



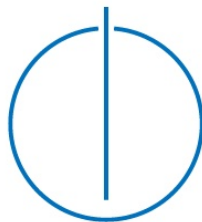
**Technische Universität  
München**

**Fakultät für Informatik**

**Master's Thesis in Informatik**

An Email-Centered Approach to Intelligent Task Management  
Using Crowdsourcing and Natural Language Processing

John Doe





**Technische Universität  
München**

**Fakultät für Informatik**

**Master's Thesis in Informatik**

An Email-Centered Approach to Intelligent Task Management  
Using Crowdsourcing and Natural Language Processing

Ein Email-basierter Ansatz für intelligente Aufgabenverwaltung  
mit Hilfe von Crowdsourcing und Natural Language Processing

**Author:** John Doe

**Supervisor:** Prof. Dr. Johann Schlichter

**Advisor:** Dr. Wolfgang Wörndl

**Submission:** DD.MM.YYYY

I assure the single handed composition of this master's thesis only supported by declared resources.

München, DD.MM.YYYY

*(John Doe)*



## **Abstract**

English abstract.

## **Inhaltsangabe**

Deutsches Abstract.

# Contents



## List of Figures





# Chapter 1

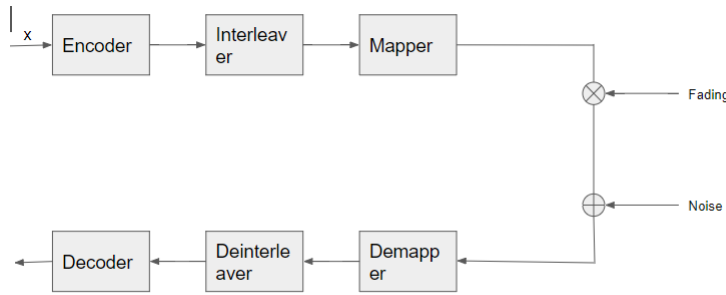
## Introduction

### 1.1 Motivation

### 1.2 Contributions

# Chapter 2

## Channelmodel



**Figure 2.1:** Channelmodel for general Transmitter/Receiver Chain

For my purpose we are looking at a simple model which will include a Mapper/Demapper, Encoder/Decoder, Interleaving, and the needed channel. We will briefly go over all blocks depicted in the graphic above.

### 2.1 Encoder/Decoder

For our Encoder/Decoder block we will be looking at a WiMax LDPC code according to the standard IEEE 802.16e. This standard code is used in small and medium distances in urban areas and fits our model quite well. LDPC, which stands for Low Density Parity Check, is a linear error correction code. While it has heavy computing demands, with the growth of computing power it sees more and more use in everyday use. In our case with WiMax we have different given block sizes ranging from 576 codewords up to 2304. The rates given are 1/2, 2/3, 3/4, and 5/6. We also only look at coding class A for our simulations. With a given codeword  $x$  of length  $n$  and a generator matrix  $G = [I^T | P]$ . The

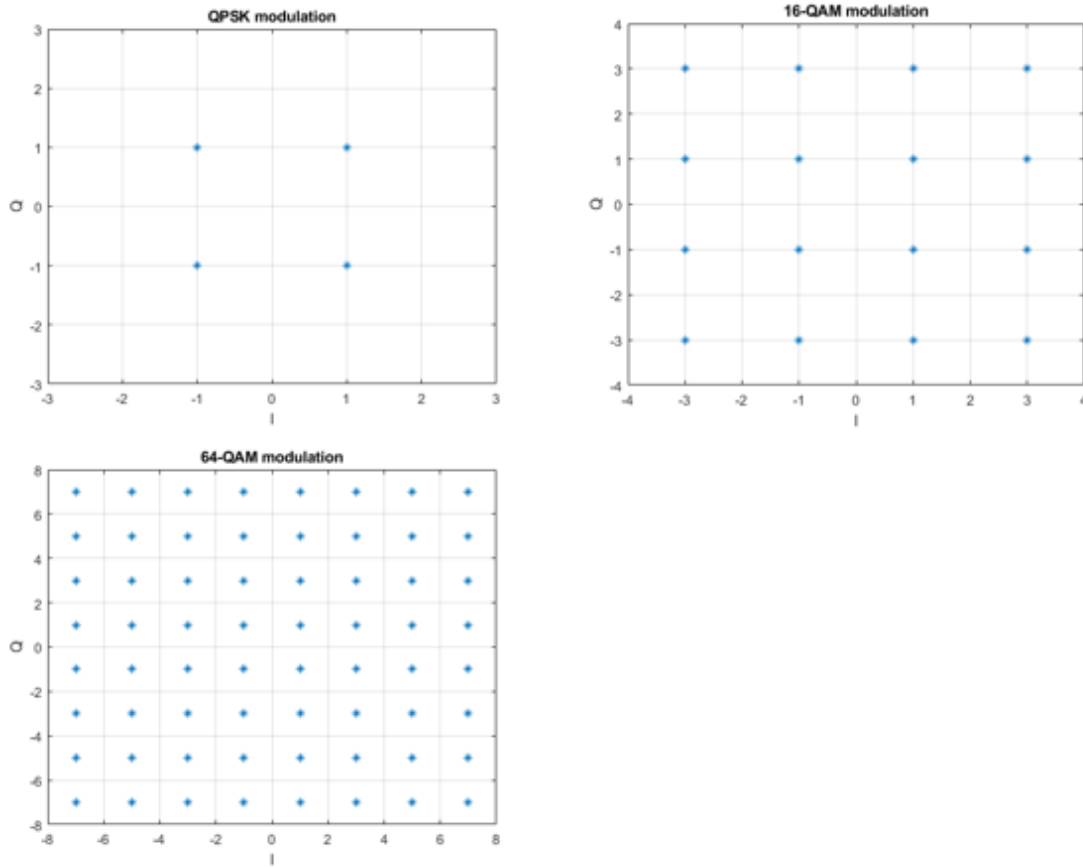
parity check matrix  $H$  can now be derived as  $H = [-P^T | I_{n-k}]$ . With the parity check matrix  $H$  and a code  $C = xG$  the condition for  $cH^t = 0$  must be fulfilled for the codeword to be valid. This whole process in MATLAB can be computed with the help of the Coded Modulation Library (CML). For this we have the given function "*InitializeWiMaxLDPC*" to create the parity-check.matrix, "*LdpcEncode*" and "*MpDecode*" to encode and decode our codeword.

## 2.2 Bitinterleave/Deinterleaver

With the help of bitinterleaving we can avoid any kind of "bursterrors", i.e. we avoid any longer blockerrors. The interleaving in MATLAB is accomplished by creating a random permutation array  $k$ . The permutation array can be used to randomly shuffle our codeword and also return them back to default at receiver side.

## 2.3 Mapper/Demapper

The mapper or modulation is used to assign a specific codelength a symbol which is transmitted. The symbols are located in a real/imaginary plane. With the distance from the nullpoint of the axis giving the amplitude of our signal and the angle to the real axis the phase shift. There are many forms of modulation schemes, with the most common ones being M-PSK, M-FSK, M-AM and M-QAM. For our simulations we will have a further look in QPSK, 16-QAM and 64-QAM, which are depicted below.



**Figure 2.2:** Modulation in I/Q planes for QPSK, 16-QAM and 64-QAM

QPSK, which stands for Quadrature Phase shift keying is one of the easier models we will be looking at. The symbols all share the same amplitude and only differ in their respective phase angle. With the information entropy  $S = \log_2(M)$  we can identify the maximum number of bits we can assign in every symbol, with M being the number of symbols in the modulation scheme. So for QPSK the number of bits per symbol amounts to 2.

With M-QAM, Quadrature Amplitude Modulation, we add the phase shift already

implemented into QPSK the additional differentiation with the amplitude of symbols. For QAM we send signals which differ in their phase shift and also the amplitude. For 16-QAM we get a maximum of 3 bits per symbols and for 64-QAM 4 bits per symbol. The modulation schemes makes it possible to increase our rate of transmission and is used for any kind of practical transmission of information.

## 2.4 Channel

The channel can be modified in many different ways. We can different sources of noise or fading, which can relate to realworld interferences. Some interferences experienced in real life transmission are e.g. thermal noise, distance, doppler effect and reflection of signals. To approach those kind of interferences there are many different channel models in simulations, like an AWGN-Channel or Rayleigh/Rician fading. We will havea further look into the AWGN-Channel and the Rayleigh fading.

### 2.4.1 AWGN-Channel

The easiest channel manipulation is to add random gaussian noise to the channel, also commonly known as AWGN-Channel (Additive White Gaussian Noise). Like the name says we will add noise which is randomly distributed in a gaussian distribution. The probability density function is defined as follows:

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

With  $x$  being the aquired point,  $\mu$  being the mean or expection of the distribution and  $\sigma^2$  the variance of the distribution. For our gaussian noise we will take a mean of 0 and a variance of 1, which will simplify further calculations in the following chapters. We will also always look at complex gaussian noise in our simulations. More or less every communication link will have some kind of gaussian noise interference, so we will add the AWGN-Channel to every simulation we run.

### 2.4.2 Rayleigh-Channel

Another common channel model used in communication theory is Rayleigh fading. Rayleigh fading is used to simulate multipath reception, which means that for a receiver antenna in a wireless link there are many reflected and scattered signals reaching it. This results into construction or destruction of waves. Rayleigh distribution can be defined like

this:  $R = \sqrt{X^2 + Y^2}$  with X and Y being two independent gaussian distributed random variables. Further calculations will lead to the following pdf:

$$f(x; \sigma) = \frac{1}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

# Chapter 3

## Capacity in an AWGN channel

We will now look into the maximum capacity we can achieve for our communication model in Chapter 2 with added AWGN noise.

### 3.1 Capacity and Monte-Carlo-Simulation

In general capacity  $C$  can be defined as the rate  $R$  at which information can be reliably transmitted over a channel, which means as long  $R \leq C$  we can achieve a transmission without errors even with noise. All the capacities we will be looking at will be for complex channel models.

For a AWGN-Channel we will have a simple channel model defined by  $Y = X + N$  with  $X \sim N(0, \sigma^2)$  and  $N \sim N(0, 1)$ . With this our 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. We will calculate the capacity as the maximum of mutual information  $I$  between  $X$  and  $Y$ :

$$C = \max(I(X; Y)) \quad (3.1)$$

with  $X$  and  $Y$  being to independent randomly normal distributed variables. With the maximum mutual information we calculate the maximum information we can achieve with the given parameters, like modulation, encoding, channel.

For the mutual information we can further part it into the differential entropy:

$$I(X; Y) = h(Y) - h(Y|X) \quad (3.2)$$

$$h(Y) = h(X + N) = \log(\pi * e^{\sigma^2 + 1}) \quad \text{and} \quad h(Y|X) = h(N) = \log(\pi * e^1) \quad (3.3)$$



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

$$C = \log\left(1 + \frac{\sigma^2}{N}\right) \quad (3.4)$$

With this approach we have good approximation values for further calculations with added modulation schemes. It is given that for only AWGN the capacity is at his maximum, there should be no capacity value over the calculated ones here.

### 3.1.1 Approach in Matlab

The above mentioned formula 2.4 will be simply implemented in MATLAB. With our noise being randomly distributed around 1 our formula simplifies even more into:

$$C_{\text{AWGN}} = \log(1 + \text{SNR}) \quad (3.5)$$

The SNR here must be transformed into power and not in decibel.

## 3.2 Capacity for QPSK and M-QAM

Now we will look into different modulation schemes, which were already mentioned in Chapter 1.3. We will implement these modulation schemes into our capacity calculations in an AWGN channel.

### 3.2.1 QPSK

For QPSK we will have 4 symbols and resulting 2 codebits per symbol. Before any simulation or calculation were run we can already be sure that we will not pass the upper bound of 2 bits/Symbol. So the plot will approach the 2 bits/Symbol for high SNR. After creating a random codeword modulated with the fitting modulation scheme. Noise is added to the signal, which is then received as the bit array Y. The next step to calculate the capacity differs from above.

We know that our signal is normal random distributed variables and we have to calculate the differential entropy for  $h(Y)$ , which is:

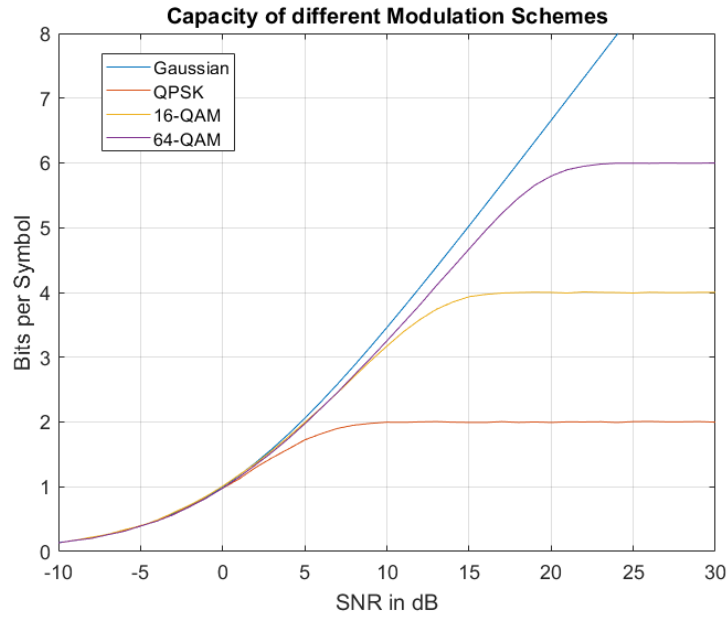
$$h(Y) = \sum_{n=0}^N (-\log(p(y_n))) * \frac{1}{N} \quad (3.6)$$

with  $p(y)$  being the probability of y for a normal distributed variable and N the codelength.

$$p(y) = \frac{1}{n * \pi} * \sum_{i=1}^n (e^{y-x_i}) \quad (3.7)$$

Here we 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  $n = 64$ . For the QAM modulation only the above mentioned  $n$  must be changed.

### 3.2.2 Implementation in Matlab



**Figure 3.1:** Capacity plot for general AWGN-channel, QPSK, 16-QAM and 64-QAM

The results of the calculation in MATLAB can be seen above. We can clearly see the modulated channel approaches the desired bit/symbol rate in a good speed. The gaussian channel clearly outperforms the modulated channels, clearly seen after 0 dB SNR.

## 3.3 Results and Conclusion

## Chapter 4

# Transmitter Receiver Chain in MATLAB

### 4.1 LDPC and the CML Library

### 4.2 FER and comparison with capacity plots

# Chapter 5

## Communication link for Rayleigh fading channels

### 5.1 Rayleigh fading and implementation

#### 5.1.1 Channel estimation

### 5.2 Rayleigh fading FER with AWGN channel

### 5.3 Problems and challenges

### 5.4 Results and comparison with AWGN channel

# Chapter 6

## Conclusion

6.1 Comparison between fading and AWGN channel

6.2 Fazit