

Diplomarbeit

Channel Adaptive OFDM Systems with Quantized Feedback

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Diplom-Ingenieurs unter der Leitung von

Univ. Prof. Dipl.-Ing. Dr. techn. Markus Rupp
Dipl.-Ing. Christian Mehlführer

E389
Institut für Nachrichtentechnik und Hochfrequenztechnik

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Dagmar Bosanska
Matrikelnummer: 0626949
Furdekova 5, 851 04 Bratislava, Slowakei

Wien, November 2008

I certify that the work presented in this diploma thesis was done by myself and the work of other authors is properly cited.

Dagmar Bosanska
Vienna, November 2008

Acknowledgements

I am particularly indebted to Christian Mehlführer who kindly shared his great knowledge with me. His constant advice and thorough proofreading resulted in many useful suggestions that helped a lot to improve this thesis with regard to both technical content and presentation.

I wish to express my gratitude to Professor Markus Rupp for his support, willingness and continual interest in the progress of this work. Also his bright ideas and suggestions are gratefully acknowledged.

My appreciation and thanks go to my parents. I can never thank my parents enough for their commitment, consistent encouragement and support. I can not imagine my life without their love and friendship.

I am also grateful to Pavol Svac for several enlightening discussions in which he generously shared with me his expertise on coding.

My thank goes to Veronika, Veronika (for their encouragement, understanding and making me happier at worse times, they know why I did not distinguish their names), Zuzana (for her friendship and going together through good and bad times), Janco (for believing in me and checking my working progress), Igi (he knows why), Markus and Manni (for their help and great times we had together) and to my friends on Kamossi Forum (for staying by my side even though I did not have enough time to be with them).

Finally and most importantly, I have been permanently supported by my fiance Peter. His sympathy, encouragement, love, and help have been vital for the completion of this thesis. I owe him more than anyone else.

Abstract

In this diploma thesis, I developed algorithms to improve the throughput of data networks based on single-input single-output (SISO) and multiple-input multiple-output (MIMO) wireless links. The aim is to use the spectrum more efficiently and flexible, while providing the required reliability of the transmission and adaptation of network behavior. These techniques are based on theoretically optimal solutions maximizing the throughput and are too demanding to be used directly in the practice. Therefore, some simplifications are needed and I evaluated their impacts on the network performance.

The optimal solutions are originally designed for uncoded systems. To overcome this shortcoming in more practical applications, I created a framework for coded systems with bit error ratio and packet error ratio adaptation. In this framework, the type of coding can be easily replaced once the theoretical formulas for the new modulation and coding scheme are known. Similarly, the error distribution modeling used in the packet error ratio evaluation can be further improved. The performance of this approach is evaluated in a WiMAX simulator and compared to the achievable channel capacity as a reference.

Contents

1	Introduction	1
1.1	OFDM Basics	2
1.2	Modularity of OFDM Systems	8
1.3	The WiMAX standard	9
2	SISO OFDM	12
2.1	System Model	12
2.2	Constrained Optimization Problem	14
2.3	Power and Bit Loading Algorithm for Uncoded Systems	16
2.4	Power and Bit Loading Algorithm for Coded Systems	22
2.4.1	Bit Error Probability for Convolutional Codes and M -QAM	23
2.4.2	Adaptation of Packet Error Ratio	39
2.5	Quantized Feedback	41
2.5.1	Greedy Power Allocation Algorithm with Quantization	43
2.5.2	Greedy Power Allocation Algorithm with Group Quantization	44
2.5.3	Entropy and Huffman Coding	44
2.6	Simulation Results	46
3	MIMO OFDM	51
3.1	System Model	51
3.2	Capacity of MIMO channels	53
3.3	Power and Bit Loading Algorithm for Coded Systems	56
3.4	Linear Pre-filtering	61
3.5	Quantized Feedback	64
3.6	Simulation results	65
4	Conclusions and Further Work	73

List of Figures

1.1	Multi-path propagation and its effects.	3
1.2	Principle behind Orthogonal Frequency Division Multiplexing.	4
1.3	Implementation effective IFFT/FFT processing.	7
2.1	Discrete-time equivalent baseband SISO OFDM model.	13
2.2	Resulting bit and power allocation across K subcarriers for a certain channel realization with SNR=16 dB and target bit error ratio $\text{BER}_0 = 10^{-5}$	19
2.3	Greedy Power Allocation Algorithm - way to combat frequency-selective fading (a certain channel realization with SNR=16 dB and target bit error ratio $\text{BER}_0 = 10^{-5}$).	20
2.4	Block diagram of a M -QAM modulator with a convolutional encoder.	24
2.5	Example of a trellis.	25
2.6	Uncoded BER approximation for M -QAM constellations.	30
2.7	The 16-QAM signal constellation with the choice of $a^{(0)}$	32
2.8	Performance analysis of 16-QAM with the half-rate convolutional code with $d = 4$ and conservative choice of $a^{(0)}$	33
2.9	Performance analysis of 64-QAM with the half-rate convolutional code with $d = 3$ and conservative choice of $a^{(0)}$	36
2.10	Performance analysis of 4-QAM with the half-rate convolutional code with $d = 6$	38
2.11	Frame error ratio of channel adaptive algorithms in a 1×1 Pedestrian B channel with PER adaptation ($\text{PER}_0 = 10^{-2}$) and uncoded BER adaptation ($\text{BER}_0 = 10^{-1}$).	47
2.12	Performance of channel adaptive algorithms in a 1×1 Pedestrian B channel with and without coding.	48

2.13	Bit error ratio comparison of the channel adaptive algorithms with PER and BER adaptation with the non adaptive system in a 1×1 Pedestrian B channel.	50
2.14	Achievable capacity and the non frame-based throughput with uncoded BER adaptation ($\text{BER}_0 = 10^{-1}$) in a 1×1 Pedestrian B channel.	50
3.1	Discrete-time equivalent baseband MIMO OFDM model.	52
3.2	Threshold metric $\{d_r^2[k]\}_{k=1}^K$ and a certain channel realization with SNR equal to 16 dB and target bit error ratio $\text{BER}^{(0)} = 10^{-5}$	60
3.3	A linear pre-filtering designed for a MIMO OFDM system by exploiting channel knowledge.	62
3.4	Optimized discrete-time equivalent baseband MIMO OFDM model with linear pre-filtering.	63
3.5	Performance of channel adaptive algorithms in an uncorrelated 2×2 Pedestrian B channel with PER adaptation (target packet error ratio $\text{PER}^{(0)} = 10^{-2}$).	68
3.6	Frame error ratio adaptation for the target packet error ratio $\text{PER}^{(0)} = 10^{-2}$	69
3.7	Performance of channel adaptive algorithms in an uncorrelated 2×2 Pedestrian B channel with PER adaptation (target packet error ratio $\text{PER}^{(0)} = 10^{-2}$) with and without quantization.	70
3.8	Performance of channel adaptive algorithm in an uncorrelated 2×2 Pedestrian B channel without coding (uncoded BER adaptation with $\text{BER}^{(0)} \in \{0.0001, 0.001, 0.1\}$).	72

List of Tables

1.1	<i>Types of access to a WiMAX network.</i>	11
2.1	<i>The Spectra $\beta(d)$ for $[133\ 171]_8$ convolutional code when applied to different M-QAM modulators.</i>	29

1 Introduction

From the first successful wireless transmission of telegraph messages by Guglielmo Marconi [1] in the 1890s, the wireless communication systems of today have evolved dramatically, but the road to the current status has been quite long and challenging. To appreciate the overall progress, let us for example consider how cellular systems have evolved from an expensive technology for a few selected individuals to today's global mobile communication systems used by almost half of the world's population [2]. Beside the cellular telephony, which is the biggest market segment, wireless computer networks have emerged in recent times by changing working habits (e.g. the possibility to answer e-mails anytime and anywhere). Moreover, there is also a large number of other applications that we do not come across in every day life, but they are starting to change our lives. Just few examples include:

- wireless sensor networks monitoring factories;
- cables between computer and keyboard, mouse and other devices replaced by wireless links;
- wireless positioning systems monitoring the location of trucks;
- goods identified by wireless RF (Radio Frequency) tags.

Current wireless technologies are expected to be broadband due to increasing demand for high data rates and they can support fixed, nomadic, portable and mobile access. To choose the right technology for a certain application, not only data rate and mobility have to be considered but also many other requirements, like range, power consumption, reliability, or number of users have to be fulfilled [3].

As the aim of the wireless design is not only to optimize performance for specific applications, but also to minimize cost, there will always be many challenges and possible improvements of the current wireless systems and networks.

1.1 OFDM Basics

As it was underlined, the wireless technologies and solutions vary for different applications. I focus on the possible ways to increase the total throughput in the wireless network. When the higher data rates have to be provided, implying as good coverage as possible, the transmission bandwidth should be at least of the same order as the data rates. Otherwise, providing higher data rates (for example by means of higher-order modulation) within a limited bandwidth is possible only in situations where relatively high signal-to-noise and signal-to-interference ratios can be made available. This is of course not possible to ensure for users in large-cells environments with high traffic load or at the cell border.

On the other hand, increasing transmission bandwidth relates to several critical issues [2]:

- Spectrum is a scarce and expensive resource, and it is often impossible to find spectrum allocations of sufficient size, especially at lower-frequency bands.
- The use of wider transmission and reception bandwidths has an impact on the complexity, power consumption and the price of the radio equipment, both at the base station and at the mobile terminal.
- The transmitted signal is significantly more corrupted when transmitted over wider bandwidth due to the time dispersion on the radio channel.

The last problem will be in the center of interest. The radio channel between transmitter and receiver consists of multiple propagation paths with different delays and magnitudes (see Figure 1.1(a)). These propagation paths also have various phases that depend on the path delay τ . At the receiver, the signals coming from different directions are simply added, so that they interfere with each other. Depending on the path phases, this interference may be constructive or destructive. Therefore, the total signal amplitude is changing due to the interference of the different paths. This effect is crucial in narrow-band systems and is called small-scale fading.

In systems with large bandwidth, and thus a good resolution in the time domain, the major consequence of the multi-path propagation is the time dispersion of the signal. In other words, the impulse response of the radio channel is not a single delta pulse, but rather a sequence of pulses with distinct arrival time and different amplitude and phase (Figure 1.1(b)). This time dispersion leads to the inter-symbol interference. In the

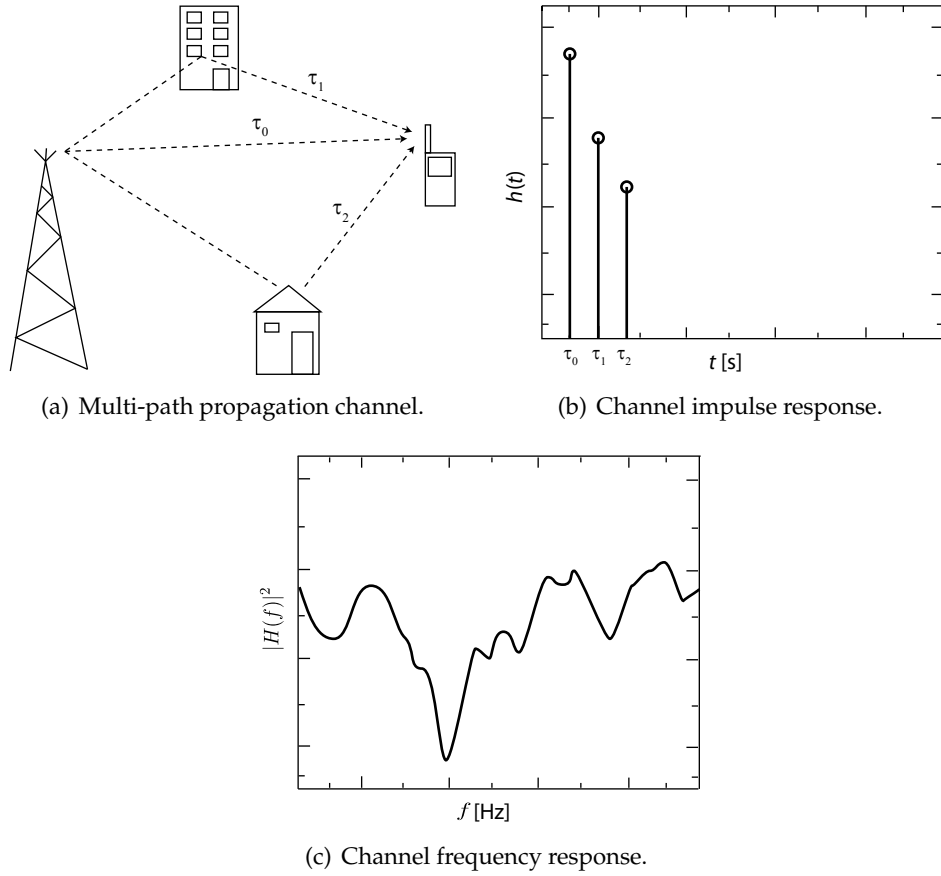


Figure 1.1: Multi-path propagation and its effects.

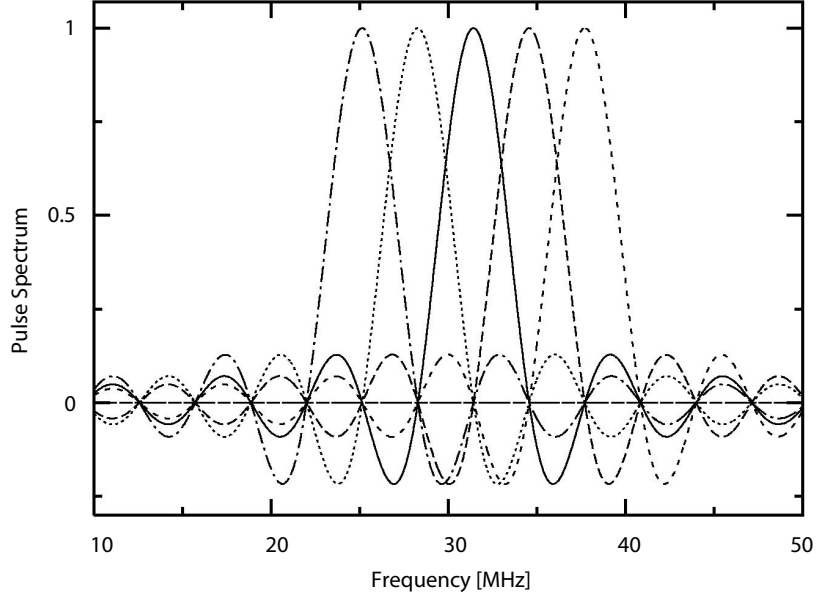


Figure 1.2: Principle behind Orthogonal Frequency Division Multiplexing.

frequency domain, a time-dispersive channel corresponds to a non-constant channel frequency response as illustrated in Figure 1.1(c). This radio-channel frequency selectivity will corrupt the frequency-domain structure of the transmitted signal and leads to higher error rates for given signal-to-noise ratios. The frequency selectivity has larger impact on wider-band transmission. The amount of radio-channel frequency selectivity also depends on the environment with typically less frequency selectivity (less time dispersion) in case of small cells and in environments with few interacting objects such as rural environments.

It should be noted that Figure 1.1(c) illustrates a snapshot of the channel frequency response. As a mobile terminal is moving through the environment, the channel frequency response may vary rapidly in time. The rate of the variations in the channel frequency response is related to the channel Doppler spread f_D , defined as $f_D = v/c \cdot f_c$, where v is the speed of the mobile terminal, f_c is the carrier frequency, and c is the speed of light. To counteract signal corruption due to the radio-channel frequency selectivity, the receiver-side equalization is needed. However, if the transmission bandwidth is further increased up to, for example 20 MHz, the complexity of the straightforward high-performance equalization starts to become a serious issue [2]. Therefore, the specific transmission scheme and signal design are required that allow for a good radio-link performance also in case of the substantial radio channel frequency selec-

tivity without a prohibitively large receiver complexity. One beneficial solution is the multi-carrier transmission called OFDM (Orthogonal Frequency Division Multiplex). It converts a high-rate data-stream into a number of low-rate streams that are transmitted over parallel, narrow-band channels that can be easily equalized. These channels are created by modulating N_{FFT} distinct subcarriers. In order to be able to separate the signals carried by different subcarriers, they have to be orthogonal. Let subcarriers be at the frequencies $f_k = kB/K$, where k is an integer, and B the total available bandwidth (in the simplest case equal to the nominal bandwidth $B = K/T_s$, where T_s is a per-subcarrier useful symbol duration or basically a useful OFDM symbol duration (without cyclic prefix that will be mentioned later). Furthermore, the modulation on each subcarrier is the Pulse Amplitude Modulation (PAM) with rectangular basis pulses:

$$g_k(t) = \begin{cases} \frac{1}{\sqrt{T_s}} \exp(j2\pi f_k t) = \frac{1}{\sqrt{T_s}} \exp(j2\pi k \frac{t}{T_s}) & \text{for } 0 < t < T_s \\ 0 & \text{otherwise.} \end{cases} \quad (1.1)$$

It can be easily seen that the subcarriers are mutually orthogonal, since the relationship

$$\int_{iT_s}^{(i+1)T_s} \exp(j2\pi f_k t) \exp(-j2\pi f_l t) dt = \delta_{kl} \quad (1.2)$$

holds. Due to the rectangular shape of pulses in the time domain, the spectrum of each modulated subcarrier has a $\sin(x)/x$ shape. Their spectra overlap, but the maximum of each carrier (sampling point) lies in the spectral nulls of all other carriers. This is depicted in Figure 1.2. Thus, the avoidance of interference between subcarriers is not simply due to a subcarrier spectrum separation, rather it is due to the specific frequency-domain structure of each subcarrier in combination with the specific choice of a subcarrier spacing Δf equal to the per-subcarrier symbol rate $1/T_s$ (from previous $f_k = kB/K = k/T_s$ and $\Delta f = f_{k+1} - f_k = (k+1)/T_s - k/T_s = 1/T_s$). Therefore, the proper demodulation at the receiver (multiplying by $\exp(-j2\pi f_k t)$ and integrating over OFDM symbol duration T_s) leads to no interference between any two subcarriers. This results in the natural visualization of the OFDM system as a bank of modulators/correlators, one for each subcarrier. These illustrations are, however, not the most appropriate modulator/demodulator structures for actual implementation, because the hardware effort of multiple local oscillators is too high.

Actually, due to OFDM specific structure and the selection of a subcarrier spacing $\Delta f = 1/T_s$, OFDM allows for low-complexity implementation by means of computationally efficient Fast Fourier Transform (FFT) processing. To confirm this [2], the

OFDM transmit signal is defined as

$$s(t) = \sum_{i=-\infty}^{\infty} s_i(t) = \sum_{i=-\infty}^{\infty} \sum_{k=0}^{K-1} a^{(i)}[k] g_k(t - iT_s) = \sum_{i=-\infty}^{\infty} \sum_{k=0}^{K-1} a^{(i)}[k] \exp(j2\pi k \Delta f (t - iT_s)), \quad (1.3)$$

where $a^{(i)}[k]$ is the, in general complex, modulation symbol applied to the k th subcarrier during the i th OFDM symbol interval, and where the basis pulse from Equation (1.1) were used with $f_k = k \cdot \Delta f$. Let us consider a time-discrete (sampled) OFDM transmit signal where it is assumed that the sampling frequency f_{sample} is a multiple of the subcarrier spacing Δf , i.e. $f_{\text{sample}} = 1/T_{\text{sample}} = N_{\text{FFT}} \cdot \Delta f$. The parameter N_{FFT} should be chosen so that the sampling theorem is sufficiently fulfilled¹. As mentioned before, $K \cdot \Delta f$ can be seen as the nominal bandwidth of the OFDM signal, this implies that N_{FFT} should exceed K with the sufficient margin. With these assumptions, the time-discrete OFDM signal can be expressed as²

$$\begin{aligned} s[n] &= s[nT_{\text{sample}}] = \sum_{k=0}^{K-1} a[k] \exp(j2\pi k \Delta f n T_{\text{sample}}) = \sum_{k=0}^{K-1} a[k] \exp(j2\pi kn/K) \\ &= \sum_{k=0}^{N_{\text{FFT}}-1} a'[k] \exp(j2\pi kn/K), \end{aligned} \quad (1.4)$$

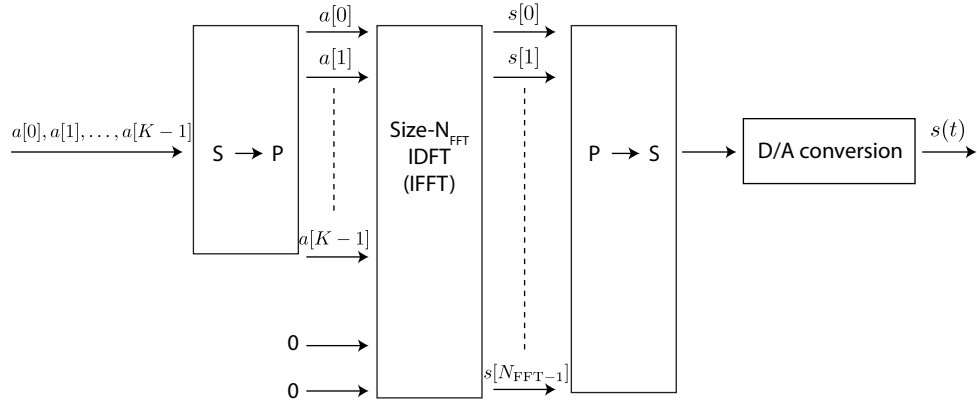
where

$$a'[k] = \begin{cases} a[k], & 0 \leq k < K \\ 0, & K \leq k < N_{\text{FFT}}. \end{cases} \quad (1.5)$$

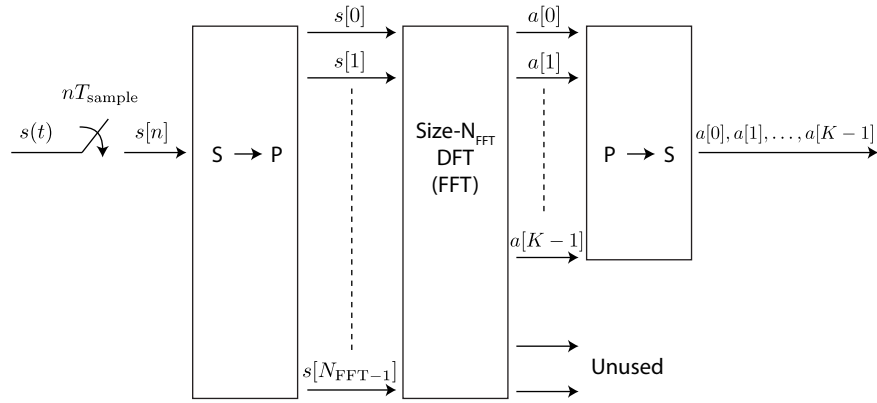
Thus, the sequence $s[n]$ is the Inverse Discrete Fourier Transform (IDFT) with size N_{FFT} of the block of modulation symbols $a[0], a[0], \dots, a[K-1]$ extended with zeros to the length N_{FFT} . OFDM modulation can thus be implemented by means of the IDFT processing followed by the digital-to-analog conversion, as illustrated in Figure 1.3(a). Especially, by selecting the IDFT size N_{FFT} equal to 2^m for some integer m , the OFDM modulation can be implemented by means of the implementation efficient radix-2 Inverse Fast Fourier Transform (IFFT) processing [2]. Similar to the OFDM modulation, efficient FFT processing can be used for the OFDM demodulation, replacing the bank of K parallel demodulators with sampling with some sampling rate $f_{\text{sample}} = 1/T_{\text{sample}}$,

¹An OFDM transmit signal defined in Equation (1.3) has in theory an infinite bandwidth and thus the sampling theorem can never be fulfilled completely.

²From now on the index i on the modulation symbols, indicating the OFDM-symbol number, will be ignored.



(a) OFDM modulation by means of IFFT processing.



(b) OFDM demodulation by means of FFT processing.

Figure 1.3: Implementation effective IFFT/FFT processing.

followed by DFT/FFT with size N_{FFT} , as depicted in Figure 1.3(b). It should be noted that the ratio N_{FFT}/K , which could be seen as the over-sampling of the discrete OFDM signal, could very well be, and typically is, a non-integer number.

The OFDM seems to be an easy-to-implement multi-carrier transmission technique with perfectly orthogonal subcarriers. This will remain true only in an AWGN channel. However, in case of a time-dispersive channel the orthogonality between the subcarriers will be at least partly lost. The reason for this loss of orthogonality is that, the demodulator correlation interval for one path will overlap with the symbol boundary of a different path. Thus, the integration interval will not necessarily correspond to an integer number of periods of complex exponentials of that path as the modulation symbols $a[k]$ may differ between consecutive symbol intervals. As a consequence, in case of a time-dispersive channel there will not only be inter-symbol interference within a subcarrier but also interference between subcarriers. To deal with this problem and to make an OFDM signal truly robust to the time dispersion on the radio channel, the so-called cyclic prefix is inserted into an OFDM symbol. The cyclic-prefix insertion implies that the last N_{CP} samples of the IFFT output block of length N_{FFT} are copied and inserted at the beginning of the block. In this way, it increases the duration of the OFDM symbol from T_s to $T_s + T_{\text{CP}}$, where T_{CP} is the duration of the cyclic prefix, with a corresponding reduction in the OFDM symbol rate (accordingly, also the block length is increased from N_{FFT} to $N_{\text{FFT}} + N_{\text{CP}}$). At the receiver side, the corresponding samples are discarded before the OFDM demodulation. If the correlation at the receiver side is still only carried out over a time interval T_s , subcarrier orthogonality will then be preserved also in case of a time-dispersive channel, as long as the span of the time dispersion is shorter than the cyclic-prefix length.

1.2 Modularity of OFDM Systems

One great advantage of OFDM systems is that once the channel state is known, the transmitter can decide on the correct transmission parameters for each subcarrier, namely, coding rate and modulation order. Furthermore, also different power levels per subcarrier can be assigned according to the channel quality. In this way, the fact that the channel shows strong variations is not just accepted, but even exploited. On the subcarriers with good SNR, it can be transmitted either with higher data rate or with less power than on those with lower SNR, while still guaranteeing a certain reliability of

the transmission. In this diploma thesis, I concentrate on the algorithms that enable to allocate the power and number of bits per subcarrier adaptively.

But OFDM is not only a modulation format that allows to transmit with higher data rate for a single user. Moreover, it can be also used as a multi-access method by assigning different subcarriers to different users. Particularly, each user can get several subcarriers that are not adjacent in the frequency domain, and thus provide considerable frequency diversity. However, in this strategy several problems arise [3]:

- the administrative effort of such an assignment is very large, an appropriate granularity in the frequency domain has to be decided;
- each of the users "sees" different channels, and thus the orthogonality between the signals from different users is destroyed;
- in the uplink, signals from different users do not arrive at the receiver in a synchronized way because of the different run-times; therefore it is more common to combine OFDM with TDMA or alternatively with FDMA.

1.3 The WiMAX standard

The WiMAX link level simulator (the current version in Matlab, an equivalent implementation in Simulink can be found in [4]), into which I implemented the channel adaptive algorithms, is based on the IEEE 802.16-2004 specification working in the frequency band 2-11 GHz [5]. For this specification, primarily intended for the stationary transmission, the WiMAX standard defines the air interface [6] that includes the definition of the medium access control (MAC) and the physical (PHY) layers. For my purposes is the most important the latter one. I concentrate on the OFDM-based PHY layer that is more suitable for non line-of-sight (NLoS) than for line-of-sight (LoS) transmissions. Particularly, the simulator implements the profiles using the 256-point FFT OFDM PHY layer specification. Furthermore, fixed WiMAX systems provide up to 5 km of service area allowing transmissions with a maximum data rate up to 70 Mbps in a 20 MHz channel bandwidth, and offer the users a broadband connectivity.

To make a most efficient use of the bandwidth, the modulation method is adjusted almost instantaneously for the optimum data transfer. This is done by designing an adaptive modulation and coding mechanism that depends on the channel and interference conditions. For this purpose, there are robust Forward Error Correction (FEC)

techniques that are used to detect and correct errors. The FEC scheme is implemented with a Reed-Solomon encoder concatenated with a convolutional one, both with variable rates according to the AMC mode (in case of the convolutional code done by puncturing), and followed by an interleaver.

Fixed WiMAX supports both time and frequency division duplexing formats, FDD and TDD, allowing the system to be adapted to the regulations in different countries. In the simulator, the FDD mode is implemented, when the uplink and downlink are separated in the frequency domain. It uses flexible channel bandwidths, comprised from 1.25 to 20 MHz, thus providing the necessary flexibility to operate in many different frequency bands with varying channel requirements around the world. Moreover, it is also crucial for cell planning, especially in the licensed spectrum. There is also optional support of both transmit and receive diversity to enhance performance in fading environments through spatial diversity. The transmitter implements the space-time coding (STC) to reduce a fade margin requirement, or it has also possibility to adapt the spatial multiplexing to increase the total throughput. The receiver uses maximum ratio combining (MRC) techniques to improve the availability of the system.

For the simulations, the WiMAX simulator is set to the nominal channel bandwidth of 5 MHz with the frame duration equal to 2.5 ms. For this channel bandwidth, the WiMAX standard defines a sampling rate of $T_{\text{sample}} = 1/5.76 \mu\text{s}$. The 5.76 MHz total signal bandwidth emerging from this sampling rate covers also some of the guard band carriers (with zero energy) that are outside the 5MHz channel bandwidth. From the frame duration and total OFDM symbol time (including the cyclic prefix duration), the number of OFDM data symbols in one frame is calculated and equal to 44. In each OFDM symbol, there are defined 8 pilot subcarriers, 192 data subcarriers and the DC carrier is not used. Moreover, the standard defines also the preamble at the beginning of each frame that consists of 3 training OFDM symbols used for the synchronization and channel estimation. Thus, there are 47 OFDM symbols in one frame.

Furthermore, to appreciate the features of WiMAX standard, it has to be noted that it supports all types of access, including nomadic, portable and mobile one. To meet the requirements of different types of access, two versions of WiMAX have been defined. The first based on IEEE 802.16-2004 and optimized for fixed and nomadic access, was already mentioned. The second one is designed to support portability and mobility, and is based on the IEEE 802.16e amendment to the standard. Table 1.3 shows how WiMAX supports different types of access and their requirements [7].

Table 1.1: *Types of access to a WiMAX network.*

Definition	Devices	Location/ Speed	Handoffs	802.16- -2004	802.16e
Fixed access	Outdoor and indoor CPEs	Single/ Stationary	No	Yes	Yes
Nomadic access	Indoor CPEs, PCMCIA cards	Multiple/ Stationary	No	Yes	Yes
Portability	Laptop PCMCIA, or mini cards	Multiple/ Walking speed	Hard	No	Yes
Simple mobility	Laptop PCMCIA, or mini cards, PDAs or smartphones	Multiple/Low vehicular speed	Hard	No	Yes
Full mobility	Laptop PCMCIA, or mini cards, PDAs or smartphones	Multiple/High vehicular speed	Soft	No	Yes

2 SISO OFDM

In this chapter, I deal with an OFDM system equipped with K data subcarriers (the remaining subcarriers up to N_{FFT} are zero) and one transmit and one receive antenna (Single Input Single Output (SISO) OFDM) signaling over a SISO frequency-selective fading channel. Due to frequency selectivity, different subcarriers experience in general different channel gains. Hence, the total transmit power should be properly allocated to different subcarriers, based on the available Channel State Information (CSI) at the transmitter and/or at the receiver. I develop a suitable algorithm to find these appropriate power allocations in the following.

2.1 System Model

Figure 2.1 depicts the equivalent discrete time baseband model of the system under the following considerations. Let k denote the subcarrier index as before, but the zero (DC) carrier is now not used for the transmission and there are K data subcarriers, so it follows that $k \in \{1, 2, \dots, K\}$. $P[k]$ will stand for the power allocated to the k th subcarrier. Depending on $P[k]$, a signal constellation $\mathcal{A}[k]$ will be selected, consisting of $M[k] = 2^{2m}$, ($m = 0, 1, 2, 3$) symbols for square QAMs. One subcarrier then entails an information symbol $a[k]$ drawn from $\mathcal{A}[k]$ and conveying

$$b[k] = \log_2(M[k]) \quad (2.1)$$

bits of information. This information symbol $a[k]$ will then be power-loaded, assuming that the average power of the signal constellation is normalized to one, and transmitted on the k th subcarrier over the transmit antenna.

The channel is supposed to be invariant during the transmission of one frame, defined in the WiMAX standard according to Section 1.3, but it may vary from frame to frame. Let $\mathbf{h} := [h[0], h[1], \dots, h[N]]^T$ be the baseband equivalent FIR channel during the given

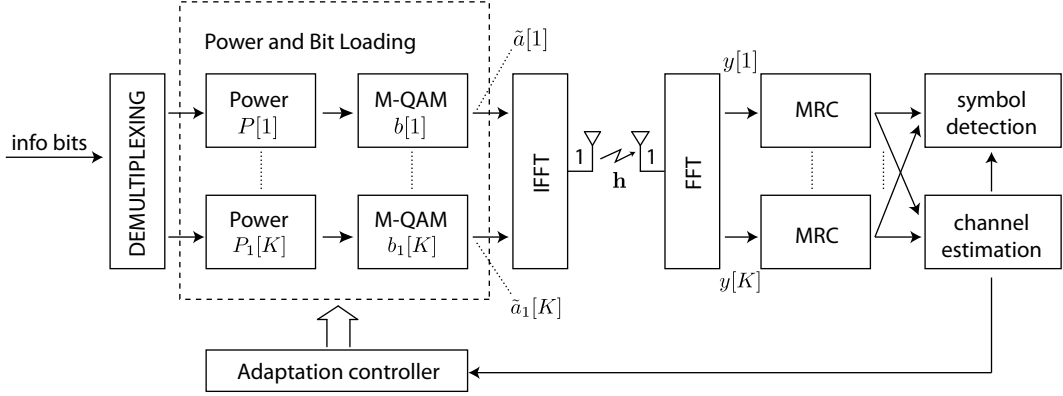


Figure 2.1: Discrete-time equivalent baseband SISO OFDM model.

block with the channel order N . The frequency response of \mathbf{h} on the k th subcarrier is given by the Discrete Fourier Transform

$$H[k] = \sum_{n=0}^N h[n] \exp\left(\frac{-j2\pi kn}{N_{\text{FFT}}}\right). \quad (2.2)$$

By stacking all frequency-domain channel taps for every subcarrier $k \in \{1, \dots, K\}$ into one vector, the frequency-domain channel $\mathbf{H} = [H[1], H[1], \dots, H[K]]^T$ is obtained. Assuming a sufficiently large cyclic prefix, the linear convolution of a time dispersive radio channel will appear as a circular convolution during the demodulator integration interval T_s [8]. The combination of OFDM modulation (IFFT processing), a time-dispersive radio channel, and OFDM demodulation (FFT processing) can then be seen as a frequency-domain channel with input-output relationship per subcarrier given by

$$y[k] = H[k] \sqrt{P[k]} a[k] + n[k], \quad (2.3)$$

where the output of the channel $y[k]$ is the transmitted power-loaded modulation symbol $\tilde{a}[k] = \sqrt{P[k]} a[k]$ scaled and phase rotated by the complex frequency-domain channel tap $H[k]$ and impaired by the complex Additive White Gaussian Noise (AWGN) $n[k]$ having variance $N_0/2$ per real and imaginary dimension. To mitigate the effect of this channel, the receiver should for example multiply $y[k]$ with the complex conjugate of $H[k]$. This is often expressed as a one-tap equalizer being applied to each received subcarrier. Obviously, to apply this simple equalizer, the channel has to be estimated beforehand and signalled back to the transmitter via a signalling channel in case of the Frequency Division Duplex (FDD, defined in WiMAX in Section 1.3). First I assume the

perfect channel knowledge at the transmitter, but I will also explore how to overcome the perfect channel knowledge at the transmitter. Note that to isolate the transmitter design from channel estimation issues at the receiver, the perfect channel knowledge of the channel $H[k]$, $\forall k$ at the receiver has to be assumed [9]. In the following sections, I optimize the system in Figure 2.1 in a frequency-selective channel by maximizing the overall throughput, while still perceiving the average target bit error ratio BER_0 per each subcarrier.

2.2 Constrained Optimization Problem

In this section the constrained optimization problem to maximize the throughput is formulated. The perfect channel knowledge at the transmitter is assumed. The use of the discrete data rate adaptation instead of the continuous one is discussed. In this context, also the general properties of the M -QAM modulations are addressed. This modulation type will be also used in the following sections.

A general approach to the network optimization is based on a utility function. It is used to balance between the efficiency in the network and the fairness among the users. It maps the resources assigned to a user into a real number, which gives us a clear idea about the performance of the given resource allocation. The satisfaction of the users is given by the reliable transmission data throughput T , which is the most important factor, and therefore the utility function $U(T)$ should be a nondecreasing function of the user data throughput T . Obviously, the aim of the network optimization is the maximization of utilities of all users in the network. This can be achieved by adapting the resource allocation, power levels and data rate for each user, as it can be seen in the following subchapters. I concentrate on the single-user scenario that can be extended to the multi-user case by means of cross-layer optimization [10, 11]. I will make some remarks on this extension later on. In the single-user scenario, the utility function $U(T)$ is simply equal to the user throughput T , as the fairness does not play a role in this case. I will show that the user throughput strongly depends on the power allocated across subcarriers, where the transmission takes place. Therefore, this power allocation is one expected solution of the constrained optimization problem.

To obtain the performance bound of the optimization of the OFDM based network, the bandwidth Δf of each orthogonal subcarrier in an OFDM symbol has to be assumed

infinitesimal ($\Delta f \rightarrow 0$) [10]. This is regarded as an extreme case of an OFDM system with an infinitesimal granularity of the resource allocation. However, in practical systems this minimum granularity is just one subcarrier, therefore we have only finite number of subcarriers and our optimization problem turns out to be discrete, not continuous.

My goal in this chapter is to optimize the SISO OFDM system model depicted in Figure 2.1. Here I assume the perfect channel knowledge at the transmitter. The power levels $P[k]$ assigned to subcarriers with index k are grouped into a transmit power vector $\mathbf{p} = [P[1], P[2], \dots, P[K]]^T$, so are the number of bits $b[k]$ into a vector $\mathbf{b} = [b[1], b[2], \dots, b[K]]^T$. Particularly, I want to maximize the user throughput subject to a total transmit power constraint \bar{P} and discrete modulation levels while maintaining the target bit error ratio $\text{BER}_0[k]$ on each subcarrier. These target BERs can be identical or different across subcarriers, depending on the specifications. My objective can thus be formulated as the following constrained optimization problem:

$$\text{maximize } T(\mathbf{p}, \mathbf{b}) = \sum_{k=1}^K b[k] = \|\mathbf{b}\|_1 \quad (2.4)$$

$$\text{subject to } \overline{\text{BER}}[k] \leq \text{BER}_0[k] \quad (2.5)$$

$$\|\mathbf{p}\|_1 = \sum_{k=1}^K P[k] \leq \bar{P} \quad (2.6)$$

$$P[k] \geq 0 \quad (2.7)$$

$$b[k] \in \{0, 1, 2, 3, 4, \dots\} \quad (2.8)$$

It is clear from this formulation that the power and bit loadings $\{P[k], b[k]\}_{k=1}^K$ have to be adapted jointly. The problem is more challenging in the frequency-selective fading scenario compared to the frequency-flat channels, where the constant power transmission is used. When the continuous rate adaptation¹ can be used, the optimal power allocation for an OFDM symbol is a utility-based water-filling [10, 11]. But in practice, continuous rate is infeasible, and there are only discrete modulation levels, as indicated already in Constraint 2.8. Thus, the optimal power level on each subcarrier is not continuous either. Therefore, I have to search for another algorithm than the classical water-filling solution for discrete modulation levels.

¹The throughput is assumed to be continuous, what means that it can be any real number. It follows that the number of bits transmitted within each symbol can be from an infinite set of values.

Note that in Constraint 2.8 both rectangular ($b[k] \in \{1, 3, 5, \dots\}$) and square ($b[k] \in \{2, 4, 6, \dots\}$) QAMs are allowed ($b[k] = 0$ refers to the case of an unused subcarrier). However, I want to reduce the possible constellations just to square ones, what significantly simplifies an adaptive demapper. This excludes also the BPSK modulation, which is used in the first AMC mode of the WiMAX standard. By doing this, one bit at most is lost from the throughput per one OFDM symbol. The proof is given in [9]. Let $d_{\min}[k]$ denote the minimum square Euclidean distance for the given constellation on the k th subcarrier. For QAMs constellations, it holds that [12]

$$d_{\min}^2[k] = 4g(b[k])E_s[k] = 4g(b[k])P[k]T_s, \quad (2.9)$$

where $E_s[k] = P[k]T_s$ is the average energy of the constellation chosen for the k th subcarrier, T_s is the useful symbol duration (OFDM symbol duration without cyclic prefix) and the constant $g(b[k])$ depends on whether the given constellation is square or rectangular QAM:

$$g(b[k]) := \begin{cases} \frac{6}{5 \cdot 2^{b[k]} - 4}, & b[k] = 1, 3, 5, \dots; \\ \frac{6}{4 \cdot 2^{b[k]} - 4}, & b[k] = 2, 4, 6, \dots \end{cases} \quad (2.10)$$

Equation (2.9) simply gives us the decreasing minimum Euclidean distance in the constellation with the increasing modulation order and can be easily verified. From the definition (2.10) of the constant $g(b[k])$ it is clear that the square QAMs are more power efficient than the rectangular QAMs. Thus, with K subcarriers, it is always possible to avoid the usage of less efficient rectangular QAMs and save the remaining power for other subcarriers to use higher order square QAM [9].

2.3 Power and Bit Loading Algorithm for Uncoded Systems

In this section I focus on the adaptive algorithm for uncoded systems. The solution for the constrained optimization problem formulated in Section 2.2 is the so-called *Greedy Power Allocation Algorithm* based on maximizing the total throughput for the discrete rate adaptation [11]. The key idea of this algorithm is to allocate bits and the corresponding power levels successively and to maximize the throughput (number of bits) per power in each step of the bit loading. The influence of this algorithm on an OFDM system is explored at the end of this section.

To derive the solution, an appropriate transmission data rate per subcarrier has to be expressed first depending on the channel condition. This greedy algorithm converts the frequency-selective channel into frequency-flat one, what is the main idea behind the water-filling algorithm [13] and will be explained later. Therefore, the performance of the variable-rate M -QAM modulation in an AWGN channel may be considered per subcarrier. The bit error ratio in this scenario for the k th subcarrier is shown to be well approximated by [14, 15]:

$$\text{BER}_0[k] \approx 0.2 \exp\left(\frac{-1.5\gamma[k]}{M[k] - 1}\right) \quad (2.11)$$

for $0 \leq \gamma[k] \leq 30$ dB, where $\gamma[k]$ is the SNR on the particular subcarrier and $\gamma[k] = P[k] \left(\frac{|H[k]|^2}{N_0}\right)$. Taking the natural logarithm and rearranging Equation (2.11), and assuming the equality and the same target bit error ratio per subcarrier, I get

$$M[k] = 1 + \frac{1.5}{-\ln(5\text{BER}_0)}\gamma[k] = 1 + \beta\gamma[k], \quad (2.12)$$

where $\beta = 1.5/(-\ln(5\text{BER}_0))$ is a constant called the SNR gap for a target bit error ratio. It indicates the difference between the SNR needed to achieve a certain throughput for a system and the theoretical limit. Using Approximation (2.12), the data rate (number of bits) per subcarrier is given by

$$b[k] = \log_2(M[k]) = \log_2(1 + \beta\gamma[k]) \left[\frac{\text{bits/s}}{\text{Hz}} \right] \quad (2.13)$$

Only square QAMs ($b[k] \in \{2, 4, 6, \dots\}$) are allowed without significant loss of optimality as mentioned in Section 2.2. Hence, two bits instead of one are loaded per each stage. By setting $\gamma[k] = P[k] \left(\frac{|H[k]|^2}{N_0}\right)$ in Equation (2.13), the required power to transmit $b[k]$ bits/s/Hz is

$$P[k] = \frac{2^{b[k]} - 1}{\beta\rho[k]} = \frac{-\ln(5\text{BER}_0)N_0}{g(b[k])|H[k]|^2}, \quad (2.14)$$

where $\rho[k] = \frac{|H[k]|^2}{N_0}$, and $g(b[k])$ is given by Equation (2.10). Furthermore, I constrain the maximum number of bits allocated to one subcarrier to 6 (64-QAM). This results in the remaining transmit power at high SNR values ($\sum_{k=1}^K P[k] < \bar{P}$). The power cost incurred when loading the l th and $(l-1)$ th bits to the k th subcarrier is

$$c(k, l) = \frac{-\ln(5\text{BER}_0)}{g(l)\rho[k]} - \frac{-\ln(5\text{BER}_0)}{g(l-2)\rho[k]}, \quad l = 2, 4, 6; \forall k. \quad (2.15)$$

This cost is quantified by the additional power needed to maintain the target bit error ratio performance. For $l = 2$, $g(l - 2)$ is set to ∞ , and thus $c(k, 2) = \frac{-\ln(5\text{BER}_0)}{g(2)\rho[k]}$. In the following algorithm, I will use P_{rem} to record the remaining power after each iteration, $b^{(n)}[k]$ to store the number of bits already loaded to the k th subcarrier, and $P^{(n)}[k]$ to denote the power level in the iteration step n on the k th subcarrier. Now, the *Greedy Power Allocation Algorithm* can be described step-by-step [9]:

1. Initialization step $n = 1$: Set the remaining power equal to the power constraint $P_{\text{rem}} = \bar{P}$. For each subcarrier, set $b^{(n)}[k] = P^{(n)}[k] = 0$.
2. Compute $c(k, b^{(n)}[k] + 2)$ for all subcarriers, where $b^{(n)}[k] \neq 6$. If $b^{(n)}[k] = 6$, then set $c(k, b^{(n)}[k]) = \infty$. Choose the subcarrier that needs the least power to load two additional bits, i.e. select

$$k_0 = \arg \min_k c(k, b^{(n)}[k] + 2). \quad (2.16)$$

3. If there is not enough power remaining, i.e. if $P_{\text{rem}} < c(k, b^{(n)}[k] + 2)$ (it always happens if all subcarriers are loaded with 6 bits ($b^{(n)}[k] = 6, \forall k$)), then exit with $\{P[k] = P^{(n)}[k], b[k] = b^{(n)}[k]\}_{k=1}^K$. Otherwise, load two bits to the subcarrier k_0 , and update iteration variables:

$$P_{\text{rem}} = P_{\text{rem}} - c(k, b^{(n)}[k] + 2) \quad (2.17)$$

$$P^{(n)}[k_0] = P^{(n)}[k_0] + c(k, b^{(n)}[k] + 2) \quad (2.18)$$

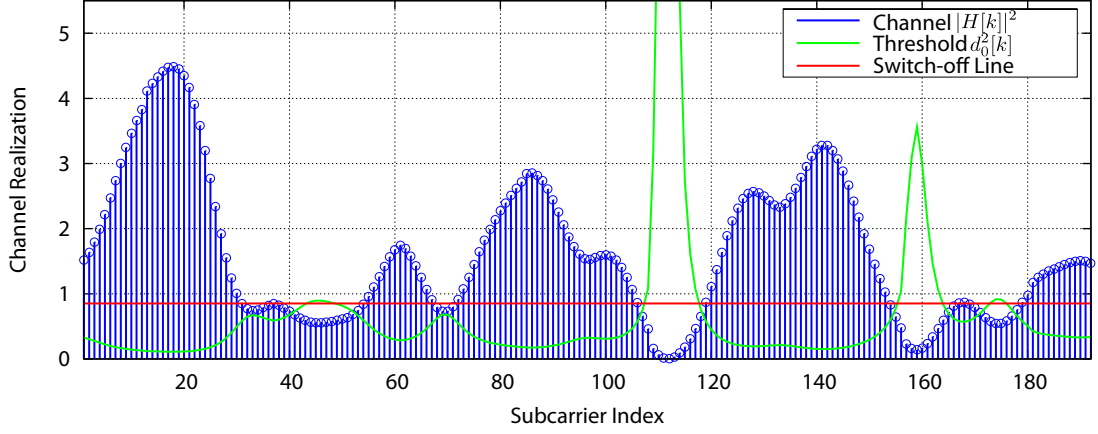
$$b^{(n)}[k_0] = b^{(n)}[k_0] + 2. \quad (2.19)$$

4. Loop back to step 2 with $n = n + 1$.

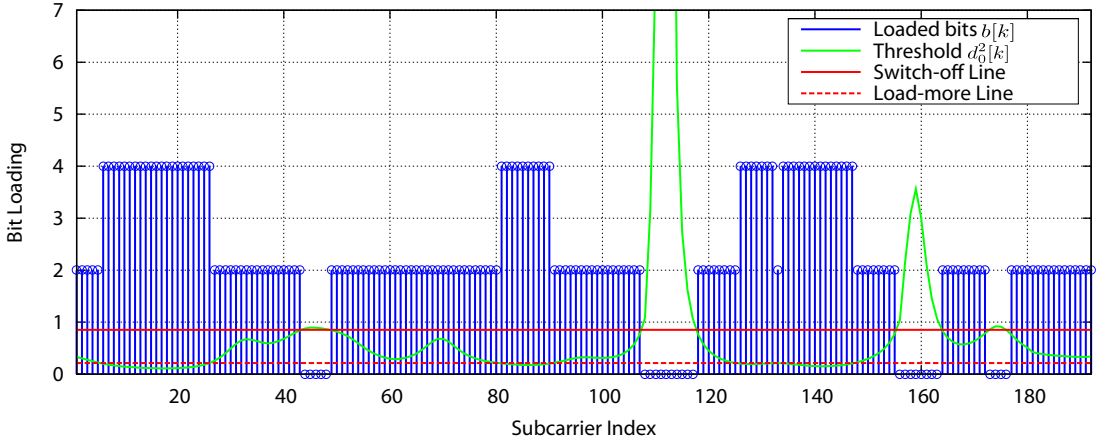
It is possible to show that this greedy algorithm with one bit loaded per iteration results in the global optimal bit loading and power allocation. The proof can be found in [16].

Now, I explore in more detail the effects of the mentioned joint power and bit loading on an OFDM system. The resulting number of bits loaded across subcarriers and the appropriate power levels are depicted in Figure 2.2. From Equation (2.9), the allocated power can be calculated as

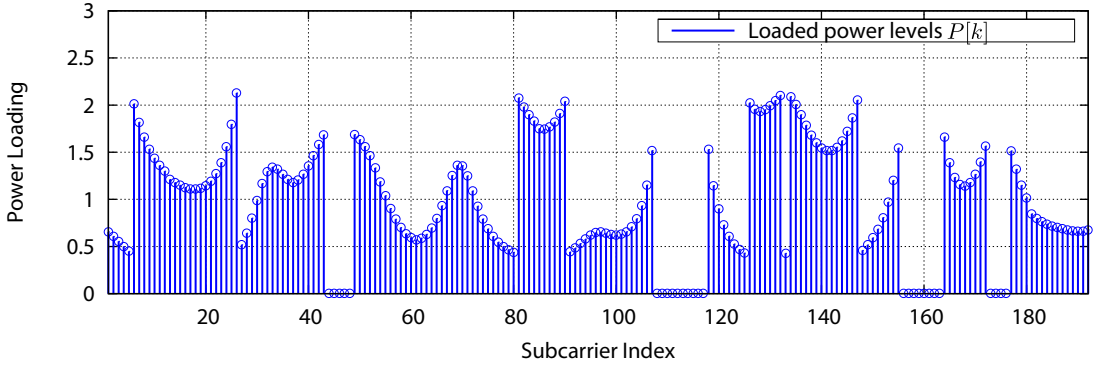
$$P[k] = \frac{d_{\min}^2[k]}{4g(b[k])} = \frac{d_0^2[k]}{g(b[k])}, \quad (2.20)$$



(a) Threshold metric $\{d_0^2[k]\}_{k=1}^K$ and a certain channel realization.

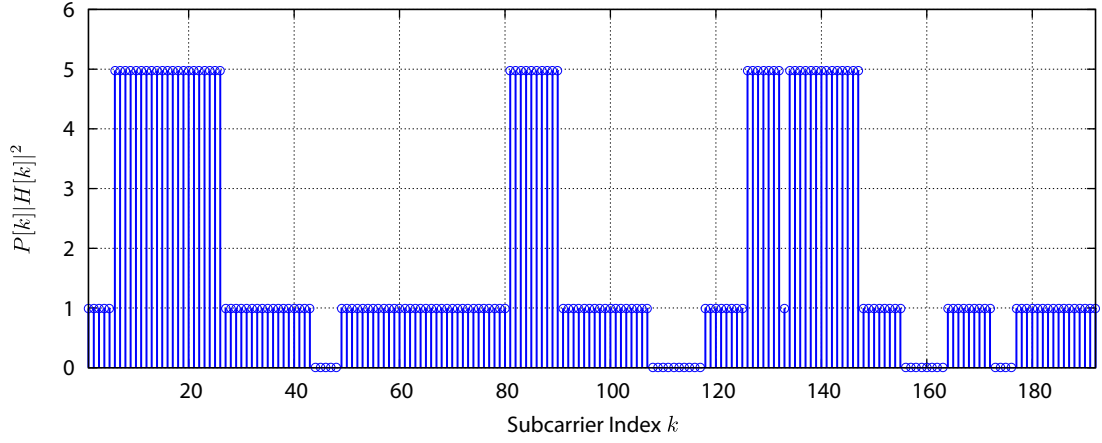


(b) Bit loading snapshot for a certain channel realization.

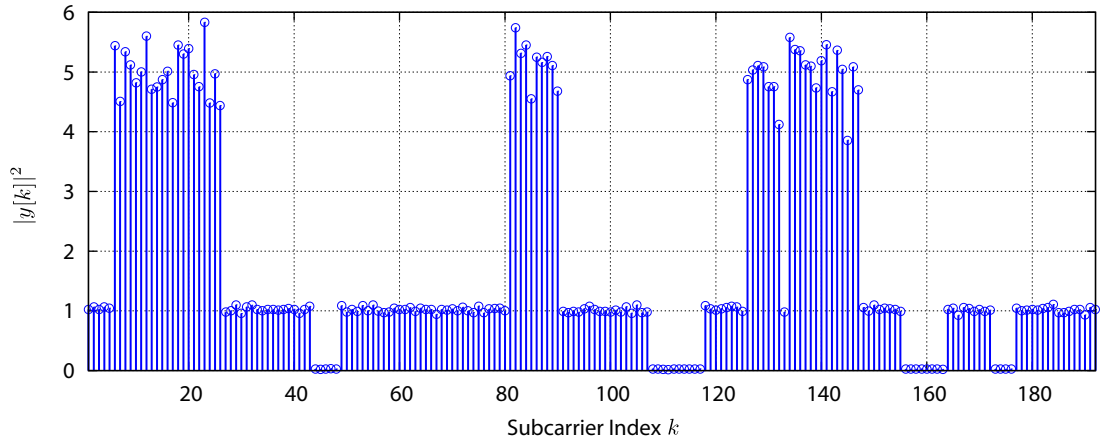


(c) Power loading snapshot for a certain channel realization.

Figure 2.2: Resulting bit and power allocation across K subcarriers for a certain channel realization with SNR=16 dB and target bit error ratio $\text{BER}_0 = 10^{-5}$.



(a) Pre-equalized channel at the transmitter.



(b) Power levels of the received signal.

Figure 2.3: Greedy Power Allocation Algorithm - way to combat frequency-selective fading (a certain channel realization with SNR=16 dB and target bit error ratio $\text{BER}_0 = 10^{-5}$).

where T_s is assumed to be one second and $d_{\min}^2[k]/4$ will be the scaled distance metric $d_0^2[k]$ that summarizes the power and bit loading information per subcarrier. Comparing the latter expression in Equation (2.20) with Equation (2.14), the threshold (distance) metric $d_0^2[k]$ can be expressed as

$$d_0^2[k] = \frac{-\ln(5\text{BER}_0)N_0}{|H[k]|^2}. \quad (2.21)$$

In Figure 2.2(a) a certain realization of the frequency-selective channel is shown with the corresponding threshold metric. It can be seen that the threshold metric is proportional to the inverse of the channel and it is corrected appropriately to meet the required bit error ratio.

It is possible to find the threshold (the red line in Figure 2.2(a), not needed by the algorithm itself, just for illustration), when the channel conditions are too bad and it is not feasible to accommodate the minimum number of bits (2) with the given power constraint at the corresponding subcarriers. This can be also seen in Figure 2.2(b) and 2.2(c), where no power and bits are loaded to these subcarriers (they are switched off). At these subcarriers, there are deep fading dips, which are recognized by the channel transfer $|H[k]|^2$ being close to zero and below the red threshold or by the metric $d_0^2[k] \sim 1/|H[k]|^2$ being far above the red threshold. If there was more transmit power available, the algorithm would decide whether it is more efficient to load additional bits to used subcarriers or to switch on more subcarriers.

In Figure 2.2(b) it can be seen how many bits are loaded to each subcarrier (2 bits - 4-QAM and 4 bits 16-QAM) and another threshold (the red dashed line in Figure 2.2(b)) can be found, again just for illustration, where the channel conditions are so favorable that higher modulation order may be used.

According to Figure 2.2(c), higher average power is assigned to larger signal constellations, as they require more power for the reliable transmission due to their decreased minimum Euclidean distance d_{\min} . In Figure 2.2(c) one may observe how the power levels across the subcarriers are proportional to the threshold metric $d_0^2[k]$. This results into the flat pre-equalized channel, depicted in Figure 2.3(a), and calculated as follows

$$P_{\text{flat}}(b[k]) = P[k]|H[k]|^2 = \frac{-\ln(5\text{BER}_0)N_0}{g(b[k])|H[k]|^2}|H[k]|^2 = \frac{-\ln(5\text{BER}_0)N_0}{g(b[k])} \quad (2.22)$$

Accordingly, almost constant power levels for each signal constellation are obtained at

the receiver (Figure 2.3(b)), with low variations due to the additive noise and different power levels of individual symbols of higher constellation (16-QAM and higher). In the following calculation, the additive noise is omitted so that the input-output relation with the power loading is only $y[k] = H[k]\sqrt{P[k]}a[k]$

$$\begin{aligned} P_{\text{flat}}(b[k]) &= \left| \frac{y[k]}{a[k]} \right|^2 = \left| \frac{H[k]\sqrt{P[k]}a[k]}{a[k]} \right|^2 \\ &= \left| H[k] \sqrt{\frac{-\ln(5\text{BER}_0)N_0}{g(b[k])|H[k]|^2}} \right|^2 = \frac{-\ln(5\text{BER}_0)N_0}{g(b[k])} \end{aligned} \quad (2.23)$$

For the example in Figure 2.3, the actual values are

$$P_{\text{flat}}(2) = \frac{-\ln(5 \cdot 10^{-5})0.0502}{0.5} = 0.9943 \quad (2.24)$$

$$P_{\text{flat}}(4) = \frac{-\ln(5 \cdot 10^{-5})0.0502}{0.1} = 4.9716. \quad (2.25)$$

This is the direct proof how the *Greedy Power Allocation Algorithm* combats the fading and transforms the frequency-selective fading channel into the frequency-flat channel. Therefore, an AWGN channel can be considered per subcarrier with some scaling factor, depending on the channel and the power loading.

2.4 Power and Bit Loading Algorithm for Coded Systems

In order to be able to implement the *Greedy Power Allocation Algorithm* into more realistic systems and to compare the performance with them, some kind of error correction coding has to be assumed. In this section the *Greedy Power Allocation Algorithm* for coded systems is developed. Prior to this, a short discussion about the general properties of the convolutional code is given. Then the bit error ratio approximation of the [133 171]₈ half-rate convolutional code is derived in an AWGN channel. This approximation is expressed for the combination of the convolutional code with the 4-,16- and 64-QAM modulator.

In OFDM systems, the coding can be very well deployed in the frequency and time domain in order to improve the performance in fading channels. If there was an efficient

interleaving across subcarriers, the coded bits would be transmitted over independent channel states, where the fading of bits would be therefore independent. This would automatically result in a high diversity order. The channel states can be created by temporal variations of the channel, or as different transfer functions of subcarriers in a frequency-selective channels. Thus, it is not really necessary to define new codes for OFDM, but rather to better design appropriate mappers and interleavers that assign the different coded bits in the time-frequency domain. This mapping, in turn, depends on the frequency selectivity as well as the time diversity of the channel. If the channel is highly frequency-selective, then it might be sufficient to code only across available frequencies, without any coding or interleaving in the time domain.

As mentioned in Section 1.3, I have a Reed-Solomon encoder concatenated with a convolutional one, both with variable rates according to the AMC mode, and followed by an interleaver. Interleaving is done in the time and frequency domain for one code-word that is created from the whole data part of one frame. This encounters a lot of difficulties, because just the convolutional codes are not easy to study unless all the basic concepts involving them, as well as the systems that surround them, are very well understood. There is a long discussion about expressing the performance bounds in [17]. I have to find an analytic expression for the probability of a bit error for the M -QAM modulation with the convolutional code and soft Viterbi decoder in an AWGN channel. Because this task is not trivial and I want better to concentrate on the channel adaptive algorithms, I cover only the first AMC mode of the WiMAX with the $[133\ 171]_8$ half-rate convolutional code and interleaver. This can be then further extended to remaining AMC modes, when especially the $\beta(d)$ spectrum is known for the punctured convolutional codes. I discuss this issue later in this section.

2.4.1 Bit Error Probability for Convolutional Codes and M -QAM

A convolutional code can be precisely described by some parameters. The main parameters are briefly described below but only in order to review the basic concepts to be used throughout this thesis. A $[n, k, K]$ convolutional code inputs k input bits at a time to the encoder shift register, and gets n bits at the output of the shift register. Consequently, the code rate is defined by the expression

$$R_c = \frac{k}{n}. \quad (2.26)$$

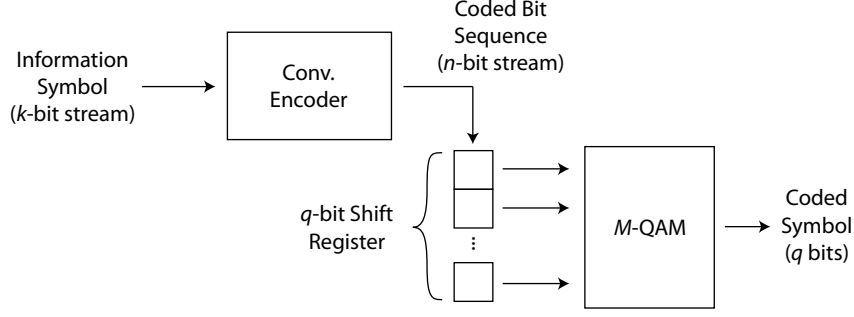


Figure 2.4: Block diagram of a M -QAM modulator with a convolutional encoder.

Notice that n is always greater than k resulting in R_c less than 1. Since R_c is less than 1, the final maximum bit rate is reduced. On the other hand, a small value of the code rate R_c indicates a high degree of redundancy, which should cause decreased error ratio at the expense of increasing the bandwidth of the encoded signal. The convolutional encoder memory consists of a shift register with K k -bit stages and n modulo-2 adders. The parameter K is called the constraint length of the convolution code. Therefore, each one of the n output bits is affected by K sets of k input bits. [18]

A compact way to represent graphically all possible convolutional code outputs as a function of their inputs is by means of a trellis diagram. Each branch in the trellis expresses not only the n -bit output associated with the k -bit input, but also graphically indicates the correspondent encoder's internal state change in each node. The set of successive branches defines a sequence of outputs also known as a path of code words. Figure 2.4 shows the n -bit output of the convolutional encoder feeding an M -QAM modulator that accepts q bits per time. Considering $n > q$, those n -bits in N sets of q bits can be grouped, where $q = \log_2 M$ and N is essentially an integer number. On the other hand, if $n \leq q$, N sets of n bits will be necessary to make a group of q bits. In both cases, the q bits are the ones that travel simultaneously through the channel in the form of a symbol, which is called a coded symbol. Therefore, each branch in the trellis can either represent N coded symbols ($n > q$) or only one coded symbol ($n \leq q$), depending on the scenario. Then any path in the trellis is composed of various branches and thus involves many coded symbols.

The distance d between the two paths in the trellis is the number of coded symbols in which these two paths differ [19]. I denote the probability of choosing a wrong path in the trellis in a pairwise comparison of the all-zero path (correct path) with another path (wrong path) separated by the distance d from the all-zero path as a pairwise error

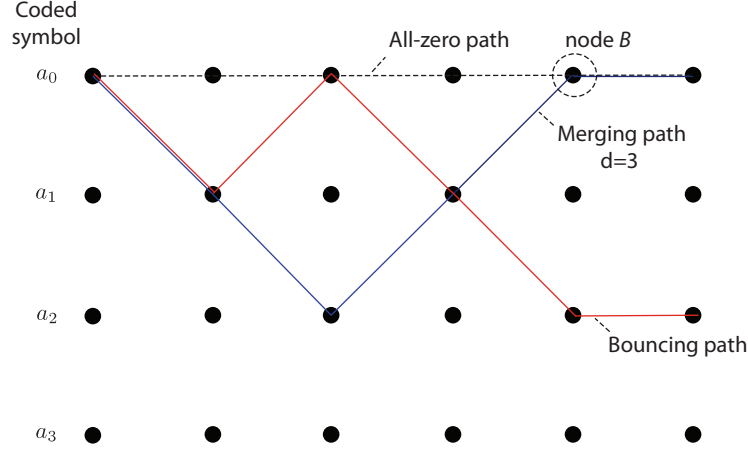


Figure 2.5: Example of a trellis.

probability $P_2(d)$.

A convolutional encoder usually starts with the reset internal shift register, i.e., in the all-zero state. In this state, its response to an all-zero input will always be an all-zero output. In other words, the encoder will remain in the all-zero state forever until a non-zero input occurs. The all-zero path is the horizontal line connecting all zero-state nodes in the trellis diagram, shown in Figure 2.5. It expresses the correct decision that a decoder should make when the information sequence with only zero symbols was transmitted. This can be used very efficiently to determine the performance of the convolutional code. The method for this is based on sending the all-zero information symbol sequence constantly. If a channel error occurs, the behavior of the decoder algorithm can be estimated by studying the paths that could leave and again merge at the all-zero path (an example of such a path can be seen in Figure 2.5). A path (the red path in Figure 2.5) that leaves and bounces the all-zero path instead of merging with it should not be considered. The minimum free distance, denoted d_{free} , is the minimum number of different coded symbols (q -bit long) for two paths among all pairs of paths that start and stop at the zero-state node. The larger d_{free} , the greater is the probability of correcting eventual channel errors. Usually the increase of d_{free} is accompanied by an increase in the constraint length for the same code rate, which will make the encoder more complex. One of the goals to find a good convolutional code is to obtain the largest d_{free} for a fixed constraint length and code rate.

Now the probability of a bit error P_b may be explored. The equations for determining P_b versus SNR vary accordingly to the convolutional code, interference, noise, demod-

ulator and implemented decoder algorithm. For convolutional codes, the optimum decoding method involves a search through the trellis for the most probable sequence, what is achieved by means of the Viterbi algorithm with soft-decision decoding. Details can be found in [18, 20].

Since the convolutional code does not necessarily have a fixed length, its performance is derived from the first-event error probability for sequences that merge with the all-zero sequence for the first time at a given node B in the trellis (an example is given in Figure 2.5). Remember that even though the convolutional encoder generates n bits at the output, the considered coded symbols are q bits long (Figure 2.4). Those q bits are really transmitted over the channel and suffer from errors. It is clear that when an incorrect path is selected, the information symbols in which the selected path differs from the correct path will be decoded incorrectly. If the pairwise error probability $P_2(d)$ is multiplied by the number of incorrectly decoded information symbols for the incorrect path at the node where they merge, the information symbol error ratio is obtained for that path. The average symbol error probability P_s is upper-bounded by multiplying each pairwise error probability $P_2(d)$ by the corresponding number of incorrectly decoded information symbols, for each possible incorrect path that merges with the correct path at the B th node, and summing over all d . The appropriate multiplication factors corresponding to the number of wrong information symbols (k -bit long) for each incorrectly selected path that merges with the all-zero path at the B th node is given by the distance spectrum $\beta(d)$. It can be obtained for every distance d between paths for example by differentiating the transfer function. Finally, we can write a general formula for determining an upper bound on P_s for the M -QAM with a convolutional code

$$P_s \leq \sum_{d=d_{\text{free}}}^{\infty} \beta(d) P_2(d), \quad (2.27)$$

where $\beta(d)$ is the distance spectrum, and $P_2(d)$ is the pairwise error probability. Obviously, the resulting upper bound on the probability of a bit error P_b has to be divided by the number of input information bits k . Thus

$$P_b \leq \frac{1}{k} \sum_{d=d_{\text{free}}}^{\infty} \beta(d) P_2(d). \quad (2.28)$$

This is a well known result that is very often used in practise (for example in [21]). There are two reasons why Equations (2.27) and (2.28) are upper bounds. One is that the events that result in the error probabilities $P_2(d)$ are not disjoint. This can be seen

from the observation of the trellis. Second, by summing over all possible $d \geq d_{\text{free}}$, it is implicitly assumed that the convolutional code has an infinite length. If the code is truncated periodically after T nodes, the upper bounds in Equations (2.27) and (2.28) can be improved by summing the error events for $d_{\text{free}} \leq d \leq T$. The effect on the performance and the appropriate values for truncation lengths T for several codes are studied in [22]. For example for the $[133\ 171]_8$ half-rate convolutional code, the truncation length $T = 36$ is derived, but it is anyway set to 50 in the WiMAX simulator.

Notice that by the definition of Equation (2.28), $\beta(d)$ depends only on the convolutional code employed. It relates the number of information symbols in error with the number of coded symbols detected in error. Since $\beta(d)$ is created by a convolution operation, its expression is mathematically feasible either by employing an analytical method or a numerical method. I will discuss the latter one.

On the other hand, $P_2(d)$ is completely independent from the convolutional code, being a result solely of the type of the noise in the channel, the demodulation and the decoder algorithm. Its calculation is complex and usually done by upper bounding considerations over a pattern of an all-zero sequence transmission as explained earlier. The independence between $\beta(d)$ and $P_2(d)$ makes it possible to split the problem of determining P_b into two completely isolated phases that are merged together in Equation (2.28). The first phase consists of the analysis of the convolutional code in order to determine $\beta(d)$, whereas the second issue concerns the channel model as well as the reception process in order to estimate a reliable upper bound on $P_2(d)$.

Determining the parameter $\beta(d)$

By observing the abrupt tapering form of $P_2(d)$ as d increases, and that $\beta(d)$ will be employed in Equation (2.28), it can usually be seen that only the first five terms of $\beta(d)$ are significant for determining P_b . Therefore, a numerical method [17] can be applied to the convolutional code in order to precisely compute just the first few terms of $\beta(d)$. This method uses the code generator polynomials for generating a large number of paths in the trellis beginning and ending at the all-zero state. These paths contain a relatively large number of branches, usually more than three times the constraint length K , to cover a sufficient range of distances d . The process of the path generation must be in an ascending order of d . The computation of the number of coded symbols different than the all-zero symbol sequence contained in each of those paths gives

d . The number of information symbols different than the all-zero information symbol sequence related to the considered path provides the contribution for $\beta(d)$ made by this specific path. If all those contributions for $\beta(d)$ are continuously summed in a multi-dimensional vector, where each coordinate represents one d , after computing a large number of paths, the vector coordinates will express the $\beta(d)$ spectrum. The accuracy of the first terms of $\beta(d)$ increases as the number of generated paths becomes larger. Furthermore, those terms will freeze as soon as all paths that could contribute to their correspondent spectra $\beta(d)$ have been generated. The process should stop when all terms of interest, usually the first three or five, are obtained. The minimum free distance d_{free} of the code is also achievable at the end of the process. The $\beta(d)$ spectrum (computed vector) will have zeros at the first positions, indicating that there are no paths with $d = 1, 2, 3, \dots, d_{\text{free}} - 1$. The first non-zero term in the $\beta(d)$ spectrum represents therefore d_{free} . More details can be found in [17].

Remember that $\beta(d)$ is a function of the convolutional code employed. However, the same convolutional code can generate different $\beta(d)$ when attached to different M -QAM modulators. Notice that the distance d is related to the number of coded symbols different than the all-zero symbol sequence and the length of the coded symbol varies with M according to $q = \log_2 M$.

Consider a certain system of a convolutional encoder generating one fixed coded bit sequence as a response to a fixed information sequence at its input. If the option of connecting this system to one of the two M -QAM modulators presenting different M 's was available, the same coded bit sequence would be clustered in a different amount of q bits in order to generate the coded symbols for each modulator considered. Consequently, different distances d would result, which would impact $\beta(d)$ directly, even though the information bit sequence and the convolutional encoder remain the same for the two systems. So if I wanted to employ a modulator with higher M and to estimate the performance impact that resulted if the same convolutional code was used, I would have to implement the method for determining $\beta(d)$ shortly described above and developed in [17].

I simply overtake the results derived in [17] for the industry-standard convolutional code used by the WiMAX simulator. The code has the following characteristics: the transfer octal generator $G = [133\ 171]_8$, $R_c = 1/2$ and $K = 7$. Table 2.1 shows the spectra $\beta(d)$ obtained for this code when used with three different M -QAM modulators (the same that are being used in the power and bit loading algorithm). The parameter

Table 2.1: The Spectra $\beta(d)$ for $[133\ 171]_8$ convolutional code when applied to different M -QAM modulators.

Modulation	d_{free}	$\beta(d)$				
		$i = 0$	$i = 1$	$i = 2$	$i = 3$	$i = 4$
4-QAM	6	1	10	38	92	346
16-QAM	4	8	44	323	2033	11575
64-QAM	3	12	140	1784	19873	207985

i denotes the increment that should be added to d_{free} to obtain d , i.e., $d = d_{\text{free}} + i$. Note that in this case the code rate is $1/2$. Thus, the convolutional encoder can only be applied to M -QAM modulators presenting q equal to 1 or q as a multiple of 2, i.e., to 2-QAM ($q = 1$), 4-QAM ($q = 2$), 16-QAM ($q = 4$) or 64-QAM ($q = 6$), etc. Analogously, if the code rate was $1/3$, allowed q would be multiple of 3 (2-QAM ($q = 1$), 8-QAM ($q = 3$), 64-QAM ($q = 6$), 512-QAM ($q = 9$), etc.

Determining the parameter $P_2(d)$ for M -QAM

Remember that the type of noise in a channel affects only the pairwise error probability $P_2(d)$. It is the probability of choosing a wrong path in the trellis in a pairwise comparison of the all-zero path with another path that differs in d symbols from the all-zero path. In [17], this probability $P_2(d)$ was explored in an AWGN channel (with noise variance $\sigma^2 = N_0/2$ per real and imaginary part) and finally upper-bound by

$$P_2(d) \leq Q \left(\sqrt{\frac{d_{\min}^2}{2N_0}} d \right), \quad (2.29)$$

where the equality can be achieved only if all distances in the signal constellations are equal to d_{\min} . Inequality (2.29) may be considered as a general upper bound that may be applied to every M -ary coherent demodulator followed by a soft-decision Viterbi decoder. From Equation (2.9), it is given that $d_{\min}^2 = 4g(b)E_s$. Using Equation (2.10) for the square M -QAM constellations, the square of the minimum distance as $d_{\min}^2 = \frac{1.5}{(M-1)}4E_s$ can be calculated, where $M = 2^b$. Substituting this result into Equation (2.29), the following expression is obtained

$$P_2(d) \leq Q \left(\sqrt{\frac{1.5 \cdot 2E_s}{(M-1)N_0}} d \right). \quad (2.30)$$

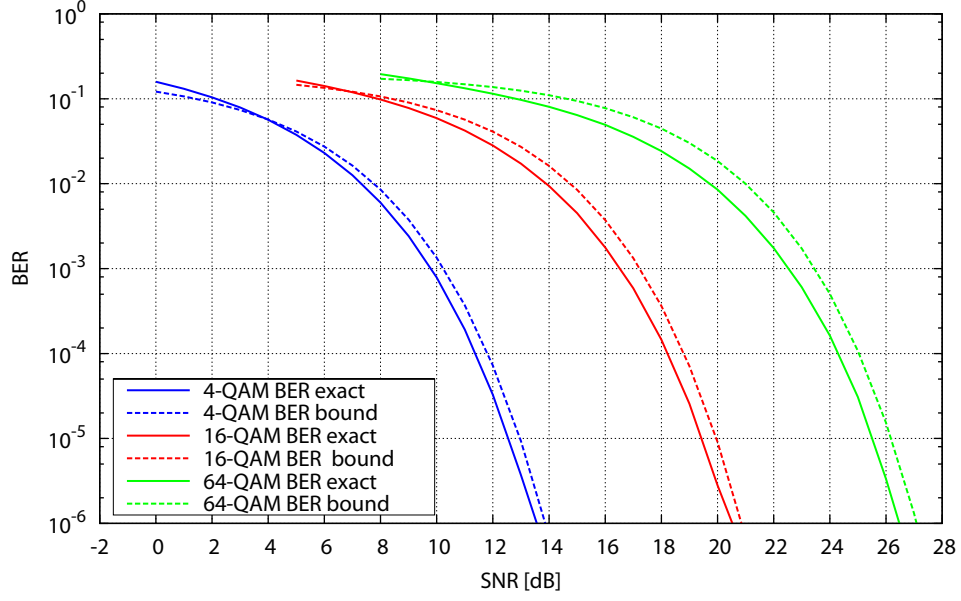


Figure 2.6: Uncoded BER approximation for M -QAM constellations.

Applying the approximation for the Q -function given by $Q(x) \approx 0.2 \exp(-x^2)/2$, from Equation (2.30) it follows

$$P_2(d) \lesssim 0.2 \exp\left(\frac{-1.5\gamma}{M-1}d\right). \quad (2.31)$$

Note that for $d = 1$ (two paths differ only in one symbol), the expression from Equation (2.11) with $\gamma = E_s/N_0$ is reobtained (of course by omitting the notation for a subcarrier k). This lower bound on $P_2(d)$ is actually for $d = 1$ a very good approximation of the bit error ratio (BER) for the M -QAM constellations in an AWGN channel by assuming that one information symbol carries only $k = 1$ information bit (what is always true for $R_c = 1/n$), as also stated in Equation (2.11) and plotted in Figure 2.6. The approximation of the bit error ratio is within 1.5 dB for all constellations at $\text{BER} \leq 10^{-3}$, as confirmed by Figure 2.6.

Calculating and Approximating the Bit Error Probability P_b for M -QAM

In this subsection, the probability of a bit error P_b is finally evaluated for the coded M -QAM modulation in an AWGN channel with the variance $\sigma^2 = N_0/2$ per real and

imaginary part with the soft-decision Viterbi decoder according to the following expression

$$P_b \leq \frac{1}{k} \sum_{d=d_{\text{free}}}^{d_{\text{free}}+4} \beta(d) Q \left(\sqrt{\frac{d_{\text{min}}^2}{2N_0}} d \right), \quad (2.32)$$

where Table 2.4.1 was used, and Equation (2.29) was inserted into Equation (2.28). For practical purposes, the approximation of the Q -function - $Q(x) \approx 0.2 \exp(-x^2)/2$ can be used in Equation (2.32)

$$P_b \lesssim \frac{1}{k} \sum_{d=d_{\text{free}}}^{d_{\text{free}}+4} \beta(d) 0.2 \exp \left(-\frac{d_{\text{min}}^2}{4N_0} d \right). \quad (2.33)$$

Obviously, Equations (2.32) and (2.33) are the worst-case upper bounds, when not all the symbol distances in the signal constellations are equal to d_{min} . Therefore, in many cases, they may be too loose, and thus they can still be improved. Of course, it is possible to influence only the term belonging to $P_2(d)$. For this purpose, the wrong symbol sequence $a^{(l)}$, for $l = 1, 2, \dots, d$ and the correct symbol sequence $a^{(0)}$ are defined. According to [17], consider the definition of an average distance d_{avg} , dependent on the specific combination of all d symbols in the wrong path, given by

$$d_{\text{avg}}^2 = \frac{1}{d} \sum_{l=1}^d \|a^{(l)} - a^{(0)}\|^2. \quad (2.34)$$

Notice that in the worst-case the upper bound $d_{\text{avg}}^2 = d_{\text{min}}^2$ holds, otherwise, $d_{\text{avg}}^2 > d_{\text{min}}^2$. This idea of making a different choice for the worst-case sequence impacts directly Equation (2.29), where d_{min}^2 will be replaced by d_{avg}^2 , and thus also the expression for the bit error probability in Equations (2.32) and (2.33). Even if d_{avg}^2 is just a bit greater than d_{min}^2 , due to the Q -function, the $P_2(d)$ will be reduced considerably. Recall that d is at least equal to d_{free} , which is usually greater than three for good convolutional codes. The main goal to improve $P_2(d)$ would be to determine an acceptable d_{avg}^2 from Equation (2.34) for $d = d_{\text{free}}$ that can replace d_{min} in Equation (2.29). Applying Equation (2.34) for all possible combinations of $d = d_{\text{free}}$ symbols, it can be seen that d_{avg}^2 lies in a finite and discrete domain. Therefore, the best way to study the possible choices of d_{avg}^2 is to plot its discrete probability density function for a certain d_{free} . However, the discrete probability density function will be very dependent on the symbol in the constellation chosen to be $a^{(0)}$. This is studied in a great detail in [17]. The choices of $a^{(0)}$ that provide larger values for $\|a^{(l)} - a^{(0)}\|^2$, will lead to greater d_{avg}^2 , and, consequently

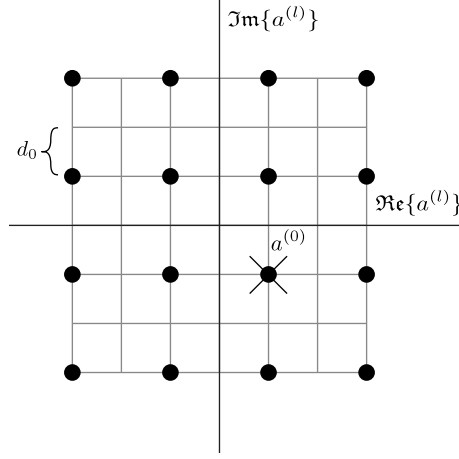


Figure 2.7: The 16-QAM signal constellation with the choice of $a^{(0)}$.

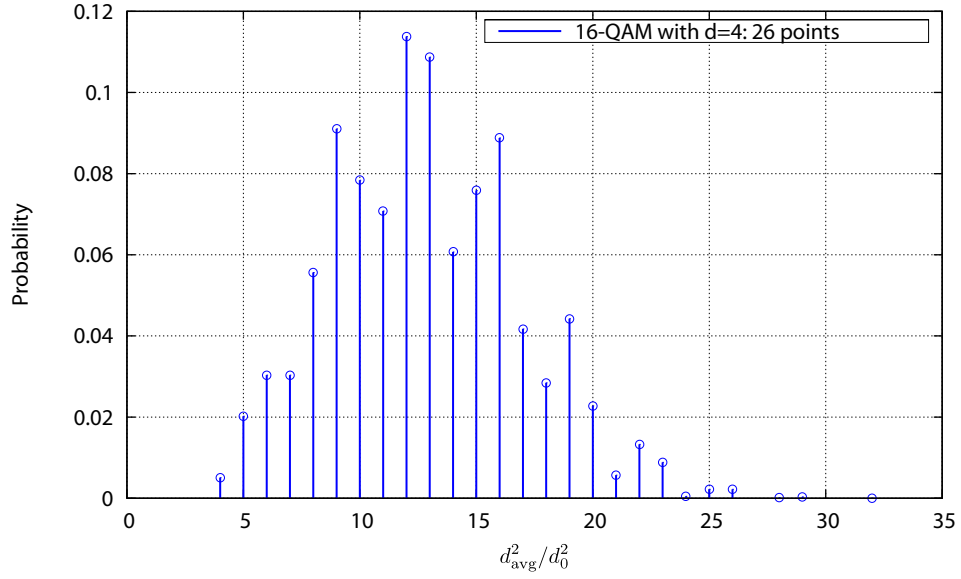
will achieve smaller $P_2(d)$. I will concentrate on a conservative choice of $a^{(0)}$ resulting in the smallest d_{avg}^2 and the largest $P_2(d)$, i.e. $a^{(0)}$ will be one of the most internal symbols in a square constellation, having the largest number of neighbors with the Euclidean distance equal to d_{min} .

Let us have a look at a practical example. The 16-QAM constellation is analyzed in association with the $[133\ 171]_8$ half-rate convolutional code. The standard Gray bit-to-symbol mapping is assumed. Actual bit allocation per symbol is irrelevant in this study. According to Table 2.4.1, the d_{free} is equal to four in this case. The discrete probability density function of d_{avg}^2 is computed by applying Equation (2.34) to all possible symbol sequences that differ in $d_{\text{free}} = 4$ symbols from $a^{(0)}$.

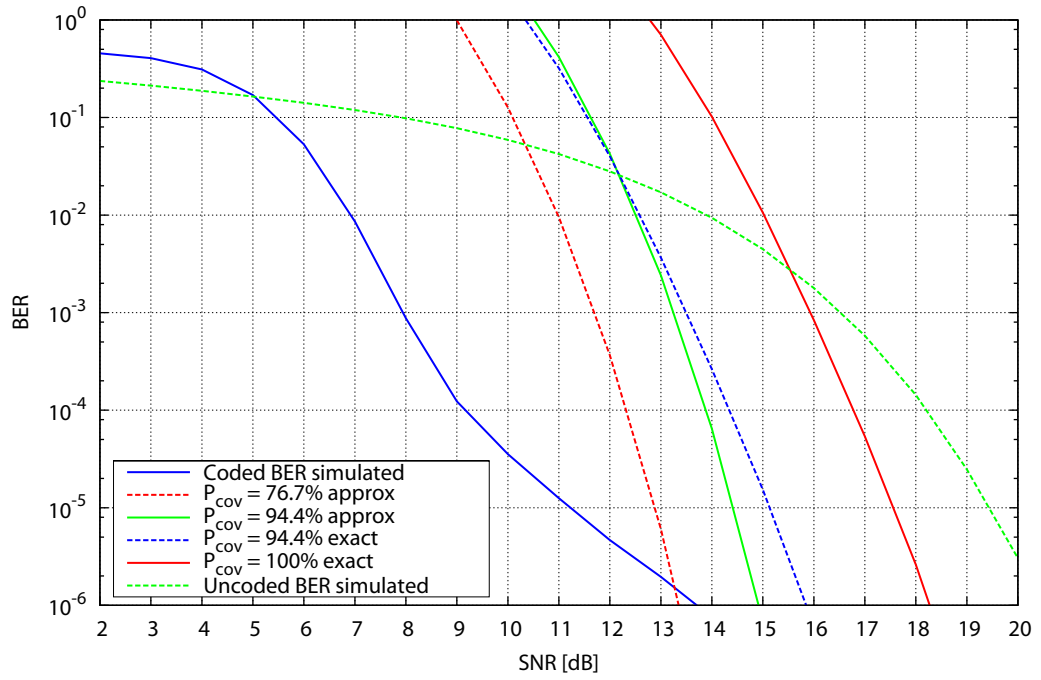
The horizontal axis of Figure 2.8(a) is normalized by d_0^2 , which is called one unit of the grid represented in Figure 2.7. For every square constellation, regardless of M , d_{min} is always equal to a constant $2d_0$. Thus, it holds that $d_{\text{min}}^2 = 4d_0^2$, and from Equation (2.9) d_0^2 can be calculated as follows

$$d_0^2 = g(b)E_s. \quad (2.35)$$

The point where $d_{\text{avg}}^2/d_0^2 = 4$ represents the probability equal to $0.005 = 0.5\%$ of one of the worst-case sequences occurring, when $d_{\text{min}}^2 = d_{\text{avg}}^2$. This value can be also determined analytically by considering the existence of four different closest symbols to $a^{(0)}$ in the constellation (see Figure 2.7). Thus, there are 4^4 possible sequences that will result in $d_{\text{min}}^2 = d_{\text{avg}}^2$. The probability that one of these sequences occurs is given



(a) Discrete probability density function of d_{avg}^2 .



(b) Simulated, upper bounded and approximated bit error ratio curves.

Figure 2.8: Performance analysis of 16-QAM with the half-rate convolutional code with $d = 4$ and conservative choice of $a^{(0)}$.

by

$$P_{d_{\min}^2} = \frac{4^4}{15^4} = 0.00506 = 0.5\%. \quad (2.36)$$

This worst-case upper bound calculated according to Equation (2.32) is plotted in Figure 2.8(b). By choosing for example $d_{\text{avg}}^2 = 5d_0^2$ instead of $d_{\text{avg}}^2 = 4d_0^2$, the final expression for P_b would be valid for at least 99.5% of all transmissions in the channel. For general choices that could be greater than the second smallest, the following formula is provided

$$P_{\text{cov}}(d_{\text{avg},n}^2) = 1 - \sum_{i=1}^{n-1} \Pr \left[\frac{1}{d} \sum_{l=1}^d \|a_l^{(1)} - a^{(0)}\|^2 = d_{\text{avg},i}^2 \right], \quad (2.37)$$

where $P_{\text{cov}}(d_{\text{avg},n}^2)$ represents the percentage of the total d -symbol combinations that the n th smallest possible value of d_{avg}^2 will cover. Note that $d_{\text{avg},1}^2 = d_{\min}^2$.

To answer the question, how far d_{\min}^2 can be decreased, each application intended for the communication link has to be deeply studied, especially the range of SNR values and acceptable bit/packet error rates. Some practical experiments should be done in order to define the acceptable threshold. So it follows that the process of adapting upper bounds has to be performed carefully for every combination of the convolutional code and modulator, and it requires a lot of patience. Therefore, I decided to use just the first AMC mode of WiMAX with the $[133\ 171]_8$ half-rate convolutional code, not to devote so much effort and computational power into this part.

Now the problem is, if the specific d_{avg}^2 has to be determined for every $d > d_{\text{free}}$. Notice that when d increases, the domain of the possible d_{avg}^2 also increases and all probabilities of the previous discrete probability density function will be diluted along this new longer domain. Consequently, by using a certain threshold determined for d_{free} in a path presenting $d > d_{\text{free}}$, the probability of obtaining d_{avg}^2 greater than this threshold is greater than this equivalent probability when using the same threshold for a path in which $d = d_{\text{free}}$. In other words, a threshold that is good for d_{free} will be even better for $d > d_{\text{free}}$.

Coming back to solve the problem what kind of bit error probability approximation should be used in the channel adaptive algorithms, the upper bound given in Equation (2.33) needs to be further simplified. By setting $d = d_{\text{free}} + 4 = \hat{d}$, the part for $P(d)$ is made independent from d . Also the result from the discussion in the last paragraph supports this decision, as the chosen d_{avg}^2 for d_{free} will be even better for $d > d_{\text{free}}$. Then, the expression for d_0^2 , given in Equation (2.21) for an uncoded system, can be derived

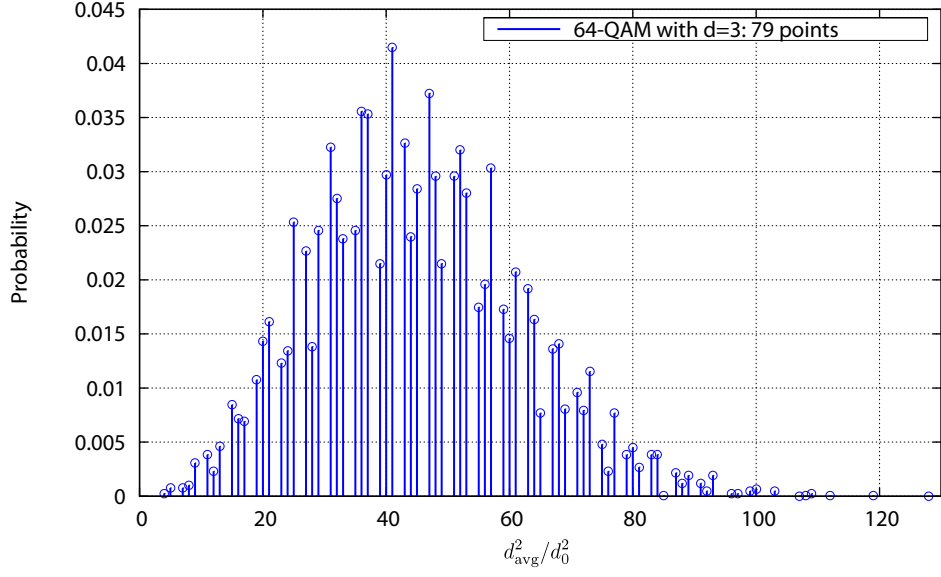
from

$$P_b \approx \frac{1}{k} \sum_{d=d_{\text{free}}}^{d_{\text{free}}+4} \beta(d) 0.2 \exp \left(-\frac{c(b) \cdot d_0^2}{4N_0} \hat{d} \right) \quad (2.38)$$

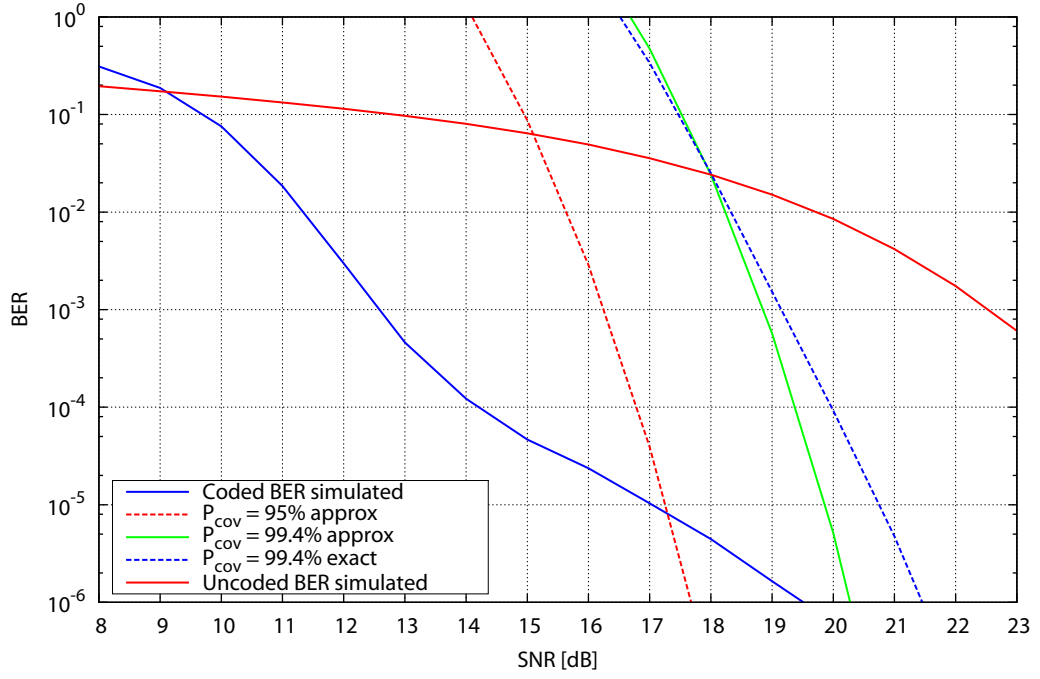
analytically. Compared to Equation (2.21), the metric for a coded system now depends on the used signal constellation.

If $d_{\text{avg}}^2 = 7d_0^2$ was chosen, this assumption would be valid for $100\% - 5.6\% = 94.4\%$ of the transmission according to Equation (2.37). However, the remaining 5.6% would still follow the original worst-case upper bound. But this is still acceptable and due to improvements, it is possible to tighten the original upper bound up to 2 dB (Figure 2.8(b)) and still have a reliable curve that covers 94.4% of all data in the channel. This tightened bound can be approximated by using Equation (2.38), as it is shown in Figure 2.8(b) with the slight performance degradation at high SNR values. For the simulations, the even more pessimistic value of $d_{\text{avg}}^2/d_0^2 = 10$ is used that results only in the 76.7% validity for the transmission (of course without approximation). But this choice will be proved to work well in the channel adaptive algorithms, as it will be seen in Section 2.6. The system is mostly operating in the region of the actual bit error ratio between 10^{-6} and 10^{-5} , where according to Figure 2.8(b) the approximation is still within 1 dB away from the simulated curve.

In Figure 2.9(a), the discrete probability density function for the 64-QAM modulator with the $[133\ 171]_8$ half-rate convolutional code is plotted. In this case, the distance d_{free} is equal to three (see Table 2.4.1). For the simulations, $d_{\text{avg}}^2/d_0^2 = 20$ has been chosen in order to improve $P_2(d)$, when the probability that the updated upper bound covers all data transmission is 95%. To calculate this probability, twelve values on the x -axis in Figure 2.9(a) preceding 20 were inserted into Equation (2.37). The approximated bit error curve is again plotted in Figure 2.9(b). The worst-case upper bound is not even shown in Figure 2.9(b), as it is approximately 4 dB apart from the curve for $P_{\text{cov}} = 99.4\%$. The simple assumption of $d_{\text{avg}}^2 = 11d_0^2$ made the upper bound for P_b tighter in more than 4dB, and thus much closer to the actual bit error probability of the convolutional code. The large shift in the upper bound for 64-QAM when compared with 16-QAM can be explained by analyzing the discrete probability density function of d_{avg}^2 for both constellations shown in Figures 2.8(a) and 2.9(a), respectively. The greater granularity presented by the discrete probability function of 64-QAM points out that d_{avg}^2 for covering 99.4% of all possible sequences is much further away from $4d_0^2$ (d_{avg}^2 for 100%) than it is for 16-QAM. The further the considered d_{avg}^2 is from the minimum value, the larger the shift of the improved upper bound compared to the original one



(a) Discrete probability density function of d_{avg}^2 .



(b) Simulated, upper bounded and approximated bit error ratio curves.

Figure 2.9: Performance analysis of 64-QAM with the half-rate convolutional code with $d = 3$ and conservative choice of $a^{(0)}$.

will be.

Finally, the 4-QAM modulator connected to the same convolutional encoder is considered. Its discrete probability density function of d_{avg}^2 is depicted in Figure 2.10(a). In the 4-QAM constellation, there is no conservative choice of $a^{(0)}$. Also, once $a^{(0)}$ is fixed, among the other three remaining symbols, only one of them represents an Euclidean distance different from the minimal one, which allows the existence of a timid discrete probability density function that can only obtain higher granularity if d becomes too large. For $d = 6$, there are only seven points and the minimum improvement $d_{\text{avg}}^2/d_0^2 = 4.667$ can now cover 91.2% of the data. Notice in Figure 2.10(a) that there is 8.8% of probability that the minimum average distance occurs. The worst-case upper bound, its minimum improvement and its approximation are depicted in Figure 2.10(b). It is interesting that for the simulations the best value for d_{avg}^2/d_0^2 seems to be equal to 8, as proved in the results in Section 2.6. This choice is actually valid only for 0.1% of the transmission. But as can be seen from Figure 2.10(b), the approximated bit error probability curve is still approximately 1 dB apart from the simulated curve. This is sufficient for 4-QAM modulation, which is anyway very robust.

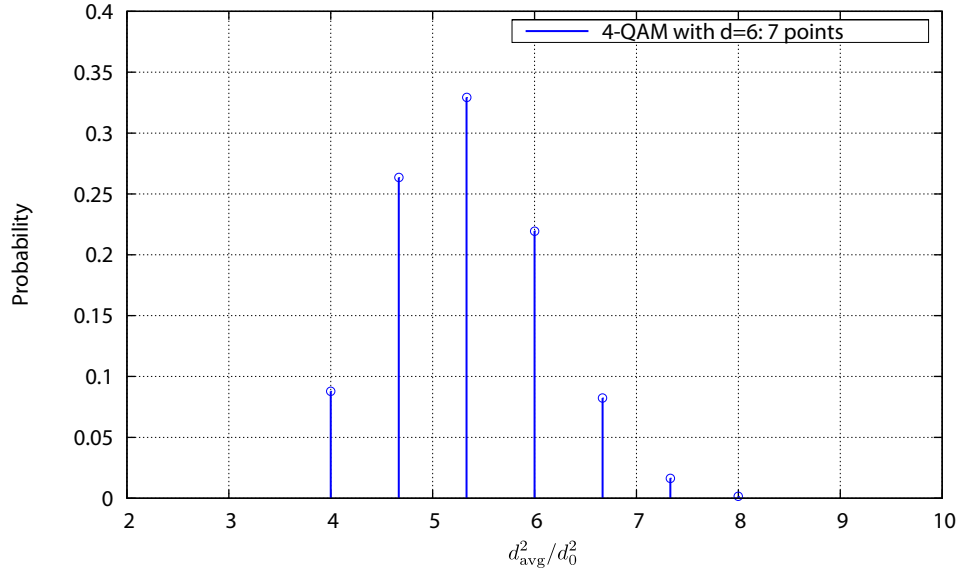
To summarize the latest results, I decided to use for the simulations $d_{\text{avg}}^2/d_0^2 = c(b) = 8, 10, 20$ for 4-, 16-, and 64-QAM, respectively, in order to achieve the bit error ratio in the range of 10^{-6} to 10^{-5} .

Greedy Power Allocation Algorithm for Coded Systems

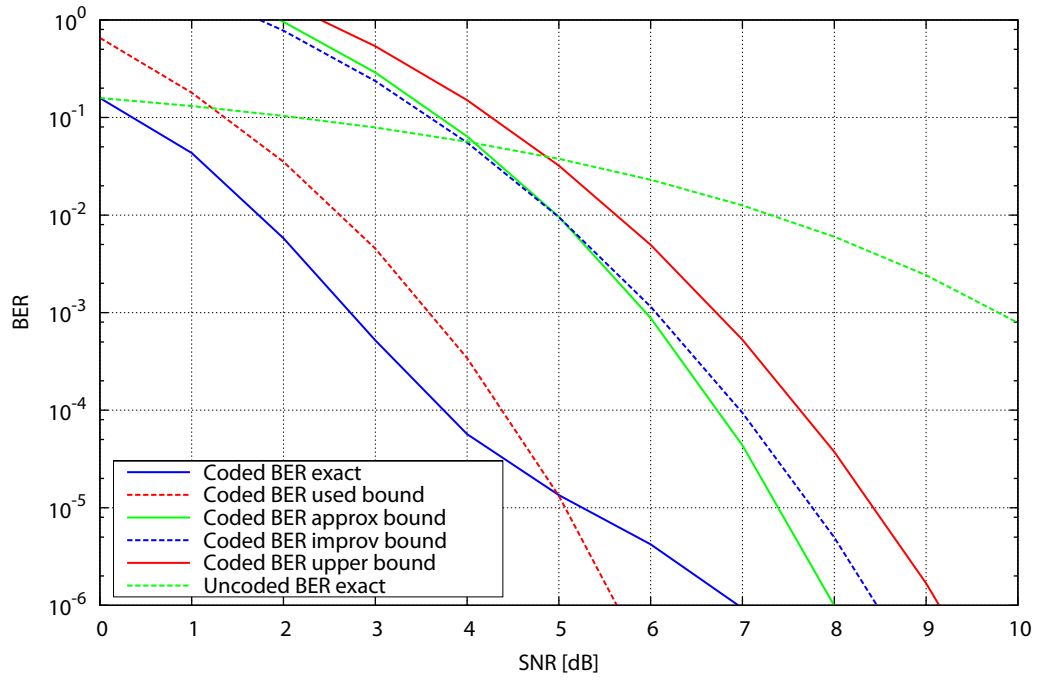
In this part, I return to the notation with k for the k th subcarrier. It should not be confused with the k -bit long information symbol, because with the half-rate convolutional code, the number of information bits k in one information symbol at the input of the convolutional encoder is always equal to one. The threshold metric or the unit grid $d_0^2[k]$ can be expressed from Equation (2.38) as follows

$$d_0^2[k] = \left[\ln \left(\sum_{d=d_{\text{free}}}^{d_{\text{free}}+4} \beta(d)[k] \right) - \ln(5P_b) \right] \frac{4N_0}{c(b[k]) \cdot \hat{d}(b[k]) \cdot |H[k]|^2} \quad (2.39)$$

with the additional term $|H[k]|^2$, because the unit grid $d_0^2[k]$ is scaled by factor $|H[k]|^2$ due to the frequency-flat channel. According to Equation (2.9) replacing $d_{\text{min}}^2[k]$ by term



(a) Discrete probability density function of d_{avg}^2 .



(b) Simulated, upper-bounded and approximated bit error ratio curves.

Figure 2.10: Performance analysis of 4-QAM with the half-rate convolutional code with $d = 6$.

$d_{\text{avg}}^2[k] = c(b[k]) \cdot g(b[k])P[k]$, where T_s (useful symbol duration) is equal to one second, the required power $P[k]$ to transmit $b[k]$ bits/s/Hz is equal to

$$P[k] = \frac{d_{\text{avg}}^2[k]}{c(b[k]) \cdot g(b[k])} = \frac{d_0^2[k]}{g(b[k])}, \quad (2.40)$$

because $d_{\text{avg}}^2[k]/d_0^2[k] = c(b[k])$.

Now the same *Greedy Power Allocation Algorithm* from Section 2.3 can be applied, just the power cost incurred when loading the l th bits to the k th subcarrier has to be modified so that it is adapted to the coding used. Similar to Equation (2.15), the following expression is obtained

$$c(k, l) = \frac{4 \left[\ln \left(\sum_{d=d_{\text{free}}}^{d_{\text{free}}+4} \beta(d, l)[k] \right) - \ln(5\text{BER}_0) \right]}{g(l)c(l)\hat{d}(l)\rho[k]} - \frac{4 \left[\ln \left(\sum_{d=d_{\text{free}}}^{d_{\text{free}}+4} \beta(d, l-2)[k] \right) - \ln(5\text{BER}_0) \right]}{g(l-2)c(l-2)\hat{d}(l-2)\rho[k]}, \quad l = 2, 4, 6; \forall k. \quad (2.41)$$

where $\rho[k] = \frac{|H[k]|^2}{N_0}$, P_b is replaced by the target bit error ratio BER_0 across all subcarriers, $c(0) = \hat{d}(0) = 1$, and $g(0)$ is set to ∞ .

2.4.2 Adaptation of Packet Error Ratio

In this section the *Greedy Power Allocation Algorithm* is further improved to adapt the packet error ratio. The error distribution used in the packet error ratio calculation is discussed. The formula for the BER to PER mapping is given.

As stated in [23], in the most of the error modeling schemes for wireless channels, the models are based on the assumption that data packet transmissions are independently and identically distributed. However, residual errors at the output of the physical layer are not uniformly distributed. This is due to the error correcting mechanism used at the physical layer as well as the correlation induced by the memory existing in fading channels. It is shown in [24] that when the correlation in the error process is not taken into account and the errors are assumed for example to be uniformly distributed, as done in most of the published papers in wireless networking, it leads to the significant overestimation of the packet error ratio (PER), which can go to tenfold factors. However, an analysis of the error event distribution is not easy because of the peculiar

interaction of the particular error correcting codes used in the physical layer of wireless networks. This makes the packet error ratio dependent on the convolutional code and the M -QAM modulator. Moreover, the bit error ratio cannot be easily derived from the packet error ratio. Therefore, I stick to the classical approach where it is supposed that the errors are uniformly distributed in packets with a probability given by BER. With this hypothesis the packet error ratio (PER) is

$$\text{PER} = 1 - (1 - \text{BER}_0)^{\left(\sum_{k=1}^{(K)} b[k] \cdot R_c \cdot N_s\right)}, \quad (2.42)$$

where N_s is the number of OFDM data symbols in one WiMAX frame that represents one codeword. This expression for the packet error ratio leads to overestimation especially in the low SNR region, where the codeword length is shorter due to the *Greedy Power Allocation Algorithm*. But the overestimation is not so crucial especially in an AWGN channel and it will be proved that this expression actually works quite well in the adaptive algorithm.

The *Greedy Power Allocation Algorithm* from Section 2.3 will be now slightly modified, in order to maintain the target packet error ratio PER_0 per each codeword. I still assume to have the same target bit error ratio BER_0 and target packet error ratio PER_0 across all subcarriers. Two iteration loops are given, in the inner n th iteration, it is decided whether to load two more bits or not, in the outer m th iteration it is checked if the target packet error ratio is achieved. In the first iteration step ($m = 1$), I start with the target bit error ratio BER_0 set to 10^{-5} , if the target packet error ratio equal to 10^{-2} is of interest. It is important to have the starting value of BER to obtain the first estimation of the codeword length, because the expression for the coded power cost in Equation (2.41) depends on the target bit error ratio. It follows from the simulation results, plotted in Section 2.6, that for the $\text{PER}_0 = 10^{-2}$ the bit error ratio has to be in the range of 10^{-6} to 10^{-5} (also the bit error ratio approximations were previously optimized for this BER region). Finally, the *Greedy Power Allocation Algorithm* with the packet error ratio adaptation can be described as follows:

1. In the first iteration step $m = 1$, $\text{BER}^{(1)}$ is equal to the pre-set target bit error ratio BER_0 , otherwise

$$\text{BER}^{(m)} = 1 - (1 - \text{PER}_0)^{1/\left(\sum_{k=1}^{(K)} b^{(n)}[k] \cdot R_c \cdot N_s\right)}. \quad (2.43)$$

2. Initialization step $n = 1$: Set the remaining power equal to the power constraint

$P_{\text{rem}} = \bar{P}$. For each subcarrier, set $b^{(n)}[k] = P^{(n)}[k] = 0$.

3. Compute $c(k, b^{(n)}[k] + 2)$ according to Equation (2.41) for all subcarriers, where $b^{(n)}[k] \neq 6$. If $b^{(n)}[k] = 6$, then set $c(k, b^{(n)}[k]) = \infty$. Choose the subcarrier that needs the least power to load two additional bits, i.e. select

$$k_0 = \arg \min_k c(k, b^{(n)}[k] + 2). \quad (2.44)$$

4. If there is not enough power remaining, i.e. if $P_{\text{rem}} < c(k, b^{(n)}[k] + 2)$ (it always happens if all subcarriers are loaded with 6 bits ($b^{(n)}[k] = 6 \forall k$)), then jump to step 5.
5. Otherwise, load two bits to the subcarrier k_0 , and update iteration variables:

$$P_{\text{rem}} = P_{\text{rem}} - c(k, b^{(n)}[k] + 2) \quad (2.45)$$

$$P^{(n)}[k_0] = P^{(n)}[k_0] + c(k, b^{(n)}[k] + 2) \quad (2.46)$$

$$b^{(n)}[k_0] = b^{(n)}[k_0] + 2. \quad (2.47)$$

Loop back to step 3 with $n = n + 1$.

5. Calculate

$$\text{PER}^{(m)} = 1 - (1 - \text{BER}^{(m)})^{\left(\sum_{k=1}^{(K)} b^{(n)}[k] \cdot R_c \cdot N_s\right)}. \quad (2.48)$$

If $\text{PER}^{(m)} \leq \text{PER}_0$, then exit with $\{P[k] = P^{(n)}[k], b[k] = b^{(n)}[k]\}_{k=1}^K$, otherwise loop back to step 1 with $m = m + 1$.

The power cost incurred when loading the l th and the $(l - 1)$ th bits to the k th subcarrier is assumed to be from Equation (2.41). Applying this iterative algorithm, the maximum number of iterations for the packet error ratio adaptation is equal to two ($m = 2$). It may happen that the pre-set target bit error ratio is already sufficient to obtain the target packet error ratio after the first outer iteration. Otherwise, the pre-set target bit error ratio is corrected accordingly to meet the PER requirement after the second outer iteration.

2.5 Quantized Feedback

In this section I discuss the possibilities for the feedback of the channel state information back to the transmitter. The adaptive algorithm developed in previous sections is further modified to implement the quantization of the power levels. Furthermore,

the resulting overhead per subcarrier is evaluated when the Huffman source coding is applied to the quantized power levels and number of loaded bits.

There are three possibilities where to perform the proposed versions of the *Greedy Power Allocation Algorithm* - at the transmitter, at the receiver, or both, where the latter one is not very convenient. The issues of the Time Division Duplex are discussed, where the forward and the reverse channel use the same frequency and antennas for the duplex links, therefore the principle of reciprocity can be applied. More important for this discussion is the Frequency Division Duplex, used by the fixed WiMAX, where the forward link channel is estimated at the receiver and sent to the transmitter on the reverse link. This feedback will involve some delay (or lag), δ_{lag} . Since wireless channels are time-varying, the following condition has to be fulfilled

$$\delta_{\text{lag}} \ll T_c, \quad (2.49)$$

where T_c is the coherence time of the channel.

Three ways are proposed in the literature how the transmitter and the receiver can communicate the decision about the used signal constellation and power level on each subcarrier with each other [3]:

- *Explicit transmission*: If the algorithm is performed at the transmitter, the transmitter can send, in a predefined and robust format, the quantized power level and number of bits loaded to each subcarrier. This message has to be carefully well-protected against errors during the transmission. The drawback of this method is that it is very difficult to provide the transmitter with channel-state information at least close to the perfect one in the case of the FDD mode. Therefore, it is suitable for the TDD mode.
- *Implicit transmission*: It is possible when the transmitter gets its channel state information from the receiver via feedback (in the FDD mode), or in the case of the TDD mode. Then, the receiver knows exactly the basis on which the algorithm at the transmitter is being performed. Thus, the receiver could evaluate the modulation order and power level on each subcarrier according to the same algorithm. The drawback of this solution is the increased computational complexity at both sides, and thus also increased latency. Another big disadvantage is that errors in the channel-state feedback (from receiver to transmitter) not only

lead to a wrong choice of the signal constellation and power level, but also to the wrong detection at the receiver, what could lead to high error rates.

If only the receiver feeds back the parameters that the transmitter should use, the situation is even simpler. Just a very robust feedback channel from receiver to transmitter is needed, similar to the case of the explicit transmission.

- *Blind detection*: From the received signal, the receiver can try to determine the signal constellation. This can be achieved by considering different statistical properties of the received signal, including the peak to average ratio, autocorrelation functions, and higher order statistics of the signal. The advantage of this approach is that no feedback of the transmit parameters is required. Therefore, no channel knowledge is assumed at the transmitter. It follows that this approach is inappropriate for the channel adaptive algorithms.

I propose to use the special case of the implicit transmission when the *Greedy Power Allocation Algorithm* is performed at the receiver that feeds back the decision about transmit parameters back to the transmitter. This way is more feasible in the FDD than TDD mode and well suited for the scenario of the fixed WiMAX, where the connected devices usually dispose of the increased battery life and higher computational power than traditional mobile phones. Another advantage of fixed scenarios is that the channel varies slowly and therefore the delay of the receiver feedback is not crucial, as the channel may stay constant over several frames. Thus, also the condition in Equation (2.49) can be easily fulfilled and possibly used to decrease the resulting overhead in the feedback.

2.5.1 Greedy Power Allocation Algorithm with Quantization

In this modification of the algorithm from Section 2.4.2, the number of allocated bits is preserved per each subcarrier but the corresponding power levels are quantized to four bits. This quantization takes place after the algorithm from Section 2.4.2 that assigns the number of bits per each subcarrier and the corresponding power level. The quantizer used for this purpose is in the floating-point mode rounding to the nearest value. This quantizer can express both positive and negative values, so there is actually one bit more needed for a sign. But for the positive power levels $P[k]$ a sign is not necessary.

2.5.2 Greedy Power Allocation Algorithm with Group Quantization

The algorithm from Section 2.4.2 is again running unchanged at the receiver, but the resulting assignment $\{P[k], b[k]\}_{k=1}^K$ is further quantized. For the group of three subcarriers, the minimum modulation order allocated to them is chosen. The corresponding power level is averaged over three subcarriers and quantized to six bits. For the quantization, the same quantizer from the previous subsection is used. The resulting overhead will be evaluated in the following subsection.

2.5.3 Entropy and Huffman Coding

In this subsection, capital P denotes the random variable that takes on the values equal to power levels $P[k]$ across subcarriers with the probabilities p_j . These power levels are first quantized to 16 bits to obtain the discrete random variable. This random variable is an outcome of a specific source (the *Greedy Power Allocation Algorithm*) that assigns this power levels inversely proportional to the magnitude of the random channel.

A measure of the average uncertainty in the random variable is called the entropy of the source and is defined by

$$H(P) = - \sum_j p_j \log_2(p_j). \quad (2.50)$$

The dimension of entropy is bits/source symbol. It is the number of bits on average required to describe the random variable. The entropy $H(P)$ satisfies [25]

$$0 \leq H(P) \leq \log_2 J, \quad (2.51)$$

where J is the number of outcomes of P . Inequality (2.51) is valid also for a general discrete random variable P . In the case of power loading with the quantization to 16 bits, it can be assumed that all power levels are different across 192 subcarriers. In other words, there are $K = 192$ (in the WiMAX standard) unequal values with the probability $p_j = 1/K = 1/192$, and thus according to Equation (2.50)

$$H(P) = - \sum_{k=1}^{192} \frac{1}{192} \log_2 \left(\frac{1}{192} \right) = \log_2 192 = 7.585 \text{ bits/source symbol}. \quad (2.52)$$

It follows that $H(P) = \log_2 J$ according to Equation (2.51) when all the outcomes are equiprobable.

After quantization to four bits as mentioned in Section 2.5.1, the number of various symbols in the new random variable P_{quant} will be decreased. Let us have a look at a specific power level assignment at $\text{SNR} = 30$ dB. In this case, the entropy calculated according to Equation (2.50) is equal to

$$H(P_{\text{quant}}) = 2.2696 \text{ bits/source symbol}, \quad (2.53)$$

so quite a lot of information was lost compared to the result in Equation (2.52). Without source coding, there are still four bits per source symbol ($P_{\text{quant}}[k]$) needed, what results in $192 * 4 = 768$ bits required to signal the information about power levels back to the transmitter.

A source code is a mapping from the set of possible source outcomes (e.g. P_{quant}) to another set of possible codewords C . The expected length of the code is defined as

$$L = \sum_j p_j l_j, \quad (2.54)$$

where l_j is the length of a specific codeword. Since an optimal source code can minimize this expected length, the following theorem is important [26].

Theorem 2.5.1. *Let $l_1^*, l_2^*, \dots, l_m^*$ be optimal codeword lengths for a source distribution and a D -ary alphabet, and let L^* be the associated expected length of an optimal code ($L^* = \sum_j p_j l_j^*$). Then*

$$H_D(P) \leq L^* < H_D(P) + 1. \quad (2.55)$$

An optimal (shortest expected length) prefix code for a given distribution can be constructed by a simple algorithm discovered by Huffman. This algorithm produces a variable-length code called Huffman code for either block or variable length sequences of source outputs. The procedure works best when the different source sequences have dramatically varying probabilities. Some examples and algorithm description can be found in [25, 26].

Applying Huffman code on the quantized random variable P_{quant} results into 449 bits needed to signal the information about power levels back to transmitter, what is $449/192 = 2.3385 = L^*$ (Equation (2.54)) bits per subcarrier compared to 4 bits without using the source coding. Using Equation (2.53), the Inequality (2.55) for the expected length

L^* is fulfilled. Averaging the number of signaling bits over 1000 channel realizations, the expected length becomes $\bar{L}^* = 475.87/192 = 2.4785$.

To evaluate the total signaling overhead, two bits more are needed per subcarrier to encode the number of loaded bits ($b[k] \in \{0, 2, 4, 6\}$). There can be also some kind of a compression used, e.g. the Huffman coding. The total number of bits needed to express the bit loading per subcarrier at 30 dB averaged over 1000 channel realizations is $33.5424/192 = 0.1747$ bits/subcarrier. In the middle SNR range, the signaling overhead necessary for the bit loading is around 1.5 bits/subcarrier. In the first case, the reason for the low overhead is that all subcarriers are fully loaded ($b[k] = 6, \forall k$) most of the time. In this case, I send only two bits to encode the 64-QAM modulation used because the Huffman coding is not applicable. At the transmitter, it is clear that there are not only two bits loaded in the high SNR region, but all subcarrier are fully loaded. This results into the overhead equal to $2.6532/44 = 0.0603$ bits/subcarrier for 44 OFDM symbols in the WiMAX frame, in which the channel is assumed to be constant.

2.6 Simulation Results

In this section the performance of the proposed *Greedy Power Allocation Algorithm* for coded and uncoded systems is evaluated in a standard compliant WiMAX link level simulator. The Pedestrian B channel model is applied. The target bit error ratio is set equally across all subcarriers in an OFDM symbol. The throughput results are compared to the achievable capacity \bar{C}' for the SISO OFDM system based on the WiMAX standard. It will be derived in Section 3.6 from the MIMO capacity, because the SISO capacity and achievable capacity \bar{C}' are just special cases with one transmit and one receive antenna. In Figure 2.12 the throughput of the *Greedy Power Allocation Algorithm* for coded and uncoded systems is plotted for a Pedestrian B channel. The throughput results are compared to the achievable capacity \bar{C}' . According to Figure 2.12(a), the uncoded throughput calculated before the decoder with the power and bit loading is still about 8 dB apart from the achievable throughput, but only because of the throughput calculation based on the correct frame reception². The results are plotted for the uncoded bit error ratio adaptation to $\text{BER}_0 \in \{0.001, 0.0001\}$. In Figure 2.12(b) the SNR gap is only 3 dB in the low SNR region. It strongly depends on the modulation and

²The whole frame (codeword) is discarded if only a single bit error occurs.

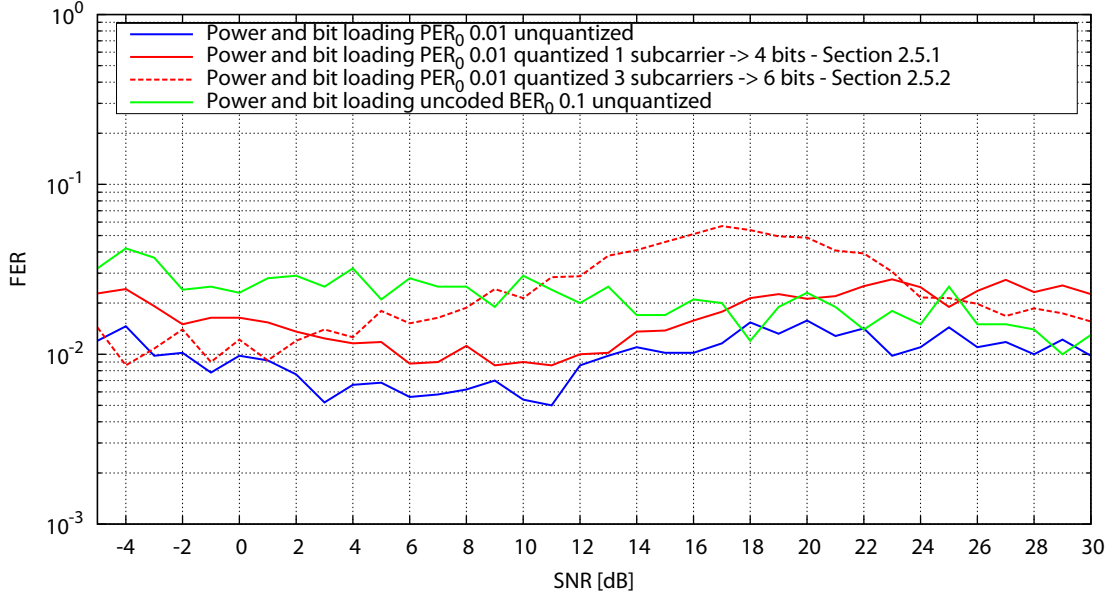
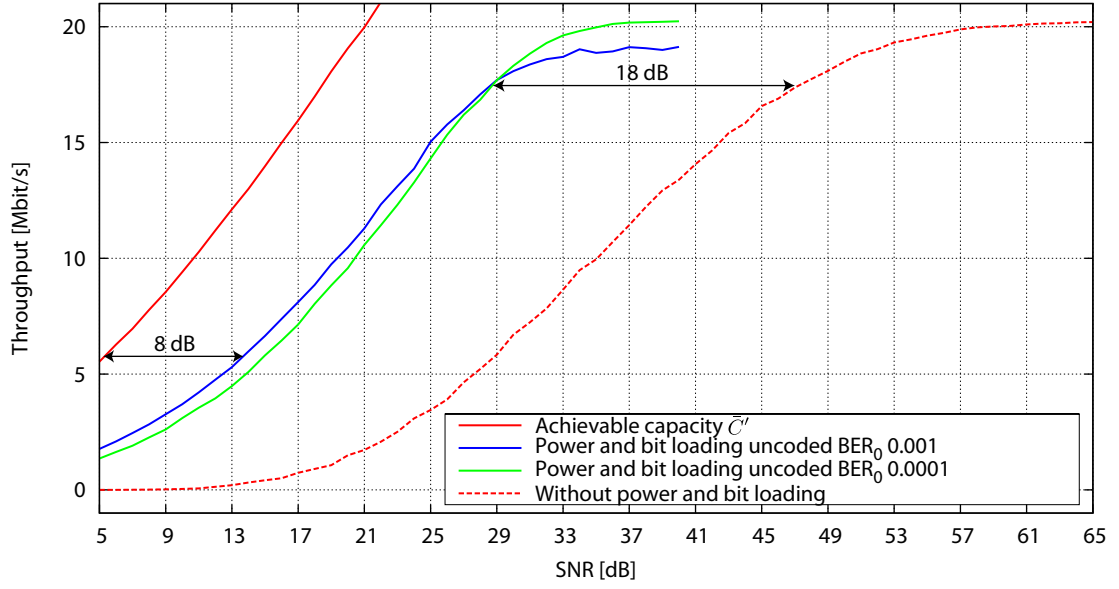


Figure 2.11: Frame error ratio of channel adaptive algorithms in a 1×1 Pedestrian B channel with PER adaptation ($\text{PER}_0 = 10^{-2}$) and uncoded BER adaptation ($\text{BER}_0 = 10^{-1}$).

coding used and can be improved by more efficient schemes. Furthermore, also the effects of the quantization are explored in Figure 2.12(b) for the coded packet error ratio adaptation to 10^{-2} . Comparing the methods from Sections 2.5.1 and 2.5.2, the *Greedy Power Allocation Algorithm* is more sensitive to the number of subcarriers, on which the adaptation takes places, rather than to the number of bits used to express the power level. the adaptive algorithm with the quantization per group of subcarriers from Section 2.5.2 causes the significant performance degradation of about 2 dB in the high SNR region. Using the algorithm from Section 2.5.1 does not cause the performance loss in terms of the throughput.

The gain of a method is in literature usually evaluated in the uncoded scenario, when the performance of the system is not influenced by the code that could already be close to its performance floor or to the achievable capacity, and therefore there would not be so much place for improvement. This can be observed in Figure 2.12 for uncoded and coded systems, respectively. In the case without coding, the power and bit loading method outperforms the standard case by approximately 18 dB, but this gain reduces to 4 dB and less in the coded scenario. The blue dashed curve in Figures 2.12(b) and 2.11 shows that to maximize the throughput, the packet error ratio does



(a) Uncoded throughput with uncoded BER adaptation.

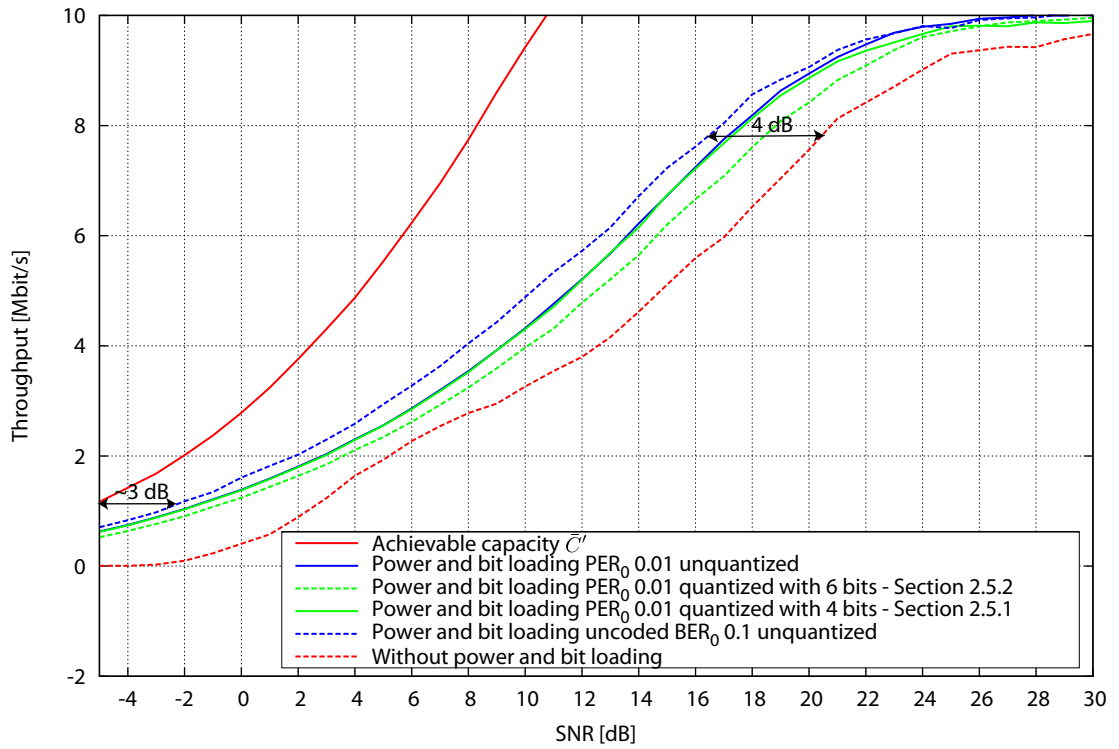

 (b) Coded throughput with PER adaptation (target packet error ratio $\text{PER}_0 = 10^{-2}$).

 Figure 2.12: Performance of channel adaptive algorithms in a 1×1 Pedestrian B channel with and without coding.

not have to be necessarily adapted to 10^{-2} , especially in the lower SNR region, where the codeword length is shorter. Here the uncoded bit error ratio is adapted to 10^{-1} , what results in the coded bit error ratio around 10^{-5} (refer to Figure 2.13) and thus $\text{PER} > 10^{-2}$.

The reason for the significant difference in the performance of the uncoded and coded systems mentioned previously could be the error floor of the $[133\ 171]_8$ half-rate convolutional code. This error floor is slightly above 10^{-5} as shown in Figure 2.13 for 16-QAM modulation in a Pedestrian B channel. This is not so far from the bit error ratio region from 10^{-6} to 10^{-5} (see the blue solid line Figure 2.13) in order to achieve the packet error ratio equal to 10^{-2} . As this adaptation works quite well (see Figure 2.11), the convolutional code has to operate a bit below its actual error floor. It is expected that the higher the error floor of the code, the larger the gain of the *Greedy Power Allocation Algorithm*.

In Figure 2.11 the influence of the quantization on the packet error ratio adaption is shown. Using the adaptive algorithm from Section 2.5.2 with one quantized power level per three subcarriers, the resulting packet error ratio adaptation is not so efficient anymore. When the power level is expressed by 4 bits per each subcarrier as in the algorithm from Section 2.5.1, the quantization causes just slightly less precise packet error ratio adaptation.

If the uncoded non frame-based throughput is calculated, that is, if all the correctly received bits are summed up before the decoder, the achievable capacity \bar{C}' is reached according to Figure 2.14.

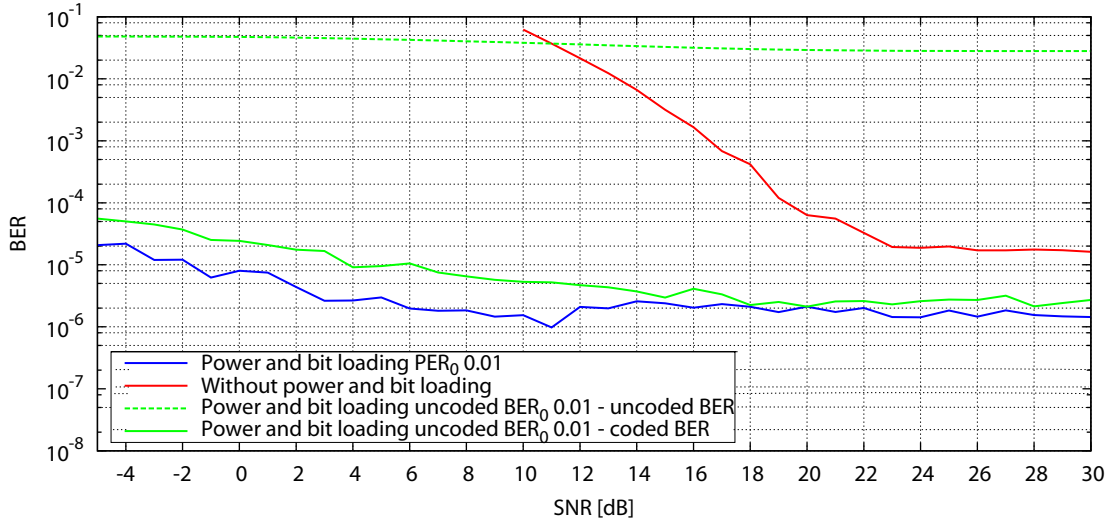


Figure 2.13: Bit error ratio comparison of the channel adaptive algorithms with PER and BER adaptation with the non adaptive system in a 1×1 Pedestrian B channel.

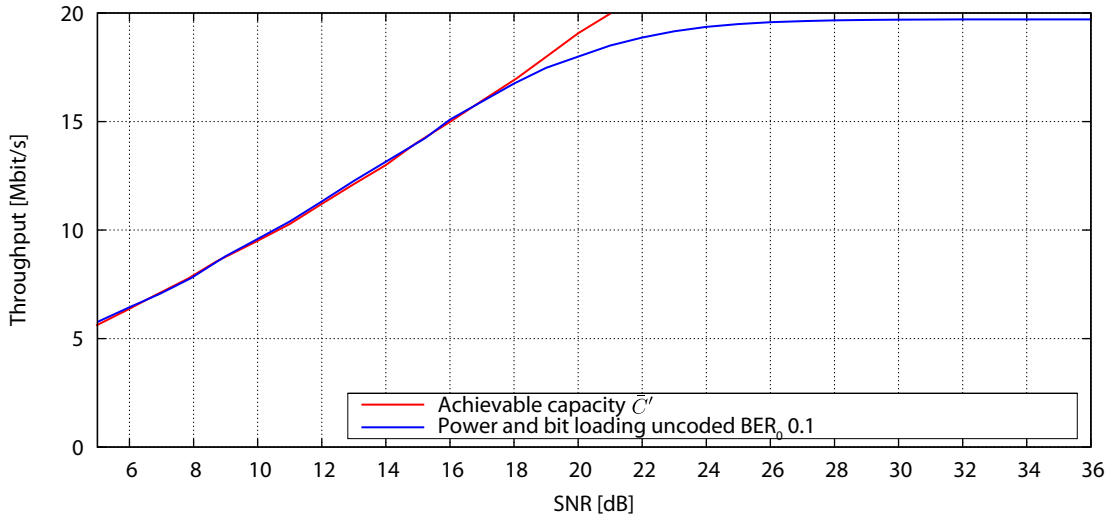


Figure 2.14: Achievable capacity and the non frame-based throughput with uncoded BER adaptation ($BER_0 = 10^{-1}$) in a 1×1 Pedestrian B channel.

3 MIMO OFDM

In this chapter, I extend the discussion from Chapter 2 about SISO OFDM to an OFDM system equipped with N_{FFT} subcarriers (K data subcarriers, the remaining $N_{\text{FFT}} - K$ subcarriers are zero), N_{T} transmit and N_{R} receive antennas (Multiple Input Multiple Output (MIMO) OFDM) signaling over a MIMO frequency-selective fading channel. It can be shown that MIMO OFDM decomposes the otherwise frequency-selective channel of bandwidth B into N_{FFT} orthogonal flat fading MIMO channels, each with bandwidth B/N_{FFT} [13]. Similar as in SISO case, due to the frequency selectivity, different subcarriers also belonging to different transmit-receive antenna combinations experience in general different channel gains. Hence, the total transmit power should be again properly allocated to different subcarriers, based on the available Channel State Information (CSI) at the transmitter. I modify the algorithms from Sections 2.3, 2.4.1 and 2.4.2 to find this appropriate power allocations and to adapt the transmitter to the MIMO channel. Moreover, different pre-filtering coefficients can be also applied to form a transmit beam.

3.1 System Model

The overall system set-up is depicted in Figure 3.1 as the equivalent discrete time baseband model under following considerations. Let k denote the subcarrier index as before, i.e., $k \in \{1, \dots, K\}$. The $N_{\text{T}} \times 1$ transmit symbol vector is given on the k th subcarrier $\mathbf{a}[k] = [a_1[k], a_2[k], \dots, a_{N_{\text{T}}}[k]]^{\text{T}}$, where each information symbol is complex from the given alphabet $\mathcal{A}_{n_{\text{t}}}[k]$, $n_{\text{t}} = 1, \dots, N_{\text{T}}$; consisting of $M_{n_{\text{t}}}[k] = 2^{2m}$, $m = 0, 1, 2, \dots$ symbols for square QAMs. This information symbol $a_{n_{\text{t}}}[k]$ will be then power-loaded with the power level $P_{n_{\text{t}}}[k]$, assuming that the average power of the signal constellation per subcarrier is normalized to one, and transmitted on the k th subcarrier over the n_{t} th transmit antenna. It is assumed that different symbols are transmitted over N_{T} transmit antennas, therefore spatial multiplexing is deployed. It is beneficial in terms

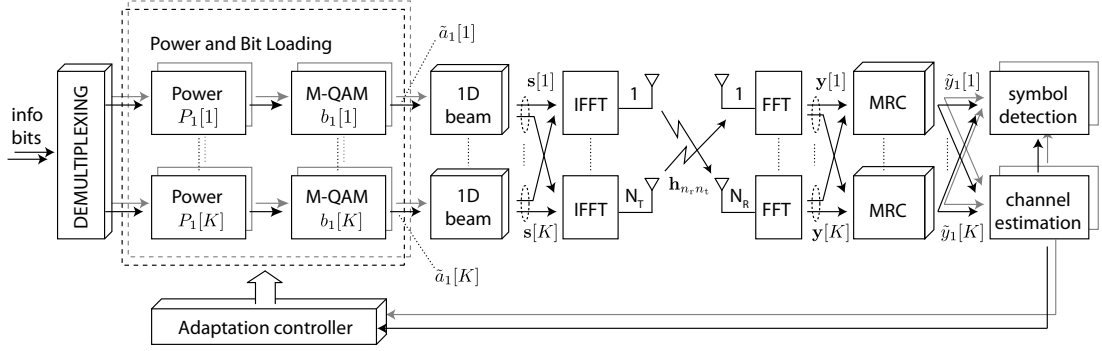


Figure 3.1: Discrete-time equivalent baseband MIMO OFDM model.

of the data throughput because it allows the direct improvement of the capacity by simultaneous transmission of multiple independent data streams from several transmit antennas.

The channel is supposed to be invariant during the transmission of one frame, defined in the WiMAX standard according to Section 1.3, but it may vary from frame to frame. Let $\mathbf{h}_{n_r n_t} := [h_{n_r n_t}[0], h_{n_r n_t}[1], \dots, h_{n_r n_t}[N]]^T$ be the baseband equivalent FIR channel between the n_t -th transmit and the n_r -th receive antenna during the given block with the channel order N . The frequency flat channel between the n_t -th transmit and the n_r -th receive antenna on the k th subcarrier is given by the Discrete Fourier Transform

$$H_{n_r n_t}[k] = \sum_{n=0}^N h_{n_r n_t}[n] \exp\left(\frac{-j2\pi kn}{N_{\text{FFT}}}\right). \quad (3.1)$$

The full MIMO frequency flat channel on the k th subcarrier is modeled by the $N_R \times N_T$ channel matrix $\mathbf{H}[k]$ consisting of the elements from Equation (3.1).

$$\mathbf{H}[k] = \begin{bmatrix} H_{11}[k] & \cdots & H_{1N_T}[k] \\ \vdots & \ddots & \vdots \\ H_{N_R 1}[k] & \cdots & H_{N_R N_T}[k] \end{bmatrix}. \quad (3.2)$$

The $N_R \times 1$ received symbol vector $\mathbf{y}[k] = [y_1[k], y_2[k], \dots, y_{N_T}[k]]^T$ is obtained as

$$\mathbf{y}[k] = \mathbf{H}[k]\tilde{\mathbf{a}}[k] + \mathbf{n}[k], \quad (3.3)$$

where $\tilde{\mathbf{a}}[k] = [\sqrt{P_1[k]}a_1[k], \sqrt{P_2[k]}a_2[k], \dots, \sqrt{P_{N_T}[k]}a_{N_T}[k]]^T$ is the power-loaded trans-

mit vector, and $\mathbf{n}[k]$ is the circularly symmetric complex Additive White Gaussian Noise (AWGN) with the covariance $\mathbb{E}\{\mathbf{n}[k]\mathbf{n}[k]^H\} = N_0\mathbf{I}_{N_R}$ and mean $\mathbb{E}\{\mathbf{n}[k]\mathbf{n}[k]^T\} = \mathbf{0}_{N_R}$. The perfect channel knowledge $\mathbf{H}[k]$ at the transmitter is assumed for each subcarrier $k \in \{1, \dots, K\}$. Again to allow for isolating the transmitter design from the channel estimation issues at the receiver, perfect channel knowledge of the channel $\mathbf{H}[k]$, $\forall k$ at the receiver has to be assumed [9].

3.2 Capacity of MIMO channels

For this section, I leave out the notation per subcarrier k and I deal just with a general $N_R \times N_T$ channel matrix \mathbf{H} , with a rank R , over which the symbol vector \mathbf{a} is transmitted. It is assumed that the channel has a bandwidth of 1 Hz and is frequency flat over this band (what is always fulfilled per subcarrier in MIMO OFDM). I evaluate the capacity of the channel with and without channel knowledge at the transmitter. I also address the type of the channels that maximize the channel capacity.

For the channel matrix \mathbf{H} , a Singular Value Decomposition (SVD) can be always found

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \quad (3.4)$$

where \mathbf{U} and \mathbf{V} are $N_R \times R$ and $N_T \times R$ matrices, respectively, and satisfy $\mathbf{U}^H\mathbf{U} = \mathbf{V}^H\mathbf{V} = \mathbf{I}_R$. The matrix $\mathbf{\Sigma} = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_R\}$ is a diagonal matrix with the singular values σ_r on the main diagonal, fulfilling the condition $\sigma_r > 0$ and $\sigma_r \geq \sigma_{r+1}$. The columns of \mathbf{U} and \mathbf{V} are also known as the input and output singular vectors, respectively. The capacity of the MIMO channel is defined in [13] as

$$C = \max_{f(\tilde{\mathbf{a}})} I(\tilde{\mathbf{a}}; \mathbf{y}), \quad (3.5)$$

where $f(\tilde{\mathbf{a}})$ is the probability distribution of the vector $\tilde{\mathbf{a}}$, and $I(\tilde{\mathbf{a}}; \mathbf{y})$ is the mutual information between vectors $\tilde{\mathbf{a}}$ and \mathbf{y} . The final expression for the capacity of the MIMO channel is given in [13] by

$$C = \max_{\text{Tr}(\mathbf{R}_{\tilde{\mathbf{a}}\tilde{\mathbf{a}}})=N_T} \log_2 \det \left(\mathbf{I}_{N_R} + \frac{1}{N_0} \mathbf{H} \mathbf{R}_{\tilde{\mathbf{a}}\tilde{\mathbf{a}}} \mathbf{H}^H \right) \left[\frac{\text{bits/s}}{\text{Hz}} \right]. \quad (3.6)$$

This capacity is often referred to as the error-free spectral efficiency, or the data rate per unit bandwidth that can be sustained reliably over the MIMO link. Thus given a

bandwidth of B Hz, the maximum achievable data rate over this bandwidth using the MIMO channel is $B \cdot C$ bits/s/Hz.

If the channel is completely unknown to the transmitter, the individual symbols a_{n_t} in the transmit symbol vector \mathbf{a} may then be independent and equi-powered (without the power loading with P_{n_t}) at the transmit antennas and it holds for the signal covariance that $\mathbf{R}_{\mathbf{a}\mathbf{a}} = \mathbf{I}_{N_T}$. The capacity of the MIMO channel in the absence of the channel knowledge at the transmitter is given by

$$C = \log_2 \det \left(\mathbf{I}_{N_R} + \frac{1}{N_0} \mathbf{H}\mathbf{H}^H \right). \quad (3.7)$$

This is not really the Shannon capacity, since with the channel knowledge a signal covariance can be chosen that outperforms $\mathbf{R}_{\mathbf{a}\mathbf{a}} = \mathbf{I}_{N_T}$. Nevertheless, for now it is referred to the expression in Equation (3.7) as a capacity without channel knowledge at the transmitter [13].

In Equation (3.7), the $N_R \times N_R$ positive semi-definite Hermitian matrix $\mathbf{H}\mathbf{H}^H$ can be expressed as $\mathbf{H}\mathbf{H}^H = (\mathbf{U}\Sigma\mathbf{V}^H)(\mathbf{V}\Sigma\mathbf{U}^H) = \mathbf{U}\Lambda\mathbf{U}^H$, by applying $\mathbf{V}^H\mathbf{V} = \mathbf{I}_R$, so that the eigenvalue decomposition of $\mathbf{H}\mathbf{H}^H$ with $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_R\}$ is obtained. In this definition of the eigenvalue decomposition, the eigenvalues $\lambda_r = \sigma_r^2$ are ordered such that $\lambda_r \geq \lambda_{r+1}$ and $\lambda_r > 0$. Given that $\mathbf{H}\mathbf{H}^H = \mathbf{U}\Lambda\mathbf{U}^H$, the capacity in the Equation (3.7) becomes

$$C = \log_2 \det \left(\mathbf{I}_{N_R} + \frac{1}{N_0} \mathbf{U}\Lambda\mathbf{U}^H \right). \quad (3.8)$$

Using the identity $\det(\mathbf{I}_m + \mathbf{A}\mathbf{B}) = \det(\mathbf{I}_n + \mathbf{B}\mathbf{A})$ for matrices $\mathbf{A}_{m \times n}$ and $\mathbf{B}_{n \times m}$ and $\mathbf{U}^H\mathbf{U} = \mathbf{I}_R$, Equation (3.8) simplifies to

$$C = \log_2 \det \left(\mathbf{I}_R + \frac{1}{N_0} \Lambda \right), \quad (3.9)$$

or equivalently

$$C = \sum_{r=1}^R \log_2 \left(1 + \frac{\lambda_r}{N_0} \right). \quad (3.10)$$

Equation (3.10) expresses the capacity of the MIMO channel as the sum of the capacities of R SISO channels, each having the power gain λ_r , for $r = 1, \dots, R$ and the transmit energy $E_s = 1$. Hence, the use of multiple antennas at the transmitter and receiver opens multiple scalar pipes known as modes between the transmitter and receiver. But obviously, without channel knowledge at the transmitter, the individual channel modes

are not accessible.

To answer the question what nature of the channel \mathbf{H} maximizes the capacity, when there is a given total channel power equal to $\|\mathbf{H}\|_F^2 = \sum_{r=1}^R \lambda_r = \zeta$, consider a full-rank MIMO channel with $R = N_T = N_R = N$. The capacity in Equation (3.10) is concave in the variables λ_r , for $r = 1, 2, \dots, N$ and is maximized subject to the constraint $\sum_{r=1}^N \lambda_r = \zeta$, when $\lambda_q = \lambda_r = \zeta/N$ for $q, r = 1, 2, \dots, N$. The channel matrix \mathbf{H} is normalized as follows $\mathbf{H}/\sqrt{N \cdot N}$. Therefore, to maximize the capacity, \mathbf{H} must be an orthogonal matrix, i.e., $\mathbf{H}\mathbf{H}^H = \mathbf{H}^H\mathbf{H} = (\zeta/N)\mathbf{I}_N$ and the resulting capacity is

$$C = N \log_2 \left(1 + \frac{\zeta}{N_0 \cdot N} \right). \quad (3.11)$$

Furthermore, if the elements of \mathbf{H} satisfy $\|H_{n_r n_t}\|^2 = 1/N^2$, then $\|\mathbf{H}\|_F^2 = 1$ and

$$C = N \log_2 \left(1 + \frac{1}{N_0 \cdot N} \right). \quad (3.12)$$

Thus the capacity of an orthogonal MIMO channel is N times the scalar channel capacity.

If the channel is assumed to be known both at the transmitter and receiver, these data pipes can be accessed through linear processing at both link ends. Now suppose that the $R \times 1$ zero mean circularly symmetric complex Gaussian symbol vector $\mathbf{a} = [a_1, a_2, \dots, a_R]^T$ is transmitted over the channel \mathbf{H} with rank R . The vector is multiplied by the matrix \mathbf{V} prior to transmission, at the receiver the received symbol vector \mathbf{y} is multiplied by the matrix \mathbf{U}^H . The effective input-output relation, recalling that $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, is given by

$$\tilde{\mathbf{y}} = \mathbf{U}^H \mathbf{H} \mathbf{V} \mathbf{a} + \mathbf{U}^H \mathbf{n} = \mathbf{\Sigma} \mathbf{a} + \tilde{\mathbf{n}}, \quad (3.13)$$

where $\tilde{\mathbf{y}}$ is the transformed received signal vector of dimension $R \times 1$ and $\tilde{\mathbf{n}}$ is the zero-mean circularly symmetric complex Additive White Gaussian Noise (AWGN) with covariance $E\{\tilde{\mathbf{n}}\tilde{\mathbf{n}}^H\} = N_0 \mathbf{I}_R$. The vector \mathbf{a} must satisfy $E\{\mathbf{a}\mathbf{a}^H\} = N_T$ to meet the total transmit energy constraint. Equation (3.13) shows how the channel \mathbf{H} can be explicitly decomposed into R parallel SISO channels with the input-output relation for the r th one

$$\tilde{y}_r = \sqrt{\lambda_r} a_r + \tilde{n}_r, \quad r = 1, 2, \dots, R. \quad (3.14)$$

Hence, the capacity of the MIMO channel is really expressed by Equation (3.10). Since

the transmitter can access the spatial sub-channels, it can allocate variable power across them to maximize the mutual information. In this way, the power-loaded vector $\tilde{\mathbf{a}}$ is reobtained from Equation (3.3) for the general case of a rank of the channel matrix R that does not have to be equal to N_T . In this case, the transmit power in the r th sub-channel is not equal to one, but to $P_r = \mathbb{E}\{|\tilde{a}_r|^2\}$, for $r = 1, \dots, R$, satisfying $\sum_{r=1}^R P_r = N_T$ (again $T_s = 1$ is assumed, then $P = E_s$).

The mutual information maximization problem now becomes

$$\text{maximize } C(\mathbf{p}) = \sum_{r=1}^R \log_2 \left(1 + \frac{P_r \lambda_r}{N_0} \right) \quad (3.15)$$

$$\text{subject to } \sum_{r=1}^R P_r = N_T. \quad (3.16)$$

The power loading vector $\mathbf{p} = [P_1^{(\text{opt})}, P_2^{(\text{opt})}, \dots, P_R^{(\text{opt})}]$ is the solution of this maximization problem. It can be solved by the water-pouring algorithm that can be found in [13]. In the following I reuse the solution from Section 2.4.1, which behaves similarly to the water-pouring algorithm (it outputs the vector \mathbf{p}) and it does additionally also the bit loading across sub-channels and subcarriers in MIMO OFDM.

3.3 Power and Bit Loading Algorithm for Coded Systems

The goal is to optimize the MIMO OFDM system model depicted in Figure 3.1. Here the perfect knowledge of the channel matrix with rank R is assumed at the transmitter. In this section the constrained optimization problem for MIMO systems is formulated and the solution based on the *Greedy Power Allocation Algorithm* from previous chapter is described. Several modifications to this algorithms are presented. In order to apply these algorithms, it has to be assumed that the transmit signal is pre-filtered by the matrix $\mathbf{V}[k]$ prior to transmission according to Equation (3.13) to guarantee the modal decomposition.

The power levels $P_r[k]$ assigned to the subcarriers with index k and to the sub-channels

with index r are grouped into the transmit power matrix

$$\mathbf{P} = \begin{bmatrix} P_1[1] & \cdots & P_1[K] \\ \vdots & \ddots & \vdots \\ P_R[1] & \cdots & P_R[K] \end{bmatrix}, \quad (3.17)$$

so are the number of bits $b_r[k]$ grouped into the matrix

$$\mathbf{B} = \begin{bmatrix} b_1[1] & \cdots & b_1[K] \\ \vdots & \ddots & \vdots \\ b_R[1] & \cdots & b_R[K] \end{bmatrix}. \quad (3.18)$$

Similarly to the SISO OFDM system,, the objective is to maximize the user throughput T subject to a total transmit power constraint \bar{P} and discrete modulation levels while maintaining a target bit error ratio $\text{BER}_r^{(0)}[k]$ on each subcarrier and in each sub-channel. These target BERs can be identical or different across subcarriers and sub-channels, depending on the specifications. This objective can thus be reformulated as the following constrained optimization problem:

$$\text{maximize } T(\mathbf{P}, \mathbf{B}) = \sum_{r=1}^R \sum_{k=1}^K b_r[k] \quad (3.19)$$

$$\text{subject to } \overline{\text{BER}}_r[k] \leq \text{BER}_r^{(0)}[k] \quad (3.20)$$

$$\sum_{r=1}^R \sum_{k=1}^K P_r[k] \leq N_T \bar{P} \quad (3.21)$$

$$P_r[k] \geq 0 \quad (3.22)$$

$$b_r[k] \in \{0, 1, 2, 3, 4, \dots\} \quad (3.23)$$

The target bit error ratio is again assumed to be identical across subcarriers and sub-channels and equal to $\text{BER}^{(0)}$. It follows that the whole theory from Sections 2.2, 2.3 and 2.4.1 can be simply adapted, just referring to the equations and solutions as per subcarrier and sub-channel, wherever it is applicable. Also the $\rho[k] = \frac{|H[k]|^2}{N_0}$ is changed to $\rho_r[k] = \frac{\sigma_r^2[k]}{N_0}$ and so is the SNR on the k th subcarrier and in the r th sub-channel $\gamma_r[k] = P_r[k] \left(\frac{\sigma_r^2[k]}{N_0} \right)$.

The power cost incurred when loading the l th and $(l-1)$ th bits (two bits are always loaded in one step) to the k th subcarrier and the r th sub-channel is from Equation (2.41)

for coded systems

$$c_r(k, l) = \frac{4 \left[\ln \left(\sum_{d=d_{\text{free}}}^{d_{\text{free}}+4} \beta_r(d, l)[k] \right) - \ln(5\text{BER}^{(0)}) \right]}{g_r(l)c_r(l)\hat{d}_r(l)\rho_r[k]} - \frac{4 \left[\ln \left(\sum_{d=d_{\text{free}}}^{d_{\text{free}}+4} \beta_r(d, l-2)[k] \right) - \ln(5\text{BER}^{(0)}) \right]}{g_r(l-2)c_r(l-2)\hat{d}_r(l-2)\rho_r[k]}, \quad l = 2, 4, 6; \forall k, r. \quad (3.24)$$

So this cost is quantified by the additional power needed to maintain the target bit error ratio performance. For $l = 2$, $g_r(l-2)$ is set to ∞ and $c_r(0) = \hat{d}_r(0) = 1$. In the following algorithm, P_{rem} will be again used to record the remaining power after each iteration, $b_r^{(n)}[k]$ to store the number of bits loaded to the k th subcarrier and the r th sub-channel in the n th iteration, and $P_r^{(n)}[k]$ to denote the power level in iteration step n on the k th subcarrier and in the r th sub-channel. The modified *Optimal MIMO Greedy Power Allocation Algorithm* can be described according to Section 2.4.1 as follows:

1. Initialization step $n = 1$: Set the remaining power equal to the power constraint $P_{\text{rem}} = N_T \bar{P}$. For each subcarrier and sub-channel, set $b_r^{(n)}[k] = P_r^{(n)}[k] = 0$.
2. Compute $c_r(k, b_r^{(n)}[k] + 2)$ for all subcarriers and sub-channels, where $b_r^{(n)}[k] \neq 6$. If $b_r^{(n)}[k] = 6$, then set $c_r(k, b_r^{(n)}[k]) = \infty$. Choose the subcarrier and sub-channel that needs the least power to load two additional bits, i.e. select

$$\{r_0, k_0\} = \arg \min_{r, k} c_r(k, b_r^{(n)}[k] + 2). \quad (3.25)$$

3. If there is not enough power remaining, i.e. if $P_{\text{rem}} < c_r(k, b_r^{(n)}[k] + 2)$ (it always happens if all subcarriers and sub-channels are loaded with 6 bits ($b_r^{(n)}[k] = 6 \forall r, k$)), then exit with $\left\{ P_r[k] = P_r^{(n)}[k], b_r[k] = b_r^{(n)}[k] \right\}_{k=1, r=1}^{K, R}$. Otherwise, load two bits to the k_0 th subcarrier and the r_0 th sub-channel, and update iteration variables:

$$P_{\text{rem}} = P_{\text{rem}} - c_r(k, b_{r_0}^{(N)}[k_0] + 2) \quad (3.26)$$

$$P_{r_0}^{(n)}[k_0] = P_{r_0}^{(n)}[k_0] + c_r(k, b_{r_0}^{(n)}[k_0] + 2) \quad (3.27)$$

$$b_{r_0}^{(n)}[k_0] = b_{r_0}^{(n)}[k_0] + 2. \quad (3.28)$$

4. Loop back to step 2 with $n = n + 1$.

Applying this algorithm, the transmit power matrix \mathbf{P} from Equation (3.17) with optimal values $P_r[k]$ that are maximizing the mutual information in 3.15 is obtained. Moreover, the outcome is also the matrix \mathbf{B} from Equation (3.18) with optimal number of bits $b_r[k]$ that should be loaded to each subcarrier and sub-channel.

Second possibility but sub-optimal one is to load only one sub-channel at a time. The remaining power P_{rem} at the initialization step can be split equally to each substream, that is, $P_{\text{rem}} = N_T \bar{P} / R$. Another option is to use the water-pouring algorithm as already suggested at the end of Section 3.2. The water-pouring algorithm finds the solution to the maximization problem in Equation (3.15). The objective for the maximization is concave in the variables $P_r[k]$, $r = 1, \dots, R$ per each subcarrier and can be maximized using Lagrangian methods. The optimal power allocation policy, $\tilde{P}_r^{(\text{opt})}[k]$, satisfies now according to [13]

$$\tilde{P}_r^{(\text{opt})}[k] = \left(\mu - \frac{N_0}{\lambda_r} \right)_+, \quad r = 1, \dots, R; \quad (3.29)$$

$$\sum_{r=1}^R \tilde{P}_r^{(\text{opt})}[k] = N_T, \quad (3.30)$$

where μ is a constant and $(x)_+$ implies

$$(x)_+ = \begin{cases} x, & \text{if } x \geq 0; \\ 0, & \text{if } x < 0. \end{cases} \quad (3.31)$$

The optimal power allocation is found iteratively through the mentioned water-pouring algorithm described in [13]. Then the remaining power P_{rem} for the r th sub-channel at the initialization step is set to

$$P_{\text{rem}}^{(r)} = \sum_{k=1}^K \tilde{P}_r^{(\text{opt})}[k]. \quad (3.32)$$

The *Suboptimal MIMO Greedy Power Allocation Algorithm* is explained below:

1. Start with the first sub-channel, that is, set $r = 1$.
2. Initialization step $n = 1$: Set the remaining power proportional to power constraint $P_{\text{rem}} = N_T \bar{P} / R$ or to $P_{\text{rem}}^{(r)} = \sum_{k=1}^K \tilde{P}_r^{(\text{opt})}[k]$. For each subcarrier and the r th sub-channel, set $b_r^{(n)}[k] = P_r^{(n)}[k] = 0$.

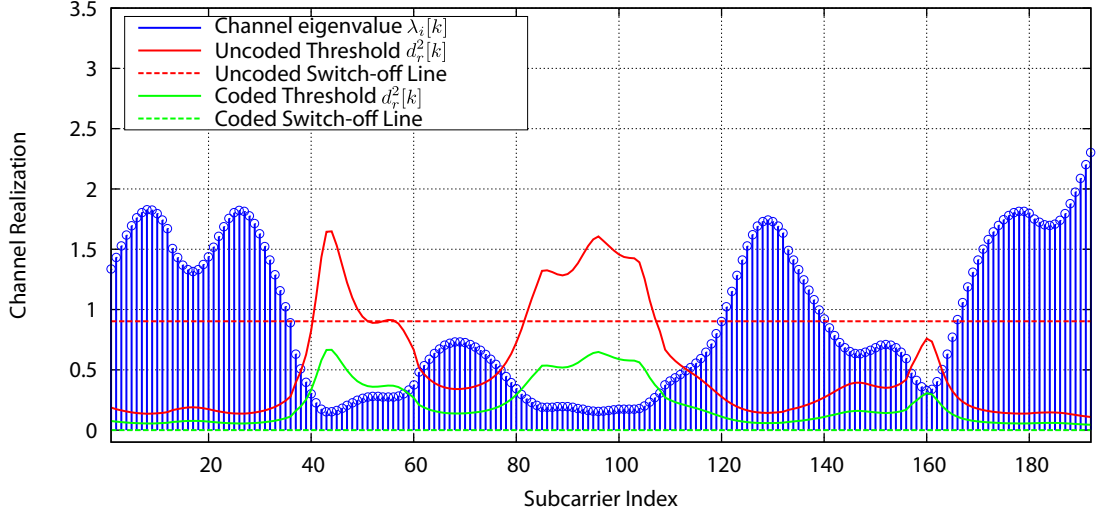


Figure 3.2: Threshold metric $\{d_r^2[k]\}_{k=1}^K$ and a certain channel realization with SNR equal to 16 dB and target bit error ratio $\text{BER}^{(0)} = 10^{-5}$.

3. Compute $c_r(k, b_r^{(n)}[k] + 2)$ for all subcarriers and r th sub-channel, where $b_r^{(n)}[k] \neq 6$. If $b_r^{(n)}[k] = 6$, then set $c_r(k, b_r^{(n)}[k]) = \infty$. Choose the subcarrier that needs the least power to load two additional bits, that is, select

$$k_0 = \arg \min_k c_r(k, b_r^{(n)}[k] + 2). \quad (3.33)$$

4. If there is not enough power remaining, i.e. if $P_{\text{rem}} < c_r(k, b_r^{(n)}[k] + 2)$ (it always happens if all subcarriers are loaded with 6 bits ($b_r^{(n)}[k] = 6 \forall k$)), and if $r = R$, then exit with $\{P_r[k] = P_r^{(n)}[k], b_r[k] = b_r^{(n)}[k]\}_{k=1, r=1}^{K, R}$. If there is not enough power remaining but $r < R$, jump to step 6. Otherwise, load two bits to subcarrier k_0 , and update iteration variables:

$$P_{\text{rem}} = P_{\text{rem}} - c_r(k, b_r^{(n)}[k_0] + 2) \quad (3.34)$$

$$P_r^{(n)}[k_0] = P_r^{(n)}[k_0] + c_r(k, b_r^{(n)}[k_0] + 2) \quad (3.35)$$

$$b_r^{(n)}[k_0] = b_r^{(n)}[k_0] + 2. \quad (3.36)$$

5. Loop back to step 3.
6. Increase $r = r + 1$, if $r \leq R$, loop back to step 2.

In Figure 3.2 the eigenvalues of a certain realization of the frequency-selective fading channel are shown across subcarriers and for the strongest sub-channel with the corresponding threshold metric $d_r^2[k]$ similar to Equation (2.21)

$$d_r^2[k] = \frac{-\ln(5\text{BER}^{(0)})N_0}{\sigma_r^2[k]}. \quad (3.37)$$

It can be seen that the threshold metric is proportional to the inverse of the eigenvalue of the channel and it is corrected appropriately to meet the required bit error ratio. It is compared with the threshold metric for the coded systems derived from Equation (2.39)

$$d_r^2[k] = \left[\ln \left(\sum_{d=d_{\text{free}}}^{d_{\text{free}}+4} \beta_r(d)[k] \right) - \ln(5P_b) \right] \frac{4N_0}{c_r(b_r[k]) \cdot \hat{d}_r(b_r[k]) \cdot \sigma_r^2[k]}. \quad (3.38)$$

The effects depicted in Figures 2.2 and 2.3 can be shown also in the MIMO case, with the only difference that the behavior of singular values is examined, not the channel magnitude across subcarriers.

The only problem with the optimality arises when the algorithm from Section 2.4.2 needs to be applied. As the power has to be allocated jointly across the subcarriers and the sub-channels, one codeword will be spread over all sub-channels. In this way, the target packet error ratio can be guaranteed per each subcarrier and sub-channel.

Sometimes it is convenient to have one codeword per spatial substream. In this case, the *Optimal MIMO Greedy Power Allocation Algorithm* may be used just with the adaptation of the packet error ratio according to the weaker stream. Another possibility is to implement the *Suboptimal MIMO Greedy Power Allocation Algorithm* with two modifications as mentioned above and the packet error ratio will be adapted in each sub-channel separately. Results for these algorithms will be shown in Section 3.6.

3.4 Linear Pre-filtering

In the previous section I assumed that the transmit signal was pre-filtered by the matrix $\mathbf{V}[\mathbf{k}]$ prior to transmission according to Equation (3.13). In this section, I show that this pre-filtering that leads to the modal decomposition is the optimal solution.

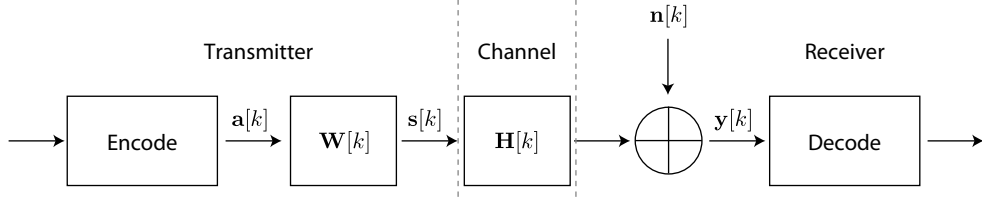


Figure 3.3: A linear pre-filtering designed for a MIMO OFDM system by exploiting channel knowledge.

For this purpose, it has to be assumed that the transmitter knows the channel perfectly.

As it is clear from Section 2.2, my main aim is to maximize the data throughput of a user or, in other words, the capacity of the system. The spatial multiplexing is deployed and the total power constraint needs to be fulfilled at the transmitter. The equal number of the transmit and receive antennas is assumed, therefore $N_T = N_R = N$ is used. One possible framework for exploiting the channel knowledge is called the linear pre-filtering and is shown in Figure 3.3. Obviously, it has to be performed on the subcarrier basis. By applying this scheme, just slightly modified input-output relation is obtained for the frequency-flat channel

$$\mathbf{y}[k] = \mathbf{H}[k]\mathbf{W}[k]\mathbf{a}[k] + \mathbf{n}[k] = \mathbf{H}[k]\mathbf{s}[k] + \mathbf{n}[k], \quad (3.39)$$

where $\mathbf{a}[k]$ satisfies $E\{\mathbf{a}[k]\mathbf{a}[k]^H\} = \mathbf{I}_N$, $\mathbf{s}[k] = \mathbf{W}[k]\mathbf{a}[k]$ and the channel matrix $\mathbf{H}[k]$ is normalized as follows $\mathbf{H}[k]/\sqrt{N \cdot N}$. The covariance matrix of the transmit signal $\mathbf{R}_{ss}[k]$ is given by

$$\mathbf{R}_{ss}[k] = \mathbf{W}[k]\mathbf{W}[k]^H. \quad (3.40)$$

What follows is that the pre-filter matrix $\mathbf{W}[k]$ must satisfy the power constraint at the transmitter. If the total average transmit power per subcarrier and all transmit antennas is required to be equal to N , then

$$\|\mathbf{W}[k]\|_F = \text{Tr}(\mathbf{R}_{ss}[k]) = N. \quad (3.41)$$

The expression for the capacity of the MIMO channel from Equation (3.6) now modified per subcarrier k is

$$C[k] = \max_{\text{Tr}(\mathbf{R}_{ss}[k])} \log_2 \det \left(\mathbf{I}_{N_R} + \frac{1}{N_0} \mathbf{H}[k] \mathbf{R}_{ss}[k] \mathbf{H}[k]^H \right) \left[\frac{\text{bits/s}}{\text{Hz}} \right]. \quad (3.42)$$

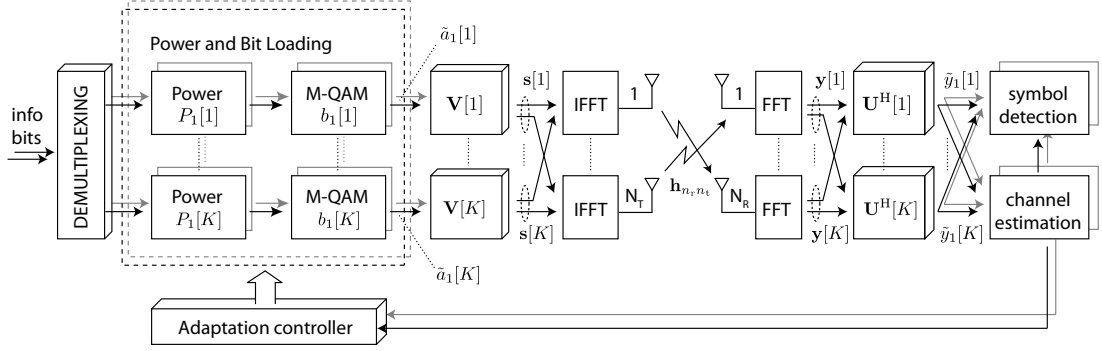


Figure 3.4: Optimized discrete-time equivalent baseband MIMO OFDM model with linear pre-filtering.

In the following, it is further assumed that the channel $\mathbf{H}[k]$ is not only perfectly known to the transmitter and the receiver, but it is also full-rank, i.e., $r = N$. Compared to the expression in Equation (3.7), it is now possible to make use of the channel knowledge at the transmitter and to maximize the capacity over $\mathbf{R}_{ss}[k]$. When the optimal $\mathbf{R}_{ss}^{\text{opt}}[k]$ is found, the optimal pre-filtering matrix $\mathbf{W}^{\text{opt}}[k]$ can be derived from Equation (3.40) as

$$\mathbf{W}^{\text{opt}}[k] = \mathbf{Q}_{\text{opt}}[k] \mathbf{\Lambda}_{\text{opt}}^{1/2}[k], \quad (3.43)$$

where $\mathbf{Q}_{\text{opt}}[k] \mathbf{\Lambda}_{\text{opt}}[k] \mathbf{Q}_{\text{opt}}^H[k]$ is the eigenvalue decomposition of $\mathbf{R}_{ss}^{\text{opt}}[k]$. Once the optimal power allocation across the spatial sub-channels and subcarriers is determined so that $\tilde{\mathbf{a}}[k] \forall k$ is given, then from Equation (3.13) it follows that $\mathbf{s}[k] = \mathbf{V}[k] \tilde{\mathbf{a}}[k]$. The optimal covariance matrix $\mathbf{R}_{ss}^{\text{opt}}[k]$ is thus given by

$$\mathbf{R}_{ss}^{\text{opt}}[k] = \mathbf{V}[k] \mathbf{R}_{\tilde{\mathbf{a}}\tilde{\mathbf{a}}}^{\text{opt}}[k] \mathbf{V}^H[k], \quad (3.44)$$

where $\mathbf{R}_{\tilde{\mathbf{a}}\tilde{\mathbf{a}}}^{\text{opt}}[k]$ is an $N \times N$ diagonal matrix (since the elements of $\tilde{\mathbf{a}}$ are independent) equal to

$$\mathbf{R}_{\tilde{\mathbf{a}}\tilde{\mathbf{a}}}^{\text{opt}}[k] = \text{diag}\{P_1^{\text{opt}}[k], P_2^{\text{opt}}[k], \dots, P_N^{\text{opt}}[k]\}. \quad (3.45)$$

Hence, from Equation (3.43) it follows

$$\mathbf{W}^{\text{opt}}[k] = \mathbf{V}[k] \left(\mathbf{R}_{\tilde{\mathbf{a}}\tilde{\mathbf{a}}}^{\text{opt}}[k] \right)^{1/2}. \quad (3.46)$$

When Equation (3.46) is substituted into the first part of Equation (3.39), left-multiplying by $\mathbf{U}^H[k]$ (multiplying with a unitary matrix at the receiver does not alter mutual in-

formation [13]) and Equation (3.4) is used per subcarrier k , Equation (3.39) reduces to

$$\tilde{\mathbf{y}}[k] = \mathbf{\Sigma}[k]\tilde{\mathbf{a}}[k] + \tilde{\mathbf{n}}[k], \quad (3.47)$$

where $\tilde{\mathbf{n}}[k] = \mathbf{U}^H[k]\mathbf{n}[k]$ and $\tilde{\mathbf{a}}[k] = \left(\mathbf{R}_{\tilde{\mathbf{a}}\tilde{\mathbf{a}}}^{\text{opt}}[k]\right)^{1/2} \mathbf{a}[k]$. This is the similar result as the one from Section 3.2 in Equation (3.13) using the modal decomposition and *Greedy Power Allocation Algorithm* from Section 3.3 for searching the power levels in $\left(\mathbf{R}_{\tilde{\mathbf{a}}\tilde{\mathbf{a}}}^{\text{opt}}[k]\right)^{1/2}$ in Equation (3.46). Moreover, also the appropriate signal alphabet $\mathcal{A}_r[k]$ is assigned to the r th sub-channel and the k th subcarrier according to the *Greedy Power Allocation Algorithm*. The resulting optimized MIMO OFDM system is redrawn from Figure 3.4 with small modifications for the optimal pre-filtering.

Inserting Equation (3.44) into Equation (3.42), and using the singular value decomposition of the channel matrix $\mathbf{H}[k]$, finally the capacity or the error-free spectral efficiency of the MIMO channel is obtained as a sum of the N parallel SISO channels

$$C[k] = \sum_{r=1}^N \log_2 \left(1 + \frac{P_r^{\text{opt}}[k]\lambda_r[k]}{N_0} \right). \quad (3.48)$$

3.5 Quantized Feedback

In this section I discuss the possibilities for the feedback of the channel state information back to the transmitter. The *Optimal MIMO Greedy Power Allocation Algorithm* is further modified to implement the quantization of the power levels and of the pre-filtering matrix $\mathbf{V}[k]$. Furthermore, the resulting overhead per subcarrier is evaluated when the Huffman source coding is applied to the quantized power levels, number of loaded bits and to the quantized pre-filtering matrix $\mathbf{V}[k]$.

The same feedback scheme is assumed as proposed in Section 2.5. The new problem that arises in the MIMO case is the need for the quantization of the pre-filtering matrix $\mathbf{V}[k]$. In the simulations in the following section, the 2×2 system is assumed ($R = N = 2$). Two OFDM symbols are sent from two transmit antennas, thus the signaling is performed for $192 \cdot 2 = 384$ subcarriers.

The pre-filtering matrix has four complex elements $v_{n_r n_t}[k]$ for $n_r = n_t = 1, 2$. Therefore, four phases $\varphi_{n_r n_t}[k]$ and four amplitudes $|v_{n_r n_t}[k]|$ have to be quantized and fed back to the transmitter per each subcarrier and both sub-channels. The pre-filtering

matrix $\mathbf{V}[k]$ is unitary, that is, $\mathbf{V}^H[k]\mathbf{V}[k] = \mathbf{I}_N$. It follows that only one amplitude per subcarrier and two sub-channels has to be fed back and the quantized pre-filtering matrix $\mathbf{V}_{\text{quant}}[k]$ can be expressed as follows

$$\mathbf{V}_{\text{quant}}[k] = \begin{bmatrix} |v_{\text{quant}}[k]| & \sqrt{(1 - |v_{\text{quant}}[k]|^2)} \\ \sqrt{(1 - |v_{\text{quant}}[k]|^2)} & |v_{\text{quant}}[k]| \end{bmatrix} \odot \begin{bmatrix} e^{(j\varphi_{11}^{(\text{quant})}[k])} & e^{(j\varphi_{12}^{(\text{quant})}[k])} \\ e^{(j\varphi_{21}^{(\text{quant})}[k])} & e^{(j\varphi_{22}^{(\text{quant})}[k])} \end{bmatrix}, \quad (3.49)$$

where \odot is an element-wise multiplication and j is an imaginary unit.

The power levels $P_r[k]$ are quantized in a floating point to 4 bits with the same quantizer as in Section 2.5. They are now fed back for 384 subcarriers (2 spatial substreams). The amplitude $|v_{\text{quant}}[k]|$ is obtained by quantization to seven-bit-long fraction length. For this purpose, a quantizer in an unsigned fixed-point mode rounding to the nearest value and saturating on overflow is used. The phases $\varphi_{n_r n_t}^{(\text{quant})}[k]$ are also obtained by quantization to seven bits, but only four bits out of seven are dedicated to the fraction length. The previously mentioned quantizer is used, now operating in a signed fixed-point mode.

Applying Huffman code on the quantized random variables $|v^{(\text{quant})}[k]|$, $\varphi_{n_r n_t}^{(\text{quant})}[k]$ and $P_r^{(\text{quant})}[k]$ and averaging the number of signaling bits over 1000 channel realizations at 30 dB, the expected lengths become $\bar{L}_1^* = 1142.53/384 = 2.9753$, $\bar{L}_2^* = 3435.714/384 = 8.9472$ and $\bar{L}_3^* = 802.4673/384 = 2.0898$, respectively.

Furthermore, also the number of bits per subcarrier and sub-channel has to be signalled. Using Huffman code again, the total number of bits needed to express the bit loading per subcarrier at 30 dB averaged over 1000 channel realizations under same considerations as in Section 2.5 is $59.7060/384 = 0.1555$ bits/subcarrier. This results into the overhead equal to $(2.9753+8.9472+2.0898+0.1555)/44 = 0.3220$ bits/subcarrier for 44 OFDM symbols in the WiMAX frame, in which the channel is assumed to be constant.

3.6 Simulation results

The performance of the proposed algorithms and techniques is again evaluated in the WiMAX link level simulator in an uncorrelated Pedestrian B channel. The target bit error ratio is assumed to be equal across all subcarriers and sub-channels and equal to $\text{BER}^{(0)}$. First, I derive the expression for the achievable capacity that will be used as a

performance bound for the simulated data throughput. For this purpose, the capacity $\bar{C}'[k]$ per subcarrier k of a MIMO channel is defined as the ensemble average of the information rate over the random realizations of the channel matrix $\mathbf{H}[k]$. I still assume that the channel matrix $\mathbf{H}[k]$ is full-rank, that is, $R = N$, what is always fulfilled in the simulations with an uncorrelated Pedestrian B channel.

The capacity per each subcarrier k averaged over the number of simulated channel realizations N_{sim} when the channel is known to the transmitter is given by

$$\bar{C}'[k] = \frac{1}{N_{\text{sim}}} \sum_{s=1}^{N_{\text{sim}}} \sum_{r=1}^N \log_2 \left(1 + \frac{P_r^{\text{opt}}[k] \lambda_r[k]}{N_0} \right). \quad (3.50)$$

where Equation (3.48) has been used. In the case of unavailable channel knowledge at the transmitter, this capacity results in

$$\bar{C}''[k] = \frac{1}{N_{\text{sim}}} \sum_{s=1}^{N_{\text{sim}}} \sum_{r=1}^N \log_2 \left(1 + \frac{\lambda_r[k]}{N_0} \right). \quad (3.51)$$

where Equation (3.10) has been used and modified for the k th subcarrier. It is important to note that Equation (3.10) was derived from Equation (3.7) by using the singular value decomposition of the channel matrix $\mathbf{H}[k]$, but the pre-filtering matrix $\mathbf{V}[k]$ is not needed at the transmitter prior to transmission.

The overall achievable capacity \bar{C}' of the OFDM system can now be calculated from Equations (3.50) and (3.51) by summing the capacities of all data subcarriers (192 data subcarriers are used in the WiMAX standard)

$$\bar{C}' = \frac{F}{N_{\text{sim}}} \sum_{s=1}^{N_{\text{sim}}} \sum_{k=1}^{192} \sum_{r=1}^N \log_2 \left(1 + \frac{P_r^{(s)}[k] \lambda_r^{(s)}[k]}{N_0} \right), \quad (3.52)$$

in case of the channel knowledge at the transmitter, and

$$\bar{C}'' = \frac{F}{N_{\text{sim}}} \sum_{s=1}^{N_{\text{sim}}} \sum_{k=1}^{192} \sum_{r=1}^N \log_2 \left(1 + \frac{\lambda_r^{(s)}[k]}{N_0} \right), \quad (3.53)$$

when no channel knowledge is assumed at the transmitter. Since the transmission of an OFDM signal requires also the transmission of a cyclic prefix to avoid inter-symbol interference, and a preamble for the synchronization and channel estimation, the achievable capacity given by Equations (3.52) and (3.53) includes the correction factor F . This

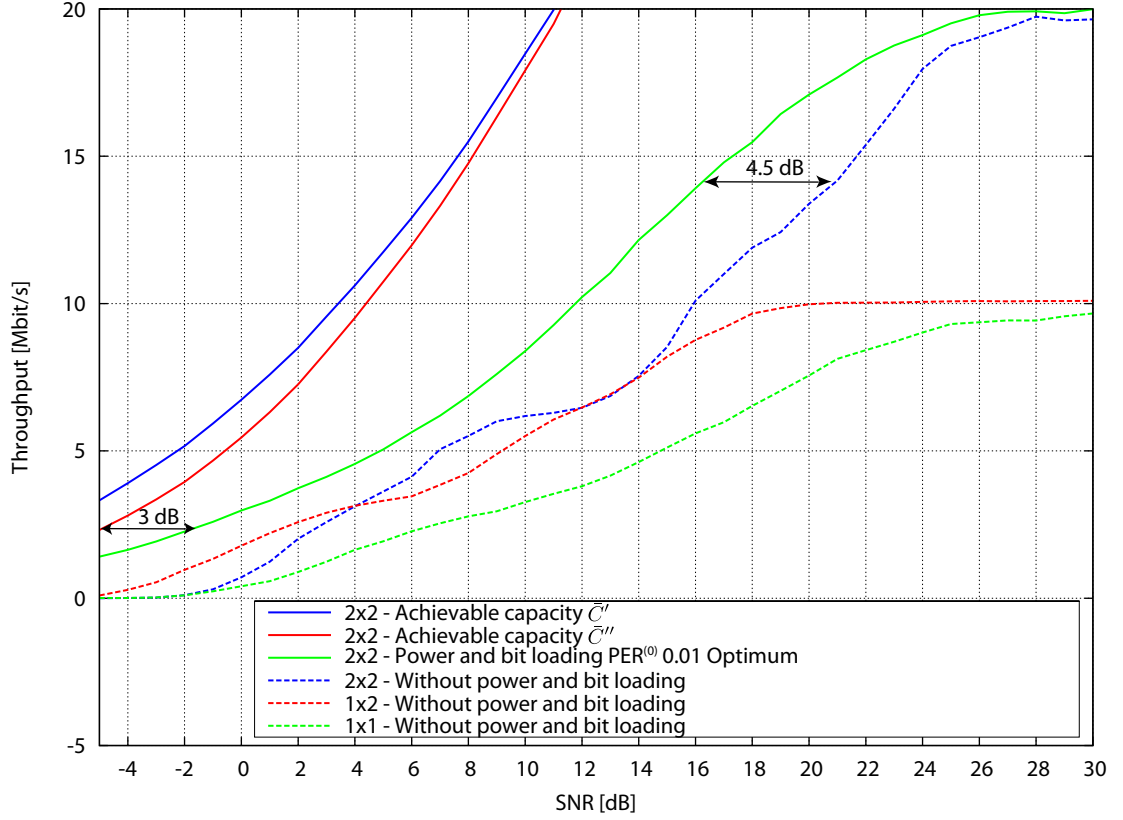
correction term accounts for these inherent system losses and is defined according to [27] as follows

$$F = \frac{1}{1+G} \cdot \frac{1/T_{\text{sample}}}{256} \cdot \frac{N_s}{N_{\text{total}}}, \quad (3.54)$$

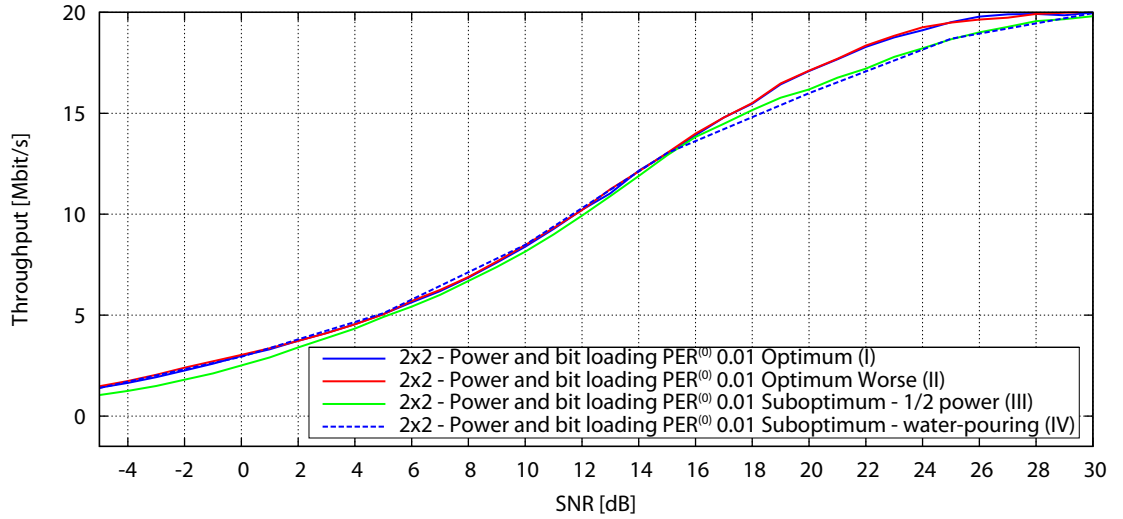
where according to Section 1.3, G ($1/4$ in the simulations) corresponds to the ratio of the cyclic prefix time and useful OFDM symbol time T_s , N_s (44 in the simulations) is the number of OFDM data symbols, N_{total} (47 in the simulations) is the total number of OFDM symbols in one transmission frame (also with the preamble), and T_{sample} is the sampling rate of the transmit signal. Therefore, the factor $(1/T_{\text{sample}})/256$ is equivalent to the available bandwidth per subcarrier (the FFT size is equal to 256) [27].

In Figure 3.5 the performance of the channel adaptive algorithms from Section 3.3 is evaluated in an uncorrelated Pedestrian B channel. The coded system is taken into account with the packet error ratio adaptation (the target packet error ratio is $\text{PER}^{(0)} = 10^{-2}$). In Figure 3.5(a) the *Optimal MIMO Greedy Power Allocation Algorithm* is compared to the achievable capacity \bar{C}' and \bar{C}'' , and to the 2×2 , 1×2 and 1×1 system without power and bit loading. The 2×2 system using this algorithm outperforms the standard non adaptive 2×2 system by maximum 4.5 dB. The difference in the low SNR region from the achievable capacity \bar{C}'' calculated from Equation (3.53) is as low as 3 dB. The difference is due to the coding that is far away from the optimal one. With increasing SNR, the difference is growing significantly, as just the half-rate convolutional code is used. As expected, the achievable capacity \bar{C}' with the channel known at the transmitter is higher than the achievable capacity \bar{C}'' in the case of unknown channel. This advantage reduces at higher SNR, as can be observed in Figure 3.8(a) and is proved in [13].

In Figure 3.5(b) the *Optimal MIMO Greedy Power Allocation Algorithm* (I) is compared to the two modifications of the *Suboptimal MIMO Greedy Power Allocation Algorithm*. When the *Optimal MIMO Greedy Power Allocation Algorithm* (II) is slightly modified to adapt the packet error ratio according to the worse stream (the second stream is always worse because it belongs to the weaker channel eigenvalue), the throughput remains almost unchanged according to the blue and red solid lines. The *Suboptimal MIMO Greedy Power Allocation Algorithm* (III) with the same power constraint for each stream loses 2 dB in the throughput performance in the high SNR region (see the green solid line). The same holds for this suboptimal algorithm (IV) when the total transmit power assigned to each substream is calculated by the water-pouring algorithm as described in Section



(a) Throughput comparison with capacity and non adaptive systems.



(b) Throughput comparison of channel adaptive algorithms.

Figure 3.5: Performance of channel adaptive algorithms in an uncorrelated 2×2 Pedestrian B channel with PER adaptation (target packet error ratio $\text{PER}^{(0)} = 10^{-2}$).

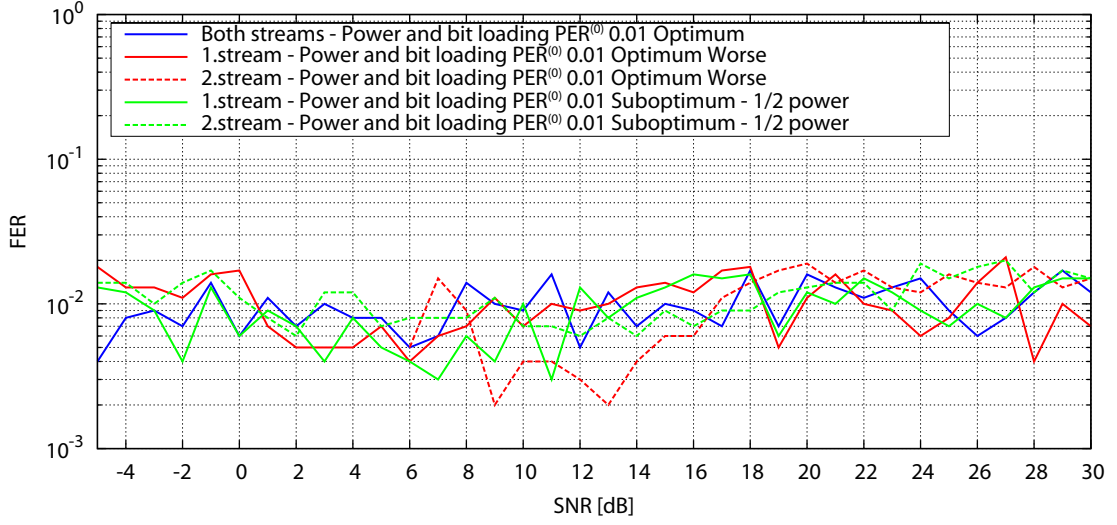
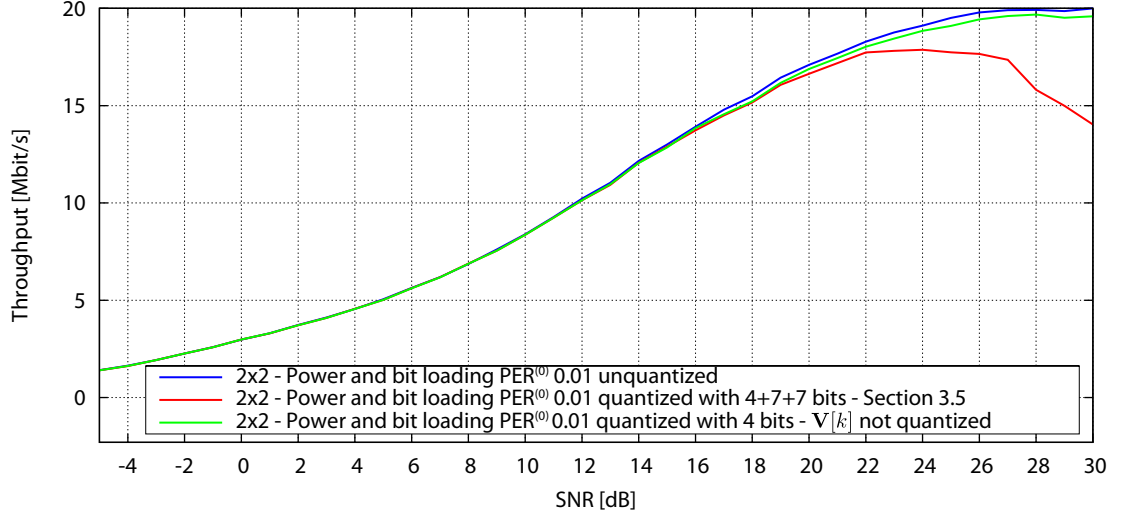


Figure 3.6: Frame error ratio adaptation for the target packet error ratio $PER^{(0)} = 10^{-2}$.

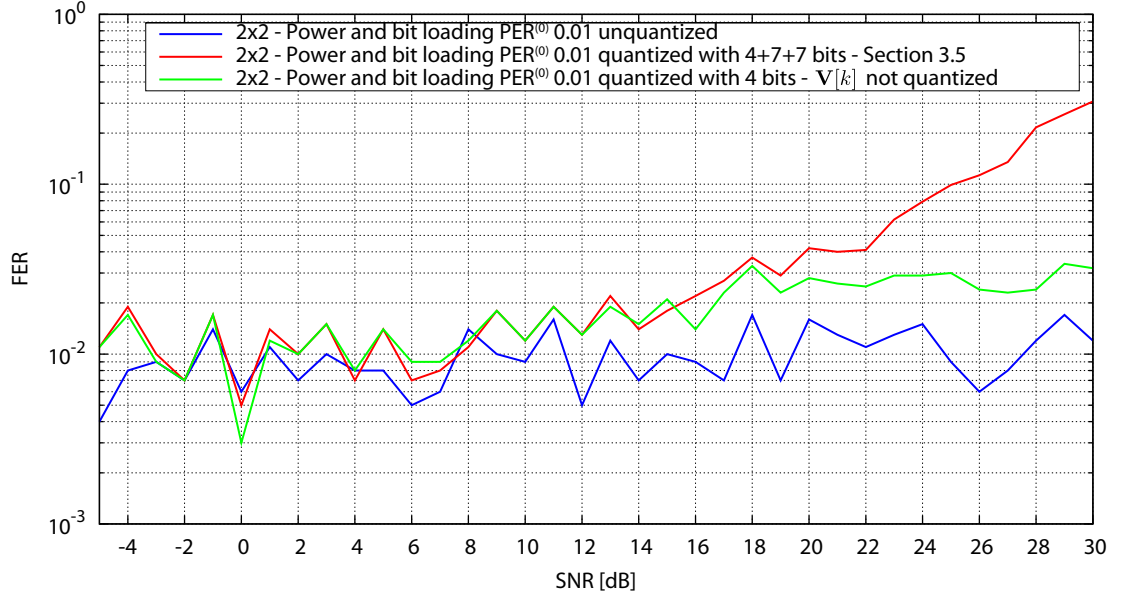
3.3 (refer to the blue dashed line). The reason for this is that the standard water-pouring algorithm is based on the uncoded system. Therefore, it assigns at the high SNR values almost the equal power to both spatial substreams. The channel coding enables to save some power on the stronger substream when all subcarriers are fully loaded (there are 6 bits per each subcarrier). Thus, it is optimal to allocate the remaining power to the weaker substream what is guaranteed only by the *Optimal MIMO Greedy Power Allocation Algorithm*.

In Figure 3.6 the resulting frame error ratio is plotted for the target packet error ratio equal to 10^{-2} . Several approaches are compared. When the *Optimal MIMO Greedy Power Allocation Algorithm* is slightly modified to adapt the packet error ratio according to the worse stream (the second stream is always worse because it belongs to the weaker channel eigenvalue), it can be observed (red solid and dashed curves in Figure 3.6) that in the low SNR region up to 6 dB, no bits are allocated to the second stream. The adaptation performance of the *The Optimal MIMO Greedy Power Allocation Algorithm* and *Suboptimal MIMO Greedy Power Allocation Algorithm* with the power split equally among the streams is satisfactory in the large SNR range.

In Figure 3.7 the effects of the quantization are explored for the coded packet error ratio adaptation to 10^{-2} . The quantization from Section 3.5 is applied and compared to the *Optimal MIMO Greedy Power Allocation Algorithm* from Section 3.3 and to the case when only the power levels are quantized according to Section 3.5. It follows that the



(a) Throughput comparison .

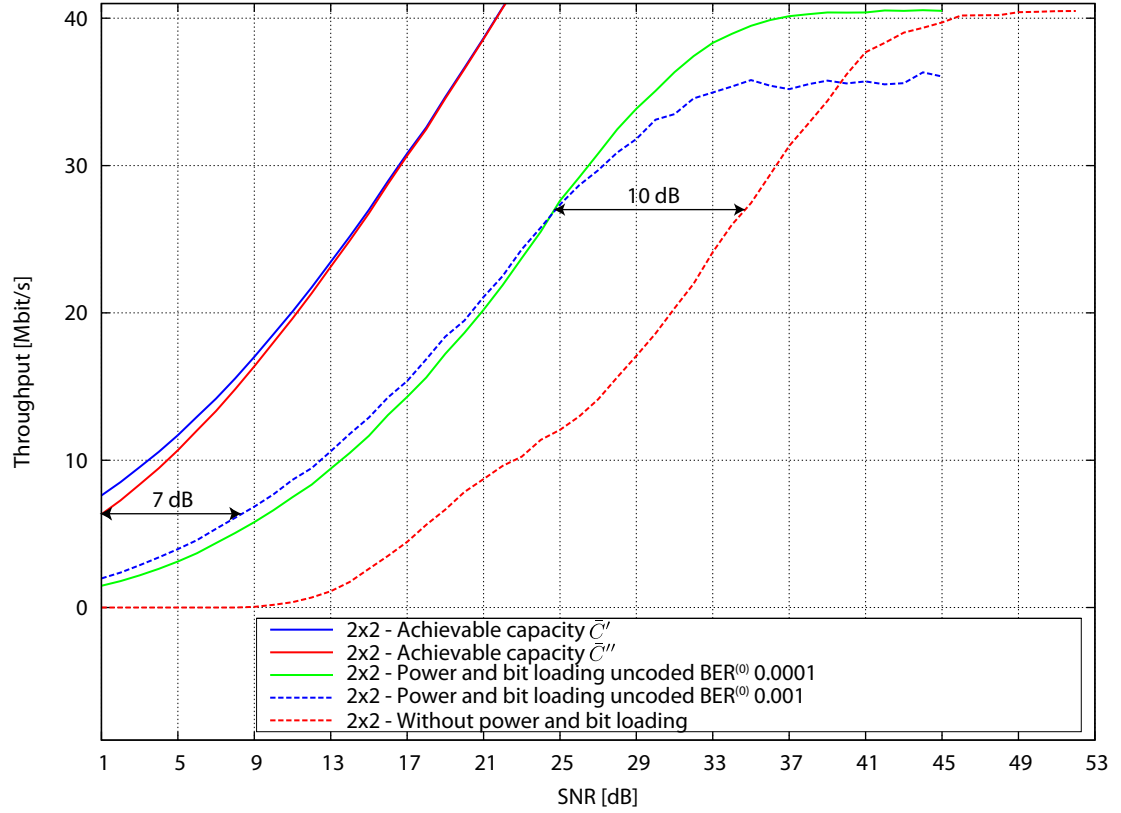


(b) Frame error ratio adaptation.

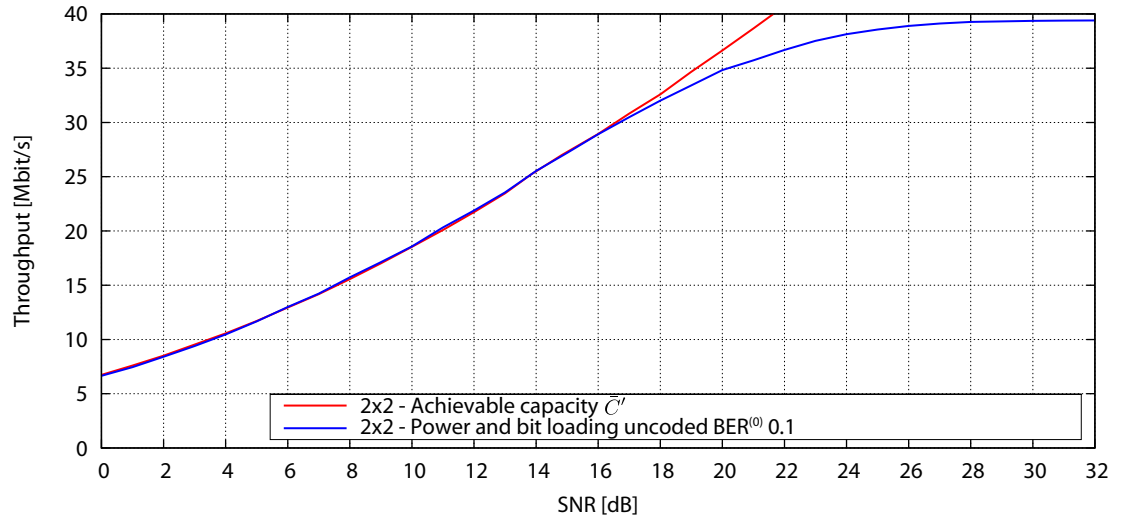
Figure 3.7: Performance of channel adaptive algorithms in an uncorrelated 2×2 Pedestrian B channel with PER adaptation (target packet error ratio $\text{PER}^{(0)} = 10^{-2}$) with and without quantization.

Optimal MIMO Greedy Power Allocation Algorithm is very sensitive to the quantization of the pre-filtering matrix $\mathbf{V}[k]$ in the high SNR region as it can be seen in Figure 3.7(a). In this region, this method does not achieve the maximum throughput and it leads to the performance degradation that starts at 22 dB. This can be also observed in Figure 3.7(b), where the frame error ratio is increasing significantly with the increasing SNR. When only the power levels are quantized to four bits, the quantization does not cause the significant performance loss in terms of the throughput and the packet error ratio adaptation is just slightly less precise.

In Figure 3.8 the performance of the *Optimal MIMO Greedy Power Allocation Algorithm* modified according to Section 2.3 for uncoded systems is evaluated in an uncorrelated Pedestrian B channel. In this case, the uncoded bit error ratio is adapted to several values and the uncoded throughput based on the number of the correctly received frames before the decoder is shown. The performance gain is much higher (about 10 dB) compared to coded systems, as can be seen in Figure 3.8(a). But without the channel coding the system is not so robust anymore. In Figure 3.8(b) the non frame-based throughput is plotted for the 2×2 channel adaptive system with the target bit error ratio $\text{BER}^{(0)} = 0.1$. This throughput is calculated from the number of bits received correctly before the decoder. It achieves the achievable capacity \bar{C}' with the channel knowledge at the transmitter.



(a) Uncoded throughput comparison with capacity and non adaptive system.



(b) Uncoded non frame-based throughput comparison with capacity.

Figure 3.8: Performance of channel adaptive algorithm in an uncorrelated 2×2 Pedestrian B channel without coding (uncoded BER adaptation with $\text{BER}^{(0)} \in \{0.0001, 0.001, 0.1\}$).

4 Conclusions and Further Work

In this master thesis, the known *Greedy Power Allocation Algorithm*, proposed for example in [9, 11] was extended to coded OFDM systems with the possibility to adapt also the packet error ratio. The framework created here enables to change the implemented model for the modulation and coding scheme easily. For this purpose, the approximation of the bit error ratio has to be known for the new modulation and coding combination in the AWGN channel. An example how to obtain such approximations was shown for the $[133\ 171]_8$ half-rate convolutional code in combination with a 4-, 16- and 64-QAM modulator.

The *Greedy Power Allocation Algorithm* for coded systems with the packet error ratio adaptation seems to be a promising method how to adapt the transmitter design to the frequency selective channel while increasing the throughput performance and the reliability of the system. To the best of my knowledge, this is the first attempt to evaluate the performance of the water-pouring based solution in the more realistic scenario of a coded WiMAX system. It remains to further explore the behavior of the proposed algorithm and the possible gains when more efficient channel coding is deployed.

However, this approach requires a lot of feedback, especially in the MIMO case. In practice, the perfect channel knowledge is not available at the transmitter, therefore it was proposed to send the assigned power levels and bits from the receiver to the transmitter. For this case, the effects of the quantized power levels on the system performance were explored and the signaling overhead was estimated. There is a significant performance degradation in the MIMO case that is very sensitive to the quantization of the optimum pre-filtering matrix. This suggests to use some suboptimal pre-filtering that is not based on the perfect channel knowledge at the transmitter. For example in the case of a correlated channel, there exist methods based on the channel covariance matrix. By changing the pre-filtering, also the proposed optimal power allocation policy

is not possible anymore and has to be adapted to different pre-filtering, because the individual SISO parallel channels are not appropriately opened.

Moreover, the complexity of the *Greedy Power Allocation Algorithm* grows linearly with the number of bits and the number of subcarriers. Obviously, the complexity is considerably large when the number of bits and subcarriers is large. The maximum number of iterations for one particular bit and power level assignment is for example for the 2×2 MIMO OFDM system with 192 data subcarriers equal to $3 \cdot 2 \cdot 192 = 1152$, where 3 is the maximum number of bit loading steps for the k th subcarrier and the r th sub-channel ($4\text{-QAM} \rightarrow 16\text{-QAM} \rightarrow 64\text{-QAM}$). For every assignment of two bits, the power cost has to be calculated across all subcarriers and sub-channels. For the adaptive transmitter design, it is recommended in [9] to use in practice the fast Lagrange bi-sectional search proposed in [28] that also provides an optimal solution with lower complexity.

The proposed *Greedy Power Allocation Algorithm* with the packet error ratio adaptation may be also extended to the multi-user scenario and used in the scheduler design. According to [11], it can cooperate with dynamic subcarrier assignment algorithm that has to be also modified to coded systems.

Abbreviations

AWGN	Additive White Gaussian Noise
BER	Bit Error Ratio
CSI	Channel State Information
DFT	Discrete Fourier Transform
FDD	Frequency Division Duplex
FEC	Forward Error Correction
FER	Frame Error Ratio
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
IDFT	Inverse Discrete Fourier Transform
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
LoS	Line of Sight
MAC	Medium Access Control
MIMO	Multiple Input Multiple Output
MRC	Maximum Ratio Combining
NLoS	Non Line of Sight
OFDM	Orthogonal Frequency Division Multiplexing
PAM	Pulse Amplitude Modulation
PER	Packet Error Ratio
QAM	Quadrature Amplitude Modulation
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
STC	Space-Time Coding
SVD	Singular Value Decomposition
TDD	Time Division Duplex
WiMAX	Worldwide Interoperability for Microwave Access

Bibliography

- [1] "Online Encyklopedia Wikipedia - Invention of radio." http://en.wikipedia.org/wiki/Invention_of_radio.
- [2] E. Dahlman, *3G Evolution: HSPA and LTE for Mobile Broadband*. Elsevier Academic Press, 2007.
- [3] A. Molisch, *Wireless Communications*. Wiley-IEEE Press, 2005.
- [4] A. Roca, "Implementation of a wimax simulator in simulink," Master's thesis, Technische Universität Wien, Institut für Nachrichtentechnik und Hochfrequenztechnik, February 2007.
- [5] L. S. Committee, "802.16 IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed Broadband Wireless Access Systems," *IEEE standards* (Oktober 2004).
- [6] C. Eklund, R. Marks, K. Stanwood, and S. Wang, "IEEE Standard 802.16: A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access," *Communications Magazine, IEEE*, vol. 40, no. 6, pp. 98–107, 2002.
- [7] "WiMAX Forum." <http://www.wimaxforum.org>.
- [8] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems: Modeling, Metodology and Techniques*. Kluwer Academic Publishers, 2002.
- [9] P. Xia, S. Zhou, and G. Giannakis, "Adaptive MIMO-OFDM based on partial channel state information," *IEEE transactions on signal processing*, vol. 52, no. 1, pp. 202–213, 2004.
- [10] G. Song and Y. Li, "Cross-layer optimization for OFDM wireless networks-part I: theoretical framework," *Wireless Communications, IEEE Transactions on*, vol. 4, no. 2, pp. 614–624, 2005.

- [11] G. Song and Y. Li, "Cross-layer optimization for OFDM wireless networks-part II: algorithm design," *Wireless Communications, IEEE Transactions on*, vol. 4, no. 2, pp. 625–634, 2005.
- [12] S. Zhou and G. Giannakis, "Adaptive Modulation for multiantenna transmissions with channel mean feedback," *Wireless Communications, IEEE Transactions on*, vol. 3, no. 5, pp. 1626–1636, 2004.
- [13] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge University Press, 2003.
- [14] A. Goldsmith and S. Chua, "Variable-rate variable-power MQAM for fading channels," *Communications, IEEE Transactions on*, vol. 45, no. 10, pp. 1218–1230, 1997.
- [15] M. Alouini and A. Goldsmith, "Adaptive Modulation over Nakagami Fading Channels," *Wireless Personal Communications*, vol. 13, no. 1, pp. 119–143, 2000.
- [16] A. Federgruen and H. Groenevelt, "The greedy procedure for resource allocation problems: Necessary and sufficient conditions for optimality," *Operations Research*, vol. 34, no. 6, pp. 909–918, 1986.
- [17] R. Manso, "Performance analysis of m-qam with viterbi soft-decision decoding," Master's thesis, Naval Postgraduate School, California, March 2003.
- [18] J. Proakis and M. Salehi, *Digital communications*. McGraw-Hill New York, 1995.
- [19] S. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1995.
- [20] A. Viterbi, "Convolutional codes and their performance in communication systems," *Communications, IEEE Transactions on [legacy, pre-1988]*, vol. 19, no. 5 Part 1, pp. 751–772, 1971.
- [21] D. Molteni, R. Bosisio, M. Nicoli, and D. Informazione, "Analytic framework for performance evaluation of multi-antenna WIMAX systems over fading channel," *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 3081–3084, 2008.
- [22] I. Onyszchuk, "Truncation length for Viterbi decoding," *Communications, IEEE Transactions on*, vol. 39, no. 7, pp. 1023–1026, 1991.

- [23] H. Bai and M. Atiquzzaman, "Error modeling schemes for fading channels in wireless communications: A survey," *IEEE Communications Surveys and Tutorials*, vol. 5, no. 2, pp. 2–9, 2003.
- [24] R. Khalili and K. Salamatian, "A new analytic approach to evaluation of packet error rate in wireless networks," *Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual*, pp. 333–338, 2005.
- [25] J. Anderson and S. Mohan, *Source and Channel Coding: An Algorithmic Approach*. Kluwer Academic Publishers Norwell, MA, USA, 1991.
- [26] T. M. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, New Jersey, 2006.
- [27] C. Mehlführer, S. Caban, and M. Rupp, "Experimental evaluation of adaptive modulation and coding in MIMO WiMAX with limited feedback," *EURASIP Journal on Advances in Signal Processing, Special Issue on MIMO Systems with Limited Feedback*, vol. 2008, Article ID 837102, 2008.
- [28] B. Krongold, K. Ramchandran, and D. Jones, "Computationally efficient optimal power allocation algorithm for multicarrier communication systems," in *Communications, 1998. ICC 98. Conference Record. 1998 IEEE International Conference on*, vol. 2, 1998.