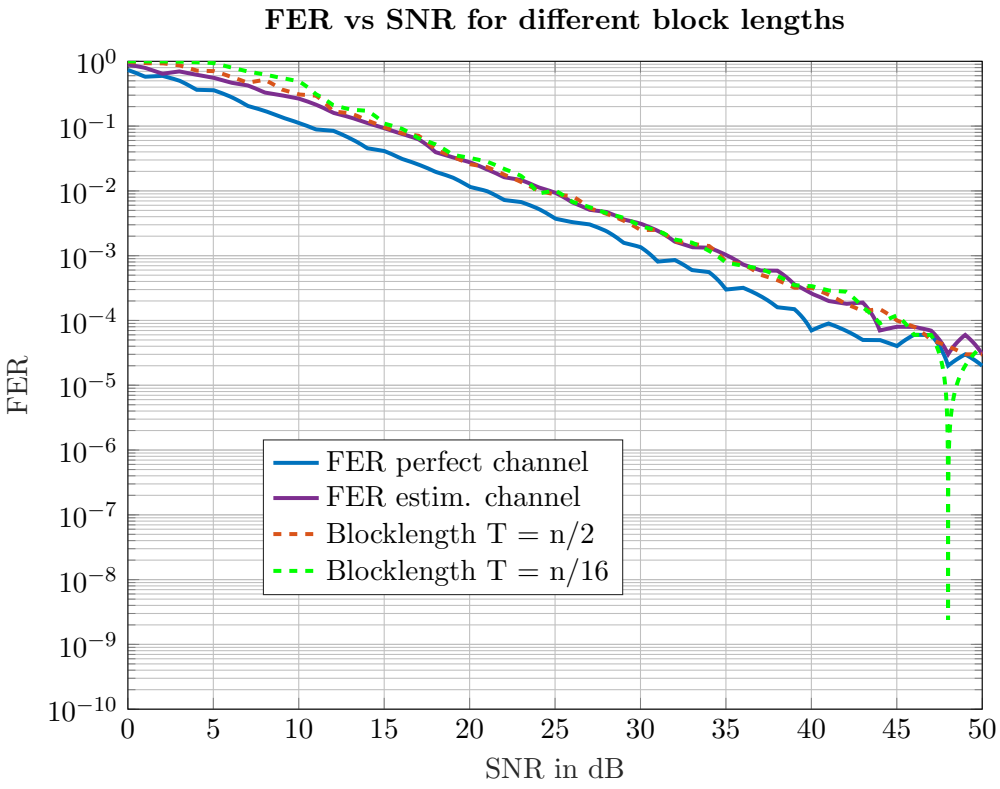

Figure 4.2: Simulation for rayleigh channel with different blocklengths

In the last figure the extreme case of 1 symbol per block will be simulated. That means **T** = number of transmission symbols. This kind of transmission from the view of viability is not feasible, because for every symbol a pilot symbol is added. With an addition for every symbol the whole transmission is slowed down by exactly half the rate, because overall twice the number of symbols need to be transmitted. But for this case the behavior for low and high SNR can be examined more precisely.
Here it is very clear that the newly simulated channel performs worse than the estimated channel for low SNR from 0dB upto 16dB, afterwards the channel outperforms the estimated channel and closely approaches the FER of the channel with perfect channel knowledge. It can be assumed that for low SNR the fading can be very strong, especially modulating every symbol independently it is to be expected that some of them will be modulated with very deep fading. The same independent single symbol fading can be an advantage for higher SNR. At the same time for higher SNR the occurence of deep fading will reduce. Of course some cases of deep fading will still happen, but only affecting one single symbol every time, this can be easily recovered by the decoder. If deep fading occurs for longer block lengths it can be very obstructing even in higher SNR. This can

be mitigated with the single symbol per block or just small block length transmission.

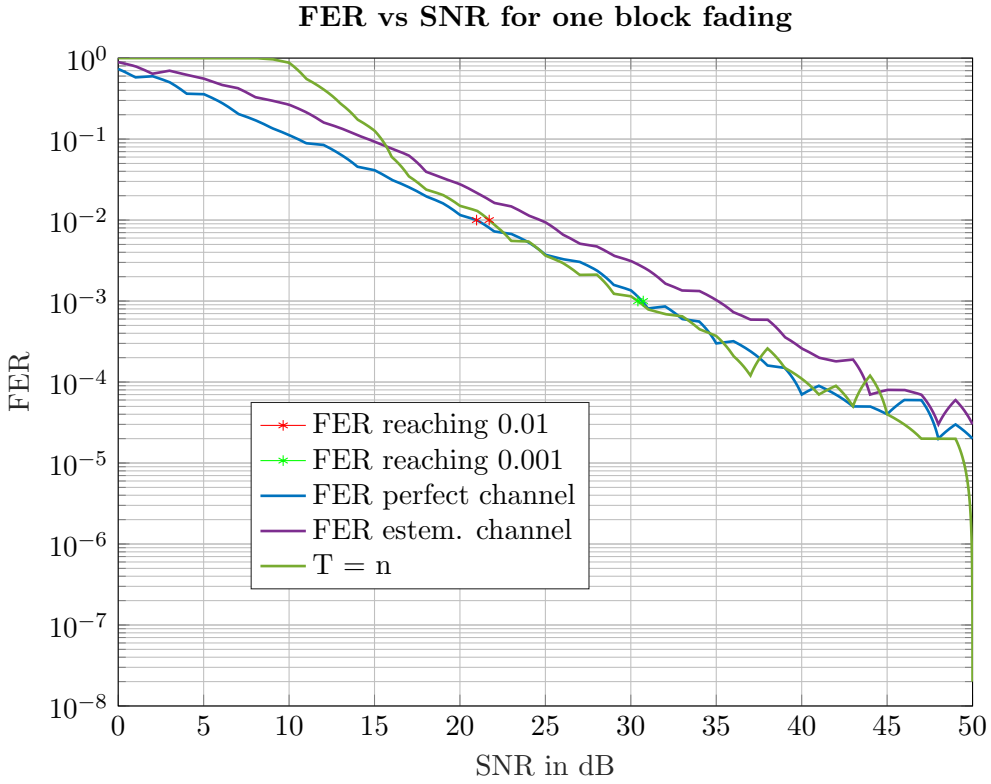

Figure 4.3: Simulation for rayleigh channel with blocklength = number of symbols in transmission

5 Experiments

In this next chapter we will discuss further simulations done to support the previous chapters. First of all a simulation to compare the FER simulated in chapter 5 will bbe initialized.

5.1 Simulated Rayleigh FER with AWGN channel

In this section a simulation based on the AWGN channel will be done to create a theoretical FER for a Rayleigh channel. This is done to have a reliable comparison for our simulation in Chapter 5. With the AWGN channel the reliability of the channel has already been proven with the capacity calculations from chapter 3. In chapter 4 the FER from the AWGN channel was in range of 0db upto 5 db SNR. In this region a valid FER was given with the number of frames set. For this simulation first of all an AWGN channel will be simulated for the given range of 0 to 5dB SNR. Then SNR from 0 to 50dB SNR will be initialized. The SNR will be modulated with complex gaussian noise to simulate the Rayleigh fading channel. For the now modulated SNR the corresponding FER can now be picked out. To get reliable results the Monte-Carlo-Simulation (MC) will be applied again. For every step of SNR many independent fading coefficients will be created. The mean of all the FER values will be taken to get the final data point plotted for the corresponding SNR value.

$$SNR_{ray} = SNR_{AWGN} * \mathbf{H} \tag{5.1}$$

$$FER_{ray} = FER_{AWGN}(SNR_{ray}) \tag{5.2}$$

In the plot above the FER for the AWGN channel is depicted. As mentioned above the valid range is only for 0dB upto 5 dB, which will be a problem for higher SNR values. For higher SNR values an error floor must be initialized. The error floor is the lowest probability the FER can take. It is proven that every transmission will have a small probability of failure, no transmission will have 0 FER which would mean that a transmission can always be without any errors. In practical transmissions this is not achievable. In this simulation we will also have a look into taking different values for the

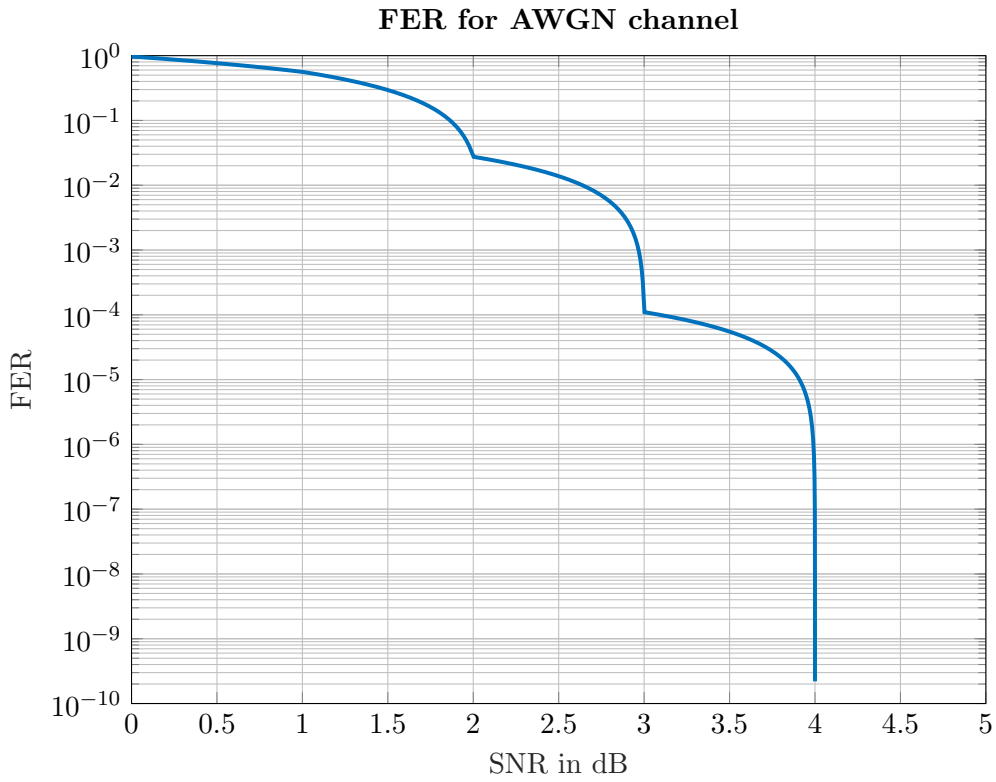

Figure 5.1: FER for a simulated AWGN channel

error floors, especially if a FER of 0 will result in better/worse performance. In this plot the FER of the AWGN channel is compared to the simulated FER of the rayleigh channel with error floors of 0 and 10^{-6}. It can be observed that rayleigh fading deteriorates the channel performace by a lot in comparison to the simple AWGN channel. Furthermore as seen in the graphic the value of the error floor also needs to be determined. There is a offset when simulating the FER for different error floors, with the 0 error floor achieving way better FER than the one with an error floor of 10^{-6}. Now comparing the simulated rayleigh fading based on the proven AWGN channel with our rayleigh fading from the rayleigh channel a statement can be given if the rayleigh channel was run properly. As seen in the figure above the simulated results based on the AWGN channel and the simulated rayleigh channel itself relate closely and behave the same way for different SNR values. With this section we proved that the simulated rayleigh channel is working as intended.

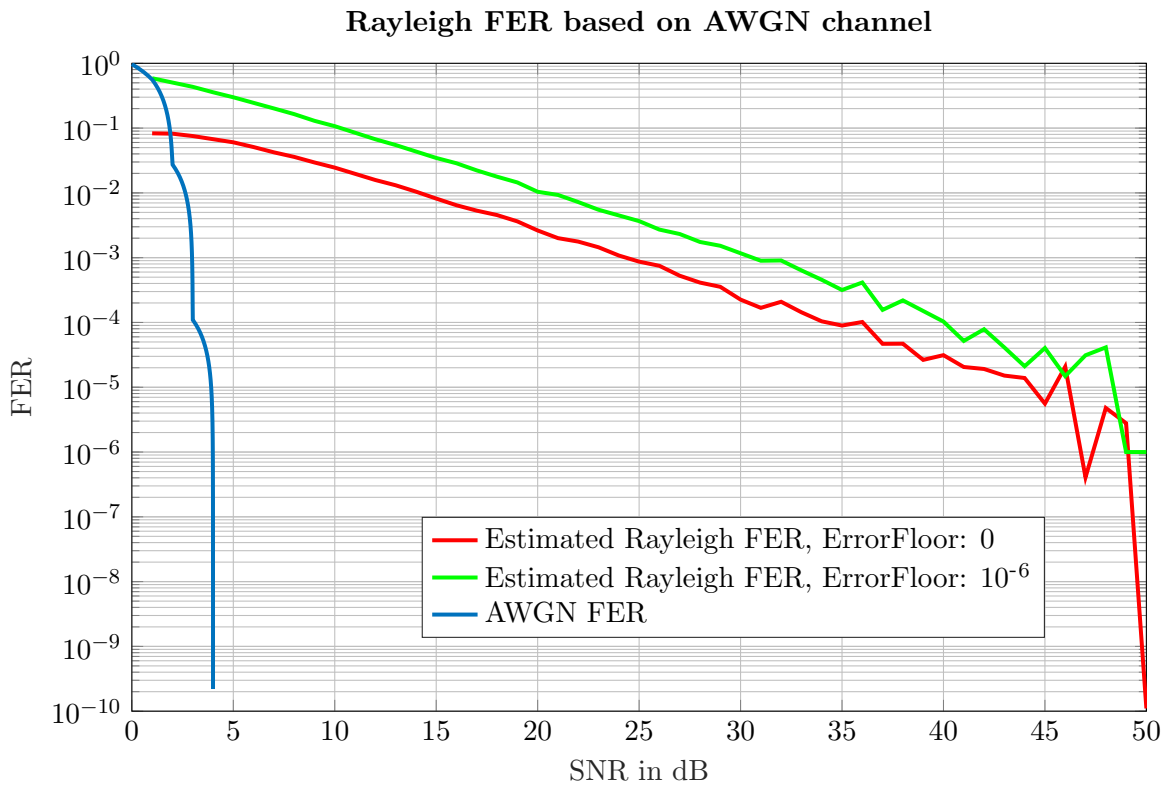

Figure 5.2: Comparison between AWGN FER and Rayleigh FER

5.2 Error floor calculation

As mentioned in chapter (5.1) the error floor should be determined to achieve proper results.

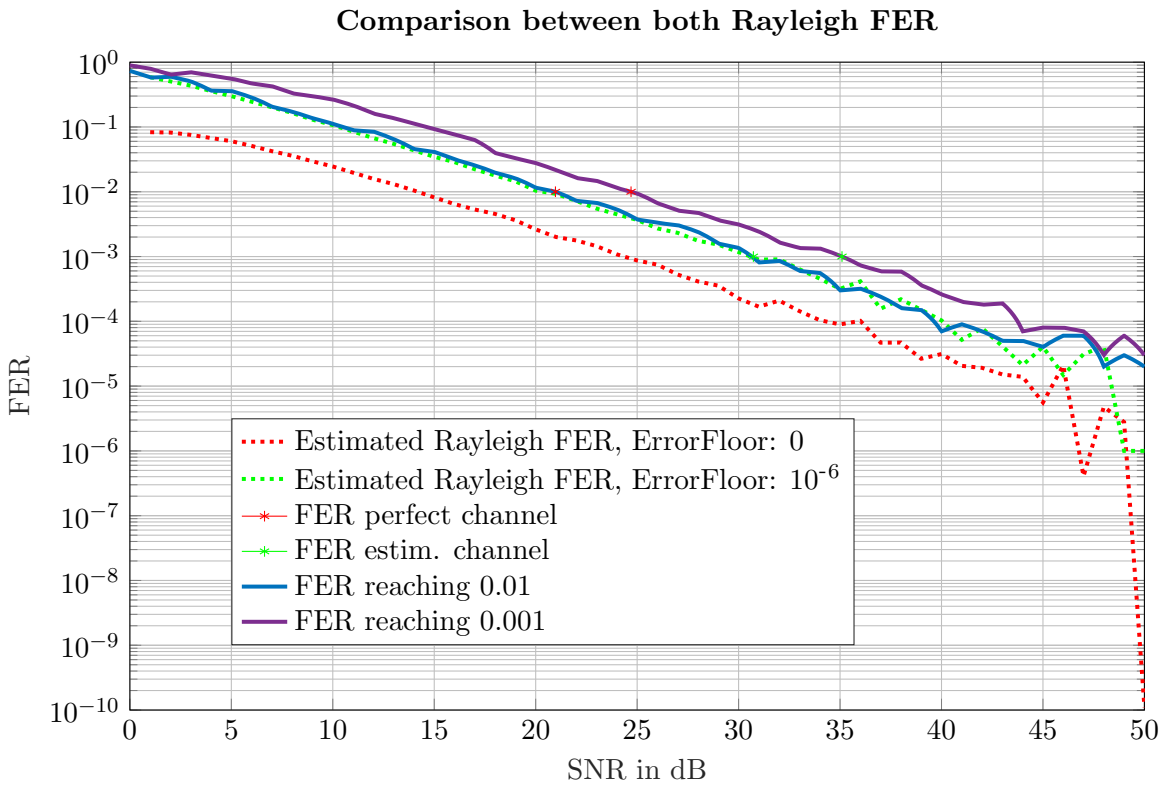

Figure 5.3: Comparison of rayleigh FER based on AWGN channel and rayleigh channel simulation

Bibliography

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *IEEE International Conference on Communications*, Geneva, Switzerland, May 1993.

[Che03] G. Chechik, "Types, super-types and the mutual information distribution," Leibniz Center, School of computer sciene and engineering, The Hebrew university of Jerusalem, Tech. Rep. 2002-61, 2003.

[Cla01] D. Clayton, "Population association," in *Handbook of Statistical Genetics*, D. Balding, M. Bishop, and C. Cannings, Eds. Chichester: John Wiley & Sons, 2001, pp. 519–540.

[CT91] T. Cover and J. Thomas, *Elements of Information Theory*, ser. Wiley series in telecommunications. New York: John Wiley & Sons, 1991.

[Hag01] J. Hagenauer, "Informationstheorie und Quellencodierung," Manuskript zur Vorlesung, Lehrstuhl für Nachrichtentechnik, Technische Universität München, 2001.

[Onk02] P. Onkamo, "Genetic mapping of complex traits: the case of type 1 Diabetes," Ph.D. dissertation, University of Helsinki, Helsinki, Jan. 2002.

[Pan03] L. Paninski, "Estimation of entropy and mutual information," *Neural Computation*, vol. 15, no. 6, pp. 1191–1253, Jun. 2003.

[Sac92] L. Sachs, *Angewandte Statistik*, 7th ed. Berlin: Springer-Verlag, 1992, ch. Die Auswertung von Mehrfeldertafeln, pp. 608–611.

[Sha48] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July, October 1948.

[Wei] E. Weisstein. Delta function. [Online]. Available: http://mathworld.wolfram.com/DeltaFunction.html