

**Low-Density Hybrid-Check Coded
Superposition Mapping and its Application
in OFDM and MIMO**

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel

vorgelegt von
Zhenyu Shi

Kiel 2014

Tag der Einreichung: 11.11.2013

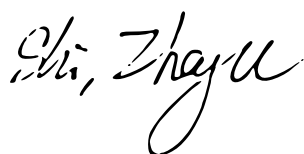
Tag der Disputation: 02.04.2014

Berichterstatter: Prof. Dr.-Ing. Peter Adam Höher

Prof. Dr.-Ing. Gerhard Bauch

Declaration

I declare that apart from the supervisor's guidance the content and design of the thesis is all my own work. The thesis has not been submitted either partially or wholly as part of a doctoral degree to another examining body and part of the thesis has been published. I also declare that the thesis has been prepared subject to the Rules of Good Scientific Practice of the German Research Foundation.

A handwritten signature in black ink, appearing to read 'Shi, Zhenyu'.

Zhenyu Shi

Kiel, October, 2013

Abstract

Since Shannon's landmark paper, many approaches have been proposed to achieve the channel capacity. In the low SNR regime, the problem has almost been solved by capacity achieving channel codes. The research on coded modulation in the high SNR regime is still under development. Among many methods in accomplishing this goal, superposition mapping is an elegant way as it does not require extra shaping to generate a Gaussian-like distributed signal. Superposition mapping has been shown to offer very close to capacity performance for the AWGN channel by combining with an irregular channel code. The aim of this thesis is to search for a code which provides stable performance for moderate sequence length and sufficient number of iterations, which is more suitable for implementation.

Concerning channel coding for superposition mapping, a generalized code design has recently been proposed. The so-called low-density hybrid-check (LDHC) coding intends to contrive coding and modulation in a joint way. The LDHC coding is constructed by integrating modulation into the Tanner graph. Thus, the complete code can be obtained by taking the effects of all the components into account. In this thesis, the LDHC code design is extended to OFDM and MIMO. For OFDM, the bit loading can be realized in the graph. In case of MIMO with spatial multiplexing, the code is extended to the spatial domain. In both cases, a suitable system structure will be proposed in this thesis. It will also be shown how this novel code design improves the system performance.

Keywords: Digital modulation, superposition mapping, bit-interleaved coded modulation (BICM), channel coding, low-density hybrid-check (LDHC) code, orthogonal frequency-division multiplexing, adaptive bit loading, multiple-input multiple-output (MIMO)

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Thesis Organization	5
2	Superposition Mapping	7
2.1	Channel Capacity	8
2.2	Mutual Information	9
2.3	Structure of SM	11
2.4	SM in Real Domain	12
2.4.1	SM-EPA	12
2.4.2	SM-GPA	13
2.5	SM in Complex Domain	15
2.5.1	PSM	15
2.6	Summary	18
3	LDHC Codes	21
3.1	BICM Structure	22

3.2	Soft-in Soft-out Detection Algorithms	23
3.2.1	Standard APP Algorithm	23
3.2.2	Tree-based APP Algorithm	24
3.2.3	Demapping Complexity	27
3.2.4	Performance Comparison	28
3.3	A Universal Structure	29
3.3.1	Matrix and Graph Representations	31
3.4	Code Design	34
3.4.1	Degree Distribution and Degree Allocation	34
3.5	EXIT Chart Analysis	36
3.5.1	DEM Curves	40
3.5.2	DEC Curves	42
3.6	Coded Modulation Curves	44
3.7	Numerical Results	49
3.7.1	Code Design by Degree Distribution	49
3.7.2	Code Design by Degree Allocation	50
3.7.3	Effects of Degree Allocations	51
3.7.4	Effects of Burst Length	53
3.7.5	Effects of Iterations	54
3.7.6	Coded Modulation Comparison	56
3.8	Summary	57

4	Superposition Mapping for OFDM	59
4.1	Theoretical Benefit	60
4.2	Water-filling Algorithm	61
4.2.1	Discrete Bit Loading	61
4.3	Mutual Information	64
4.4	LDHC Coded BICM-OFDM	68
4.4.1	Graph and Matrix Representations	70
4.4.2	Effects of Short Cycles	72
4.5	Simulation Results	74
4.5.1	Effects of Bit Loading	75
4.5.2	The Bit Allocation	76
4.5.3	Effects of Step Size	78
4.5.4	Effects of Degree Allocation	79
4.5.5	LDHC-PSM-OFDM vs. LDPC-QAM-OFDM	81
4.5.6	Bandwidth Efficiency	82
4.6	Summary	84
5	Superposition Mapping for MIMO	85
5.1	MIMO Transmission Model	86
5.2	MIMO Channel Capacity	88
5.2.1	Ergodic Capacity	89
5.2.2	Outage Capacity	98

5.3	LDHC-Code Based MIMO	100
5.3.1	Effects of Girth	104
5.4	Detection Algorithms	107
5.4.1	Gaussian Approximation Detection	108
5.4.2	MMSE Detection	112
5.4.3	Comparison	115
5.5	EXIT Chart Analysis	117
5.6	Numerical Results	118
5.6.1	Effects of Degree Allocation	118
5.6.2	Effects of Girth	120
5.6.3	Effects of Antenna Numbers	121
5.6.4	Performance Comparison	123
5.7	Summary	124
6	Conclusions	125
	Bibliography	127
A	List of Abbreviations	141
B	List of Symbols	143

Chapter 1

Introduction

1.1 Motivation

Channel capacity is the maximum mutual information between channel inputs and channel outputs [1]. The problem of finding efficient transmission schemes over Gaussian channels have driven many attentions over the past sixty years since Shannon's early work [2, 3]. To approach the capacity, different methods are suitable for different regimes. Basically, there are two regimes considered, the power-limited regime where the signal-to-noise ratio (SNR) is low, and the bandwidth-limited regime where the SNR is high. During the first few years, main focus has been put on the power-limited regime. Suitable channel codes have been designed for binary transmission [4]. Later on, noticeable progress has been obtained in the bandwidth-limited regime. In 1977, a multilevel coding (MLC) was proposed to approach the capacity [5, 6]. Another approach to the capacity by trellis-coded modulation (TCM) was introduced in 1982 [7]. Both multilevel coding and trellis coded modulation are based on set partition, and they become very popular in the late 1980's. In 1990's, there was a breakthrough in the power-limited regime with the appearance of Turbo codes [8] and the reinvention of LDPC codes [9, 10]. These channel codes operate very successfully in the low SNR regime. They have shown to be as close as just a few thousandths of decibels to the capacity with binary transmission, given long enough block lengths and a sufficient number of iterations [11]. Hence, a desirable

solution to the problem of achieving the channel capacity in the power-limited regime has almost been found out. However, the search for methods in the bandwidth-limited high SNR regime is still under development. In 1998, the proposal of bit-interleaved coded modulation (BICM) [12, 13] has drawn much excitement. The idea of this transmission is by inserting an interleaver between the encoder and the modulator, and performing iterative processing at the receiver side. This structure has delivered superior performance for higher-order modulations. Since then, BICM has become a pragmatic system. Of various higher-order modulations, the square quadrature amplitude modulation (QAM) and the phase shift keying (PSK) schemes have been long used as standard modulations. However, there is still a large gap from them to the channel capacity due to their equally distributed symbol constellations, especially for higher orders. For an additive white Gaussian noise (AWGN) channel, the capacity can be achieved only in the case that the channel output is zero-mean Gaussian distributed [2]. As the noise is Gaussian distributed, this assumption is equivalent to that the channel input is also a Gaussian variable. It is a common knowledge that the approach to the capacity in the bandwidth limited regime is by applying BICM with a certain shaping procedure [14].

In order to eliminate a gap between conventional uniform constellations and the channel capacity, the uniform signal needs to be shaped to a Gaussian constellation. By doing so, the so-called ultimate shaping gain, which is $\pi e/6$ or 1.53 dB, can be attained [15, 16]. In order to achieve this ultimate shaping gain, many researches have used different methods to generate a channel input with Gaussian distribution. One approach is by allocating different distributions to the signal sets, where the signal sets with low magnitude come with higher probabilities and the signal sets with high magnitude come with lower probabilities [17, 18, 19, 20, 21]. In [17, 18], different types of nonuniform signals were investigated. They have shown that a Maxwell Boltzmann distribution delivered the best results in minimizing the average power. Later, such a way of symbol distribution was shown to combine with the pragmatic Turbo coded BICM transmissions [20, 21]. The second technique is to extend the constellation to higher dimensions. These approaches are called signal shaping [22]. Among them are shell mapping [23, 24, 25, 26] and trellis shaping [27]. It has been shown that trellis shaping can be combined with multilevel cod-

ing [28]. Both shell mapping and trellis shaping are helpful in improving the performance with non-iterative processing. However, in BICM transmissions, the shaping procedure would bring significant extra efforts. The third approach is based on the central limited theorem. It is known that the summation of a large number of independent variables will be Gaussian distributed [29]. Thus, the extra active shaping is not necessary. Although no detailed structure was introduced in [29], it has laid a good foundation for the topic of superposition mapping.

In [30], Ma and Li Ping provided an analysis of superposition mapping. It has been shown that a direct superposition mapping delivers a superior performance. Besides, when combining capacity achieving codes, such as Turbo codes, a capacity achieving performance can be obtained. Nevertheless, the bandwidth efficiency is limited by 2 bits/symbol for the system. At the same time, Schoeneich and Hoeher proposed a multi-layer interleave-division multiple access (ML-IDMA) scheme [31, 32]. This scheme performs interleave-division multiplexing (IDM) for each user. Different from the scheme in [30], a different power allocation is assigned to each layer before superimposing and transmission. However, both of these two schemes are limited by a bandwidth efficiency of 2 bits/symbol. The key part of the IDM can also be served as a modulation scheme, which is the prototype of superposition mapping [33]. Later on, [34, 35, 36] provided theoretical analyses on why the equal power allocated superposition mapping was restricted by the 2 bits/symbol bandwidth efficiency. In [37], it has been found that conventional channel codes are not perfectly matched for superposition mapping due to its nonuniform constellations. Thus, a low-density hybrid-check (LDHC) code structure was proposed for the design of the coded modulation transmission. The LDHC code design is based on a factor graph [38] analysis. Using LDHC coded superposition mapping, a performance which is only 0.6 dB away from the capacity was reported in [36] for the AWGN channel.

In recent research on BICM with iterative decoding, the extrinsic information chart (EXIT) chart [39] has become an important method in finding a suitable coded modulation combination to approach the capacity. The issue of demodulator and decoder curve match is based on two factors, the threshold SNR and the bit error floor. Several techniques to reduce the SNR threshold has been proposed. In [40, 41], a technique has

been introduced by using label rules other than Gray-labeling so that simple codes can be combined. Later on, Fukawa et al. proposed an algorithm to optimize the labeling in [42]. In 2013, an M-APSK constellation has been proposed in [43]. In this thesis, we also focus on the SNR threshold. To completely eliminate the error floor, an accumulator [44] and modulation doping [45, 46] can be applied.

The above mentioned superposition mapping has the potential to approaching the channel capacity from a theoretical point of view. A superior performance has been reported in the AWGN channel given a extremely large number of block length and large number of iterations. However, in practical applications, the transmission block length is limited according to various reasons. The channels are more complicated than the AWGN channel. Thus, how to find a suitable coded modulation in Rayleigh fading multipath channels or channels with multiple antennas will be the main focus of this thesis.

In frequency-selective fading channels, orthogonal frequency-division multiplexing is applied to combat the multipath fading. OFDM converts a frequency-selective fading channel into a collection of frequency-flat fading subchannels so that the techniques that are appropriate for flat-fading channels can be implemented straight forward. In the early 1960s, OFDM has been applied in many different systems [47, 48, 49]. Until the 1990s, the performance and the application of OFDM in wireless communication have been comprehensive studied [50, 51, 52]. In this thesis, we apply LDHC coding to BICM-OFDM in frequency-selective fading channels for the optimization of the coded modulation transmission.

It has also been pointed out by Teletar [53] and Foschini [54, 55] that using multiple transmit antennas and multiple receive antennas would greatly improve the channel capacity. Afterwards, multiple-input and multiple-output (MIMO) has been widespread utilized and combined with many other techniques. Depending on how the MIMO channel capacity is exploited, MIMO systems can be separated into two categories. One is to transmit multiple data streams over multiple transmit antennas so as to boost the system throughput [54]. This technique is commonly called spatial multiplexing. The other scheme is to transmit different versions of one data stream over multiple transmit antennas so as to enhance the system power efficiency by exploring the spatial diver-

sity [56, 57]. From a theoretical point of view, the capacity of a MIMO channel can be achieved if the received symbols are all Gaussian distributed. This gives the motivation to replace the equally distributed signal constellation by a Gaussian or quasi Gaussian one. Besides, the formation of a received symbol is by summation of the transmit symbols with weighting factors. This resembles as the generation of a superposed symbol. Hence, it is of theoretical benefit to implement LDHC coding to BICM-MIMO with superposition mapping.

1.2 Thesis Organization

The remainder of the thesis content develops as follows:

Chapter 2 provides a fundamental introduction of superposition mapping. Different types of superposition mapping are described and compared.

Chapter 3 displays a structure of BICM with superposition mapping. A universal design, termed LDHC code, is applied for the joint design of coding and modulation. The coded modulation transmission is mainly investigated with moderate block length. Besides, a low-complexity detection algorithm has also been found for the demodulation of superposition mapped symbols.

Chapter 4 combines superposition mapping with BICM-OFDM. A “water pouring” method has been used to fully exploit the subchannels. Again, an LDHC code is implemented for the design of this kind of system.

Chapter 5 applies superposition mapping to BICM-MIMO transmission. The channel capacities for both fast-fading channels and quasi-static channels are analyzed. Different detection algorithms to mitigate the multi-antenna interference have also been introduced and compared. Finally, LDHC coding is also integrated into this transmission.

Chapter 6 summarizes the main results of the thesis.

Chapter 2

Superposition Mapping

According to Shannon's landmark derivation [1, 2, 58], the capacity of an additive white Gaussian noise (AWGN) channel can be approached by and only by a Gaussian distributed channel output. Due to the Gaussian distribution of the noise, the channel input is required to be Gaussian distributed as well. Since 1984, many approaches have been proposed to accomplish this goal. Superposition mapping (SM) is one of these approaches, which will be the main focus of this thesis. In many previous works [59, 60, 61, 62, 63, 64], superposition mapping has shown to be a modulation/multiplexing/multiple access technique to generate a Gaussian distributed signal without active signal shaping. The mapping from a certain number of bits to the transmit symbols is purely by a weighted summation of binary antipodal symbols so that the resulting constellation is not uniformly distributed. This kind of mapping sometimes results in a non-bijective mapping. Non-bijective mapping refers to that one symbol might be mapped from different bit sequences. Due to the non-bijectivity, superposition mapping is characterized by many different features than conventional modulation schemes. In this chapter, the principle of superposition mapping will be briefly introduced. Different types of superposition mapping, along with their characteristics, will be studied.

2.1 Channel Capacity

Consider an AWGN channel, which is given by the following equation:

$$y = x + z , \quad (2.1)$$

where y is the observation and x is the transmitted symbol. z is a zero mean Gaussian distributed random variable, i.e.,

$$z \sim \mathcal{CN}(0, \sigma_z^2) , \quad (2.2)$$

where σ_z^2 is the variance of the Gaussian distribution. Assume the signal energy E_s is defined and normalized as

$$E_s \doteq \mathbb{E}\{|x|^2\} = 1 , \quad (2.3)$$

and the noise power density is N_0 .

For 1-D modulation schemes, according to the notations in [65], we obtain

$$\sigma_z^2 = \frac{1}{2E_s/N_0} = \frac{N_0}{2E_s} , \quad (2.4)$$

According to Shannon, the capacity of the AWGN channel is obtained by [1, 58]

$$C = \max_{p(x)} I(x; y) , \quad (2.5)$$

where the mutual information between x and y , $I(x; y)$, is computed by

$$\begin{aligned} I(x; y) &= h(y) - h(y|x) \\ &= h(y) - h(y + z|x) \\ &= h(y) - h(z) . \end{aligned} \quad (2.6)$$

In this equation, $h(\cdot)$ denotes the differential entropy of a continuous random variable. It is known that the entropy of a zero mean Gaussian random variable with variance σ_z^2 can be calculated as

$$h(z) = \frac{1}{2} \log(\pi e \sigma_z^2) . \quad (2.7)$$

The mutual information is upper bounded by

$$I(x; y) \leq \frac{1}{2} \log(2\pi e(1 + \sigma_z^2)) - \frac{1}{2} \log(\pi e \sigma_z^2) = \frac{1}{2} \log \left(1 + \frac{2E_s}{N_0} \right) . \quad (2.8)$$

Equality holds if and only if y is Gaussian distributed. Hence, x also must be Gaussian distributed since the noise has been known as a Gaussian variable. If the above requirements are fulfilled, the capacity of an AWGN channel is given by

$$C = \frac{1}{2} \log \left(1 + \frac{2E_s}{N_0} \right) . \quad (2.9)$$

For 2-D modulation schemes, x , y , and z are all complex values. The definition of E_s in (2.3) still holds. Particularly, the noise term can be written as

$$z = z_r + jz_i . \quad (2.10)$$

The real part z_r and the imaginary part z_i are independent Gaussian random variables, with zero mean and the same variance $\sigma_z^2/2$:

$$z_r \sim \mathcal{CN} \left(0, \frac{\sigma_z^2}{2} \right) , \quad \text{and} \quad z_i \sim \mathcal{CN} \left(0, \frac{\sigma_z^2}{2} \right) . \quad (2.11)$$

In this case, σ_z^2 is given by

$$\sigma_z^2 = \frac{1}{E_s/N_0} = \frac{N_0}{E_s} . \quad (2.12)$$

Similarly, the channel capacity is

$$C = \log \left(1 + \frac{E_s}{N_0} \right) . \quad (2.13)$$

This capacity can also be approached by a Gaussian symbol distribution. To be more precise, both the real and imaginary part of the data symbols should be Gaussian distributed.

2.2 Mutual Information

The above mentioned channel capacity gives the maximum achievable rate of an AWGN channel with respect to the signal-to-noise ratio E_s/N_0 . However, in reality, the information bits are modulated before transmission. These transmitted symbols are restricted to a finite symbol alphabet, which is determined by the mapping schemes. To evaluate the maximum achievable rate of the mapping schemes, the mutual information (MI) between

x and y is applied. One commonly used MI calculation has been derived in [12]. In this method, the MI is computed as

$$\begin{aligned} I(x; y) &= h(x) - h(x|y) \\ &= N - \mathbb{E} \left\{ \log_{x,y} \frac{\sum_{x' \in \mathcal{X}} p(y|x')}{p(y|x)} \right\}. \end{aligned} \quad (2.14)$$

This method assumes an equally distributed channel input, so the entropy of x equals the number of bits per symbol N . \mathcal{X} is the symbol alphabet.

In the following, another method of MI computation will be introduced. The computation will be derived for 2-D modulation. Again, from the definition, we have

$$\begin{aligned} I(x; y) &= h(y) - h(y|x) \\ &= h(y) - h(z). \end{aligned} \quad (2.15)$$

In (2.15), the main goal is to compute $h(y)$, given $h(z) = \log(\pi e \sigma_z^2)$. Then, the entropy of a continuous variable is given by

$$h(y) = - \int p(y) \log p(y) dy, \quad (2.16)$$

where $p(y)$ is the probability density function (pdf) of the received symbol. From the equation of an AWGN channel, $p(y)$ can actually be computed. Given the distribution of the channel input $P(x)$ and the noise variance, we have

$$p(y) = \sum_{x \in \mathcal{X}} P(x) p(y|x) = \sum_{x \in \mathcal{X}} P(x) \frac{1}{\pi \sigma_z^2} \exp \left(-\frac{|y-x|^2}{\sigma_z^2} \right), \quad (2.17)$$

where \mathcal{X} is the channel input symbol alphabet and $P(x)$ is the probability mass function (pmf) of x . Since \mathcal{X} contains a finite number of symbols, it is possible to compute $h(Y)$ via computer programming. The case of a 1-D modulation can also be derived in a similar way. For 1-D modulation schemes, we obtain

$$h(z) = \frac{1}{2} \log(\pi e \sigma_z^2) \quad (2.18)$$

and

$$p(y) = \sum_{x \in \mathcal{X}} P(x) \frac{1}{\sqrt{2\pi\sigma_z^2}} \exp \left(-\frac{(y-x)^2}{2\sigma_z^2} \right). \quad (2.19)$$

2.3 Structure of SM

Fig. 2.1 provides an example of a superposition mapper with four bits per symbol. Unlike other modulation schemes, a superposition mapped symbol is formed by adding a certain number of binary symbols with a certain phase and amplitude allocation. The info bits b are first binary phase shift keying (BPSK) mapped. Then the BPSK mapped symbols are multiplied with a weighting factor α . This α is a complex value which contains the information of both amplitude and phase. In a general form, this mapping procedure can be written as

$$x = \sum_{n=0}^{N-1} \alpha_n d_n = \sum_{n=0}^{N-1} \alpha_n (1 - 2b_n), \quad b_n \in \{0, 1\}, \quad (2.20)$$

where x is the superimposed symbol, d_n denotes the n th symbol after BPSK mapping, with its corresponding amplitude and phase allocation represented by α_n . N is the number of bits per symbol. N can also refer to the bit load, or the modulation order.

Based on the selection of amplitudes and phases, superposition mapping can be classified into several categories. The weighting factor can be real-valued or complex-valued. Depending on whether α is real-valued or not, superposition mapping can be mainly separately into two classes. Without phase allocation, the resulting mapped symbols are in real domain. With phase allocation, the resulting symbols are in complex domain. Besides, superposition mapping can be further divided into more types inside each class depending on how one chooses the amplitude allocation. In the following, some typical classes will be of consideration.

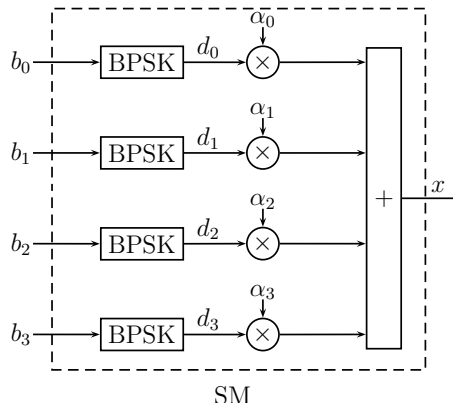


Figure 2.1: Block diagram of a superposition mapper with $N = 4$.

2.4 SM in Real Domain

2.4.1 SM-EPA

In this section, we consider the cases where all α_n in (2.20) are real values. If all α_n are equal, the resulting superposition mapping is called superposition mapping with equal power allocation (SM-EPA). Taking normalization into account, the resulting transmit symbol is given by

$$x = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} d_n, \quad d_n \in \{\pm 1\}. \quad (2.21)$$

Since an equal number of “+1”s and “−1”s will cancel each other out, the resulting constellation will be non-bijective. The mutual information (MI) curves of SM-EPA are

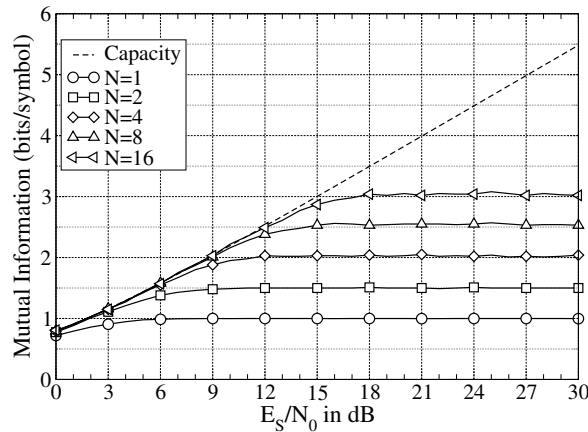


Figure 2.2: Mutual information of SM-EPA.

provided by Fig. 2.2, where the capacity curve follows (2.9). For each N , the MI curve is close to the capacity curve until it reaches its bandwidth efficiency limit. However, its entropy is far below 2^N . This is caused by its non-bijective property. The low entropy indeed limits its bandwidth efficiency. To increase the bandwidth efficiency, α_n can be unequal, such as the case of ASK. However, this will result in an equal distributed constellation, where the capacity-achieving advantage will be lost. In order to keep the Gaussian distribution while retaining the bandwidth efficiency, another way of selecting α_n comes forward. One possibility is the so-called superposition mapping with grouped power allocation (SM-GPA).

2.4.2 SM-GPA

In extreme cases where all the amplitudes are either equal or unequal, the resulting types of SM have their pros and cons. To find a mapping scheme that can take advantages of both situations, an amplitude allocation that lies in between these two cases would be appropriate. Implied by this, the amplitudes are divided into L groups. In each group, there are equal number of amplitudes. Assuming the number of amplitudes in each group is G , the overall number of bits per symbol is

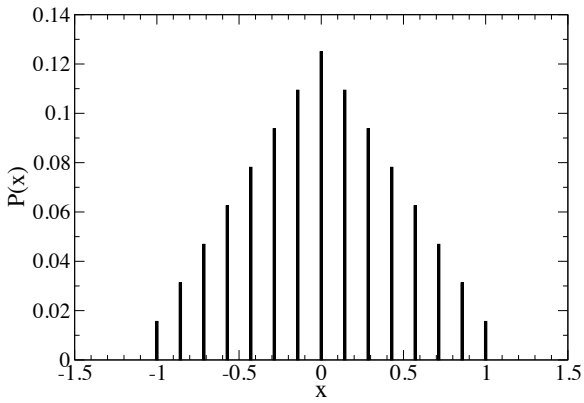
$$N = GL . \quad (2.22)$$

In each group, all G amplitudes have the same value while this value is different from group to group. Such a way of mapping remains certain equality for some amplitude values. As the equality is group-wise, this way of amplitude allocation is termed superposition mapping with grouped power allocation (SM-GPA). A symbol generated by SM-GPA is provided by

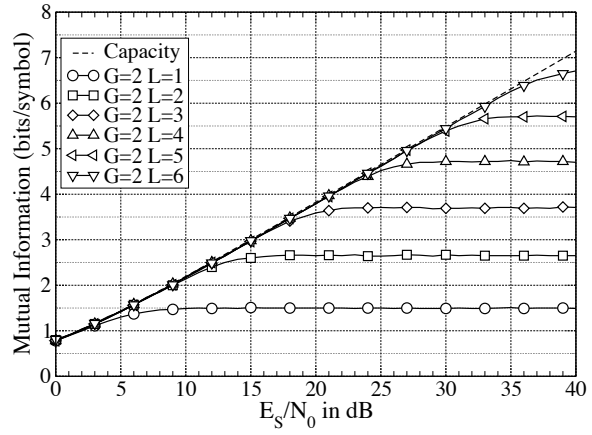
$$x = \alpha \sum_{l=0}^{L-1} 2^{-l} \sum_{g=0}^{G-1} d_{l,g} , \quad d_{l,g} \in \{\pm 1\} , \quad (2.23)$$

with α being the normalized weighting factor. The values between the adjacent groups are preferably varied by a factor of two. The amplitudes inside one group are the same, so the non-bijectionality still exists.

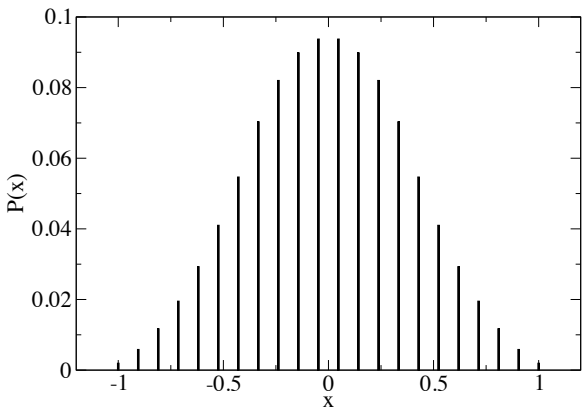
Fig. 3.4 provides the symbol distributions and the MI curves for SM-GPA. The group size is fixed to be two, and the number of levels are varied from $L = 1$ to $L = 6$. The envelope of SM-GPA in Fig. 3.4 is a triangular shape. Although not Gaussian, it is still a non-equal signal distribution. The MI curves are still very close to the capacity curve. When $G = 3$, the distribution is nearly Gaussian distributed. The capacity achieving property is also maintained. For both cases, the bandwidth efficiency is still lower than 2^N , but higher than in SM-EPA. From the MI analysis, SM-GPA still keeps its near Gaussian distribution by non-bijectionality. The unequal levels help to achieve a higher bandwidth efficiency. SM-GPA can provide a better trade-off between the constellation and the entropy. However, the group separation of bits demands that N must be an integer number of G . This adds extra efforts in choosing the modulation orders and combinations.



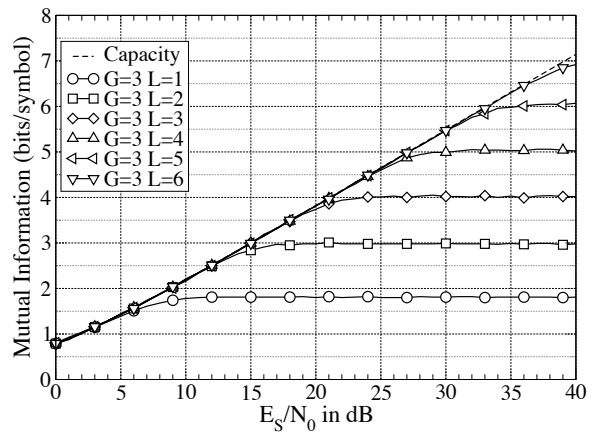
(a) Symbol distribution, $G = 2, L = 3$.



(b) Mutual information, $G = 2$.



(c) Symbol distribution, $G = 3, L = 3$.



(d) Mutual information, $G = 3$.

Figure 2.3: Symbol distribution and mutual information of SM-GPA.

2.5 SM in Complex Domain

In Section 2.4, two different types of SM have been introduced. Both of them are 1-D modulation schemes. However, in more sophisticated channels where the channel coefficients are complex-valued, 2-D modulation schemes are more widely applied. In this section, it will be shown that a Gaussian distributed constellation can also be achieved by superposition mapping in the complex domain.

2.5.1 PSM

From the previous discussions, it has been found that in order to obtain a trade-off between a desirable constellation and the bandwidth efficiency, the amplitude needs to be designed that it maintains both equalities and inequalities. If the amplitude allocation is still the same for each symbol, while the phase allocation is different from symbol to symbol, the resulting scheme is termed phase-shifted modulation (PSM). A PSM symbol is of the form

$$x = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{jn\pi/N} d_n, \quad d_n \in \{\pm 1\}. \quad (2.24)$$

The phase difference between two adjacent symbols are π/N . Fig. 2.4 provides three examples of PSM, along with their constellations and the symbol distributions. Since the constellation in either domain is symmetric, the symbol distribution in one domain would be enough to reflect its whole distribution, as is illustrated in Fig. 2.4. From the constellation diagrams, the mismatch exists between the constellation to the Gaussian distribution for each case. For $N = 4$, the constellation resembles two circles, with the points on the outer circle have a larger distance between each other than the points on the inner circle. For $N = 8$ and $N = 12$, the constellations are more like circles with different perimeters. Besides, at the boundary side, the constellation for $N = 12$ is not very regular. This mismatch to the Gaussian distribution is more obvious when observing the symbol distributions. It can be found that the distributions of PSM are not exactly Gaussian distributed. The symbol with the highest probability is not at $(0, 0)$, but close to it. Only in case of $N = 12$, the center point has one of the highest probability.

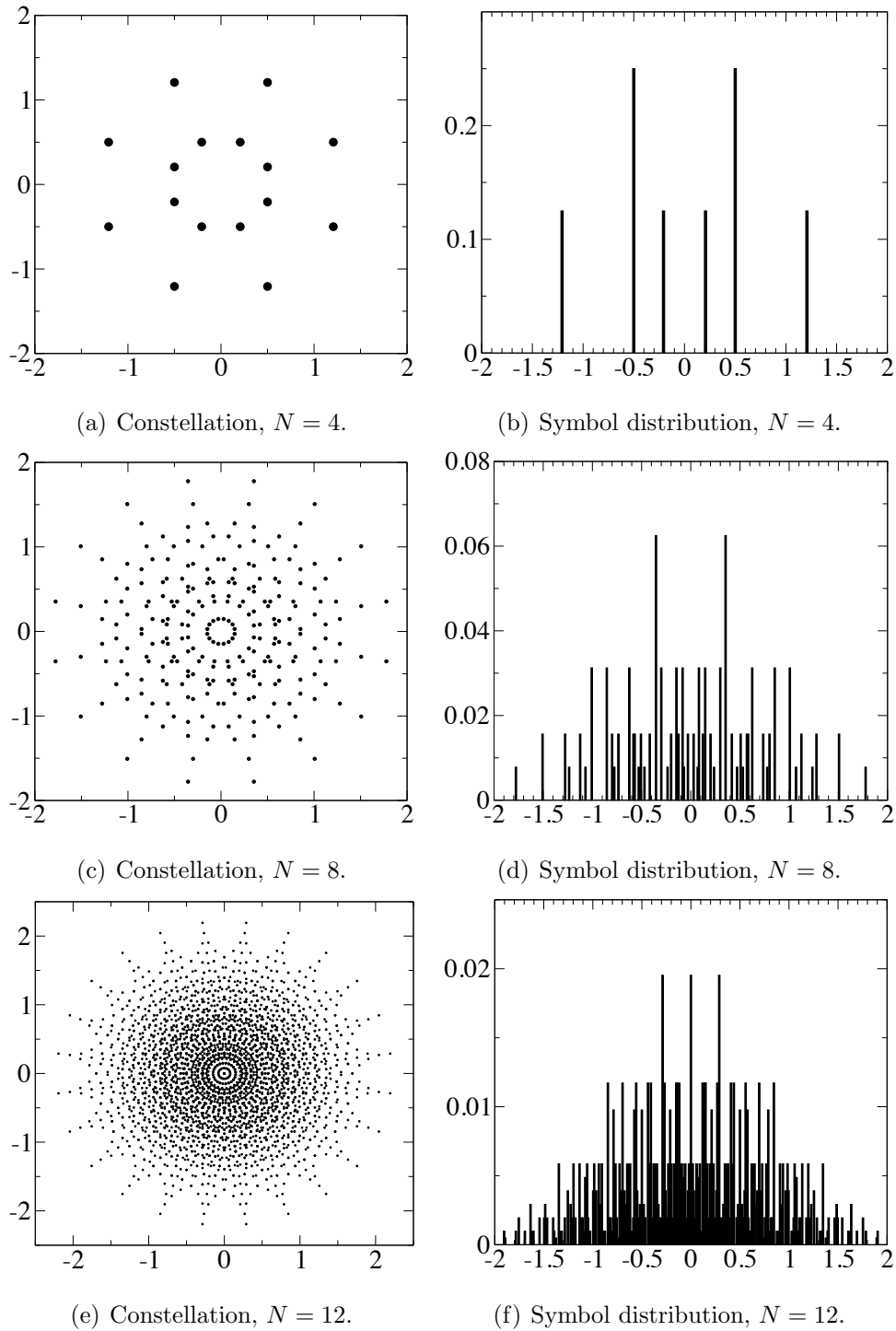


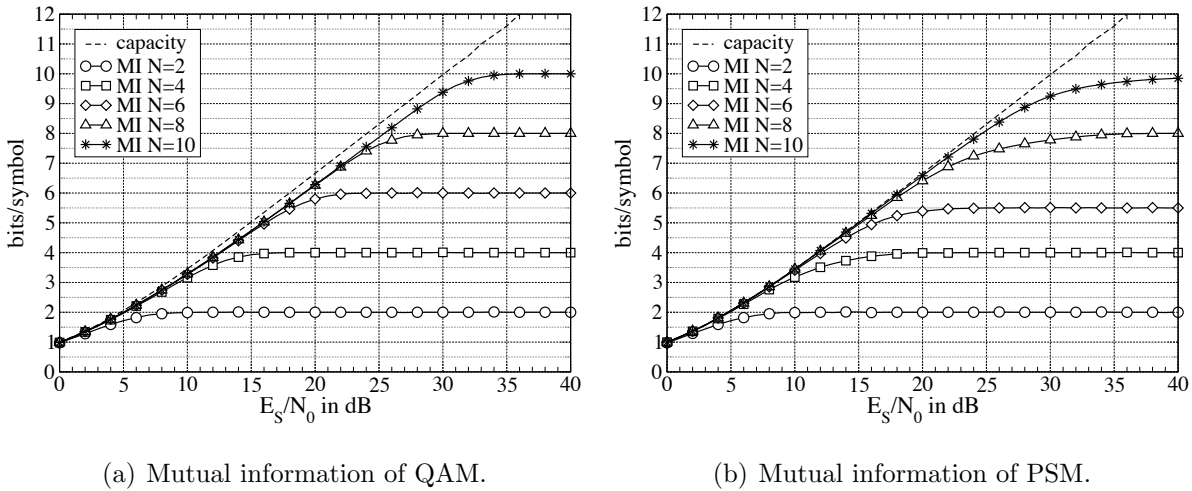
Figure 2.4: Symbol constellation and distribution of PSM.

N	2^N	Symbol alphabet
2	4	4
3	8	7
4	16	16
5	32	31
6	64	49
7	128	127
8	256	256
9	512	343
10	1024	961
11	2048	2047
12	4096	2401

Table 2.1: Cardinalities of PSM for different N .

Consider the bijectivity of SM. SM-EPA and SM-GPA are non-bijective for all N . However, this is not always the case for PSM. The maximum possible symbols and the actual symbol alphabets for PSM are listed in Table 2.1. The chosen modulation orders are from $N = 2$ to $N = 12$. As Table 2.1 shows, PSM shows non-bijective properties in most cases. However, at $N = 4$ and $N = 8$, they are both bijective. At some cases, such as $N = 3, 5, 7, 11$, the symbol sets are only one less than the maximum possible symbols. In these cases, two different sequences forms into “0”. However, in other cases, it is difficult to find the pattern of the relationship between N and the resulting alphabets.

Even though the constellation of PSM is not exactly Gaussian distributed, this way of unequal distribution is still able to approximately approach the channel capacity. Square quadrature amplitude modulation (QAM) has long been served as a standard modulation scheme in numerous applications. Square QAM delivers a uniformly distributed constellation with equal distance between neighboring symbols, which makes it a good candidate for comparison. Fig. 2.5 illustrates the mutual information curves for both PSM and (Gray-labeled) Square QAM with different N . The capacity curves in both of these figures follow (2.13). The MI curves are obtained from (2.15).

Figure 2.5: Mutual Information (MI) vs. E_s/N_0 .

In Fig. 2.5(a), a clear gap can be observed between the mutual information curves and the capacity curve. This gap, which is so-called ultimate shaping gain, is caused by the uniform symbol constellation. This gap comes from the modulation itself. Thus, it can not be compensated by channel coding unless some shaping methods are applied. However, this gap can be easily compensated by PSM, as illustrated in Fig. 2.5(b). Although the symbol constellations are not perfectly Gaussian distributed, the capacity curve can still be approached by PSM. The entropy is still lower than 2^N for some N due to the non-bijectivity, but the bandwidth efficiency is much higher than both SM-EPA and SM-GPA. In other words, PSM achieves a Gaussian-like signal by sacrificing minor bandwidth efficiency. In many practical systems employing OFDM and MIMO, PSM will be of main focus.

2.6 Summary

In this chapter, fundamentals of superposition mapping have been elaborated. Observed from the symbol constellations and the distribution graphs, superposition mapping can indeed deliver a quasi Gaussian distributed constellation. This way of modulation/multiplexing/multiple access can almost approach the channel capacity, verified by the mutual information curves. The effects of amplitude and phase allocation is also compared. To obtain a better trade-off between the Gaussian shape and the bandwidth

efficiency, the allocations should contain both equalities and inequalities. In the complex domain, PSM has been introduced and analyzed via mutual information and symbol distributions. In the following chapters, this kind of mapping scheme will be of major concern.

Chapter 3

LDHC Codes

In Chapter 2, we have introduced superposition mapping, along with its different types and their respective properties. It has also been shown that superposition mapping is able to approach the channel capacity due to its near Gaussian constellation. However, the Gaussian distribution also brings some problems. One is that it sometimes yields a non-bijective mapping. In uncoded cases, the non-bijective mapping can not guarantee a perfect transmission even under a noiseless channel. For this reason, it is mandatory to add channel coding to SM. Due to the non-bijectivity, the applied channel codes also need to be carefully designed.

In this chapter, general coded modulation transmission is considered. In order to match the coding and the modulation, we would like to design the overall transmission in a joint manner, which can be accomplished by the so-called low-density hybrid-check (LDHC) code [37, 35]. The structure of the LDHC code and its representations will be described in this chapter. With the help of an extrinsic information transfer (EXIT) chart [39] analysis, such a way of designing coded combination has delivered impressive performance in an AWGN channel [36]. Nevertheless, an extremely long sequence length is required. In this thesis, the block length is limited to a moderate length. The LDHC code method is used for searching channel codes with short to moderate block length. It is noted that the goal is to make the BER curve converge earlier instead of avoiding the error floor completely. To completely eliminate the error floor, other techniques such as an accumulator [44] or doping [45] can be applied.

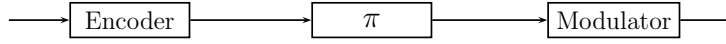


Figure 3.1: Block diagram of a BICM transmitter.

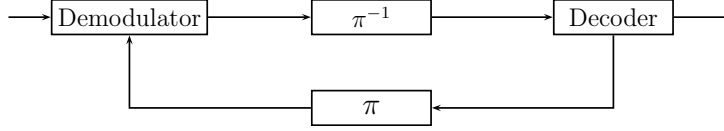


Figure 3.2: Block diagram of a BICM receiver.

3.1 BICM Structure

During the past decades, bit-interleaved coded modulation (BICM) has attracted much attention because of its high bandwidth efficiency and power efficiency [12]. Nowadays BICM is still considered as a pragmatic method. Fig. 3.1 and Fig. 3.2 provide block diagrams of the transmitter and the receiver structure of BICM, respectively, where π and π^{-1} denotes the interleaver and the deinterleaver. At the transmitter side, the info bits are first encoded to the code bits by adding redundancy. Then, a random interleaver π separates the encoder and the modulator in order to make the code bits mutually independent. In the modulator, the code bits are mapped onto complex-valued symbols to be transmitted. The mapping scheme can be Gray-labeled QAM, PSM, and other alternatives. At the receiver side, the joint demodulation and decoding is applied using the “Turbo principle” [66]. This way of detection is in an iterative manner. The input and output of each component are both soft information of the bits as log-likelihood ratio (LLR) values. Based on the channel observation and the a priori LLR values (L_a), the demodulator is able to compute the a posteriori probability (APP) for each code bit. Then the demodulator outputs the extrinsic information (L_e) by subtracting a priori information of the bit from the corresponding a posteriori LLR value. Afterwards, the extrinsic information is passed through the deinterleaver and served as the a priori information for the decoder. Then the extrinsic information is obtained by the decoder and is fed back to the demodulator.

3.2 Soft-in Soft-out Detection Algorithms

The main goal of the soft-input soft-out demapping algorithm is to compute the a posteriori LLR value of a certain bit b_n :

$$\text{LLR}(b_n) = \ln \frac{P(b_n = 0|y)}{P(b_n = 1|y)}, \quad (3.1)$$

where $P(b_n|y)$ is the a posteriori probability. Among many algorithms [67, 68, 69, 70] that try to compute the exact or the approximate a posteriori LLR value, it has been known that the standard APP algorithm always delivers an optimum performance with a relatively high computational complexity [71].

3.2.1 Standard APP Algorithm

From the relationship of the probabilities and the LLR values, given the a priori LLR value $\text{LLR}(b_n)$, its corresponding a priori probabilities are computed by

$$P(b_n = 0) = \frac{e^{\text{LLR}(b_n)}}{1 + e^{\text{LLR}(b_n)}}, \quad P(b_n = 1) = \frac{1}{1 + e^{\text{LLR}(b_n)}}. \quad (3.2)$$

The demodulator output is the extrinsic LLR, which is the a posteriori value minus the a priori value:

$$\begin{aligned} \text{LLR}_e(b_n) &= \ln \frac{P(b_n = 0|y)}{P(b_n = 1|y)} - \ln \frac{P(b_n = 0)}{P(b_n = 1)} \\ &= \ln \frac{P(b_n = 0|y)/P(b_n = 0)}{P(b_n = 1|y)/P(b_n = 1)} \\ &= \ln \frac{p(y|b_n = 0)}{p(y|b_n = 1)}. \end{aligned} \quad (3.3)$$

The a priori probabilities can be obtained via (3.2). Assume the bits are independent. According to the Bayer's rule, the a posteriori probability is of the form

$$P(b_n = 0|y) = \sum_{x \in \mathcal{X}_0} p(y|x)P(x) = \sum_{x \in \mathcal{X}_0} p(y|x) \prod_{i=0}^{N-1} P(b_i), \quad (3.4)$$

where \mathcal{X}_0 represents the symbol alphabet with $b_n = 0$. $P(b_i)$ is the a priori probability of b_i . Since the symbol alphabet is known to the receiver side, it is easy to calculate the transition probabilities $p(y|x)$ of each symbol belonging to \mathcal{X}_0 by

$$p(y|x) = \frac{1}{\pi\sigma_z^2} \exp\left(-\frac{|y-x|^2}{\sigma_z^2}\right). \quad (3.5)$$

Similarly, we have

$$P(b_n = 1|y) = \sum_{x \in \mathcal{X}_1} p(y|x) \prod_{i=0}^{N-1} P(b_i), \quad (3.6)$$

where \mathcal{X}_1 denotes the symbol alphabet with $b_n = 1$. By replacing (3.4) and (3.6) into (3.3), as a result, the output extrinsic LLR is given by

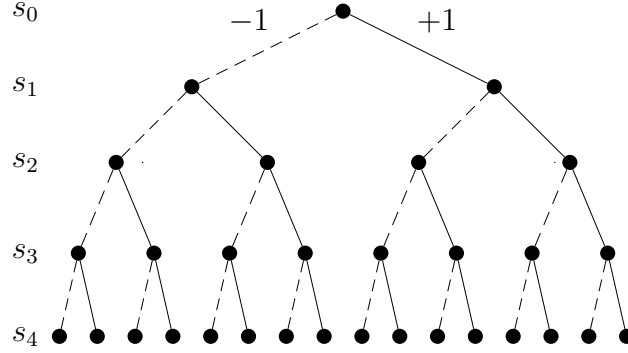
$$\text{LLR}_e(b_n) = \ln \frac{\sum_{x \in \mathcal{X}_0} p(y|x) \prod_{i=0, i \neq n}^{N-1} P(b_i)}{\sum_{x' \in \mathcal{X}_1} p(y|x') \prod_{i=0, i \neq n}^{N-1} P(b_i)}. \quad (3.7)$$

3.2.2 Tree-based APP Algorithm

Due to the summation property of superposition mapping, the soft-in soft-out demapping can be realized via a tree-based APP implementation. This algorithm has been initially proposed in [30] for superposition mapping with equal power allocation. This algorithm can be generalized for SM regardless of the phase and power allocation. As PSM will be the main concern of this thesis, the tree-based APP algorithm will be elaborated in the following as an example. Revisiting the general formula for PSM, the bits are BPSK mapped, multiplied with α_n , and summed up to generate a PSM symbol:

$$x = \sum_{n=0}^{N-1} \alpha_n d_n = \sum_{n=0}^{N-1} \alpha_n (1 - 2b_n), \quad d_n \in \{\pm 1\}. \quad (3.8)$$

According to the symbol alphabet and the number of bits per symbol, the tree can be constructed. Fig. 3.3 illustrates a tree growth based on PSM with $N = 4$. The tree starts from the root state s_0 . Since no binary symbols are added, $s_0 = 0$. When a new d_1 is added, the summation yields a new state s_1 . As d_n could be either +1 or -1, the current new state has two possibilities. In Fig. 3.3, a solid denotes that a “+1” is added and a dashed line denotes a “-1” is added. Then the tree continues to grow, with each node having two branches to the next level. The number of nodes at each state will always

Figure 3.3: Tree growth for PSM with $N = 4$.

be twice the number of the nodes at the previous state. The tree grows until the final symbol has been added. Finally, the last state shall contain 2^N nodes. A complete tree is depicted in Fig. 3.3.

This soft-in soft-out demapping algorithm operates in a recursive manner. The a priori probabilities can be obtained following (3.2), with $P(b_n)$ replaced by $P(d_n)$. Then the tree-based APP demapping proceeds as follows. From the tree growth, the probabilities of each node are obtained via two directions. Both results contribute to the output LLR values for each d_n .

In the forward propagation, the symbol probabilities are computed from s_0 to s_4 . For the root state, only one symbol exist, its probability is

$$P(s_0 = 0) = 1 . \quad (3.9)$$

Consider a certain level. When one symbol is added, the summation will yield a new symbol with two probabilities. The probability of the new symbol at the next state $P(s_{n+1})$ depends on both the probability of the current state $P(s_n)$ and the probability of the d_n . According to the Bayers' rule,

$$\begin{aligned} P(s_{n+1}) &= \sum_{s_n} P(s_{n+1}|s_n)P(s_n) \\ &= \sum_{s_n} P(s_n + \alpha_n d_n | s_n) P(s_n) \\ &= \sum_{s_n} P(d_n) P(s_n) . \end{aligned} \quad (3.10)$$

The last equation holds because d_n and s_n are independent and α_n is constant. Since each s_{n+1} is generated from only one s_n , (3.10) can be further simplified by

$$P(s_{n+1}) = P(d_n)P(s_n) . \quad (3.11)$$

Using (3.11), the symbol probabilities for each nodes can be obtained from s_0 to s_4 .

Then it comes to the backward propagation. The calculation starts from s_4 to s_0 . Consider the last state. Given the channel observation y , the transition probability of each symbol $p(y|s_4)$ can be computed by

$$p(y|s_4 = x) = \frac{1}{\pi\sigma_z^2} \exp\left(-\frac{|y-x|^2}{\sigma_z^2}\right) . \quad (3.12)$$

Unlike in the forward propagation, the transition probability of each node is computed in the backward propagation. To compute the transition probability of the previous state $p(y|s_{n-1})$, the transition probability of the current state $p(y|s_n)$ has to be known. From the tree, it can be observed that each s_{n-1} is connected with two s_n nodes. Then,

$$\begin{aligned} p(y|s_{n-1}) &= \sum_{s_n} p(y, s_n, s_{n-1})/P(s_{n-1}) \\ &= \sum_{s_n} p(y|s_n, s_{n-1})P(s_n, s_{n-1})/P(s_{n-1}) \\ &= \sum_{s_n} p(y|s_n)P(s_n|s_{n-1}) . \end{aligned} \quad (3.13)$$

The last equality holds because if s_n is given, s_{n-1} is also known. Since $P(s_n|s_{n-1}) = P(d_{n-1})$, further simplification gives

$$p(y|s_{n-1}) = \sum_{s_n} p(y|s_n)P(d_{n-1}) . \quad (3.14)$$

With (3.14), the transition probabilities for each state can be obtained. Now we have

both $P(s_n)$ and $p(y|s_n)$. Thus, $p(y|d_n)$ can be derived, also following the Bayers rule:

$$\begin{aligned}
p(y|d_n) &= \sum_{s_{n+1}} p(y, s_{n+1}|d_n) \\
&= \sum_{s_{n+1}} \frac{p(y, s_{n+1}, d_n)}{P(d_n)} \\
&= \sum_{s_{n+1}} \frac{p(y, s_{n+1}, s_{n+1} - s_n)}{P(s_{n+1}|s_n)} \\
&= \sum_{s_{n+1}} \frac{p(y, s_{n+1}, s_n)P(s_n)}{P(s_{n+1}, s_n)} \\
&= \sum_{s_{n+1}} p(y|s_{n+1}, s_n)P(s_n) \\
&= \sum_{s_{n+1}} p(y|s_{n+1})P(s_n) .
\end{aligned} \tag{3.15}$$

The last equality holds because if s_{n+1} is given, s_n is known. The extrinsic LLR of the n th bit is calculated by

$$\begin{aligned}
\text{LLR}_e(d_n) &= \ln \frac{p(y|d_n = +1)}{p(y|d_n = -1)} \\
&= \ln \frac{\sum_{s_{n+1} \in \mathcal{X}_{+1}} p(y|s_{n+1})P(s_n)}{\sum_{s_{n+1} \in \mathcal{X}_{-1}} p(y|s_{n+1})P(s_n)} ,
\end{aligned} \tag{3.16}$$

where \mathcal{X}_{+1} and \mathcal{X}_{-1} denote the symbol alphabet where $d_n = +1$ and $d_n = -1$, respectively.

This completes the tree-based APP algorithm for PSM.

3.2.3 Demapping Complexity

For the standard APP algorithm, the transition probabilities of all symbols in the alphabet are required to be calculated. Hence its complexity per bit is

$$\text{Demapping complexity of standard APP} \propto 2^N .$$

For the tree-based APP algorithm, the complexity needs to contain both the forward and backward propagation. For each direction, the complexity is affected by the number of nodes. For a modulation order N , there will be

$$2^1 + 2^2 + \dots + 2^N = 2^{N+1} - 2 \quad (3.17)$$

nodes in the tree. The same procedure is done for the backward computation, as the node number is the same. The overall complexity is the value obtained in (3.17) multiplied by two. The LLR values are calculated for the whole tree, which also represents one symbol. To evaluate the complexity for each bit, this value should also be divided by N . Hence, the complexity for a tree-based algorithm for each bit is:

$$\text{Demapping complexity of tree-based APP} \propto 2(2^{N+1} - 2)/N \approx 2^{N+2}/N .$$

It is not easy to directly compare these two formulas. In order to compare their complexities, the complexity of the standard APP is subtracted by the complexity of the tree-based APP, which gives

$$\begin{aligned} \mathcal{O}(\text{standard}) - \mathcal{O}(\text{tree-based}) &= 2^N - 2^{N+2}/N \\ &= 2^N(1 - 4/N) . \end{aligned} \quad (3.18)$$

Since N is a positive integral value, 2^N is positive. $1 - 4/N$ is always larger than zero in case $N \geq 4$. Hence, for higher-order modulation cases, the complexity of the tree-based APP algorithm is always lower than that of the standard APP demapping. For a clearer illustration, Table 3.1 provides the values of the complexities for the two concerned demapping algorithms with a range from $N = 2$ to $N = 10$.

The complexity of the tree-based APP demapping is strongly affected by the overall number of nodes. In some superposition mapping schemes such as SM-EPA and SM-GPA, the number of nodes at the n th state is in fact much lower than 2^n . In those cases, the complexity of the tree-based APP algorithm will be significantly reduced.

3.2.4 Performance Comparison

In this section, the two above mentioned demapping algorithms are compared in terms of performance. As a matter of fact, the tree-based APP demapping is valid for superposition

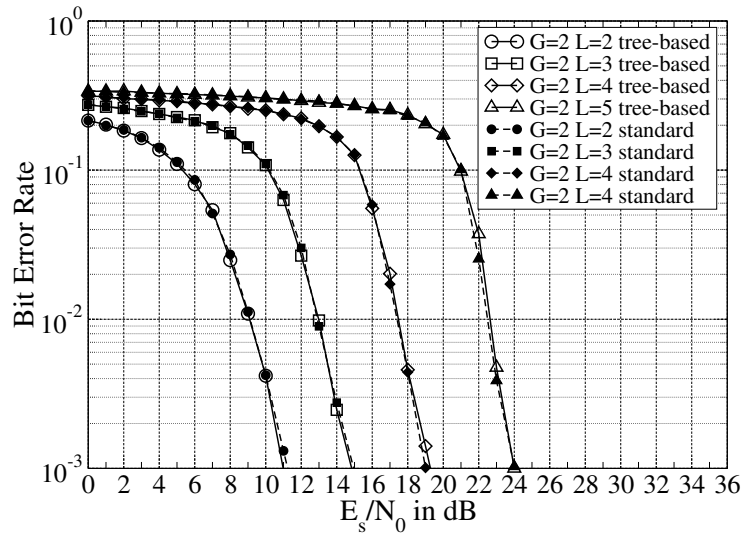
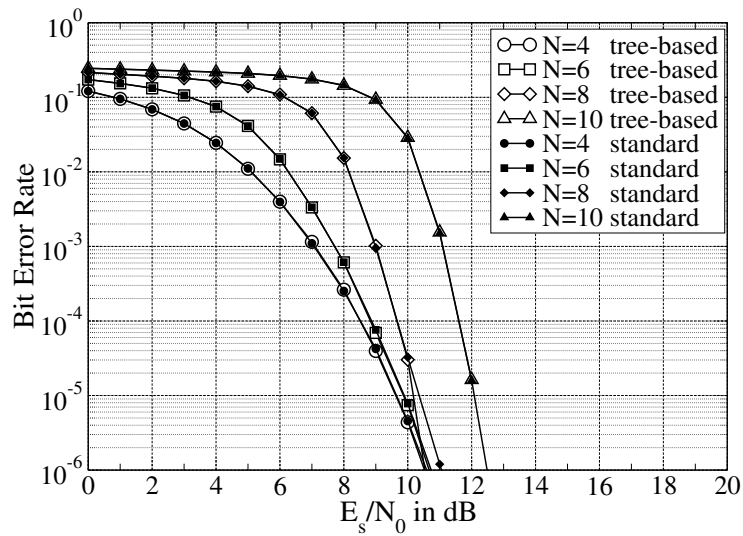
N	$\mathcal{O}(\text{standard})$	$\mathcal{O}(\text{tree-based})$
2	4	8
3	8	10.67
4	16	16
5	32	25.6
6	64	42.67
7	128	73.1
8	256	128
9	512	227.56
10	1024	409.6

Table 3.1: Comparisons of the complexities for different N .

mapping regardless of the power and phase allocation. The difference only lies in the construction of the tree. Here, we take both SM-GPA and PSM as examples. The info bits are encoded with a rate $R = 1/2$ repetition code. The number of transmit symbols is 100000. Different modulation orders N are provided herein. For SM-GPA, different combinations for G and L are also selected. As is shown in Fig. 3.4(a) and Fig. 3.4(b), there is no performance difference between the two demapping algorithms in all the cases. The tree-APP demapping algorithm can provide a lower computational complexity without performance penalty for higher-order modulations compared to standard APP demapping. Hence, the tree-based APP demapping algorithm will be utilized subsequently for the demodulation of a superposition mapped symbol.

3.3 A Universal Structure

Low-density parity-check (LDPC) codes have proved to deliver a capacity approaching performance in AWGN channels, as can be found in [11, 72, 73]. Square QAM schemes have long been served as standard modulation. In this chapter, a universal design for the coding and the modulation structure will be introduced. Such a hybrid design is termed the low-density hybrid-check (LDHC) coding.

(a) SM-GPA, $R = 1/2$.(b) PSM, $R = 1/2$.Figure 3.4: BER vs. E_s/N_0 for different demapping algorithms.

3.3.1 Matrix and Graph Representations

Implied by its name, LDHC codes share some similarities with LDPC codes. A Tanner graph describing an LDPC code is illustrated in Fig. 3.5(a). A circle denotes a variable node v , which represents a code bit after the LDPC encoder. A square box with + inside is a parity-check node p . The number of the connections to one node is called the degree of the node. In case of Fig. 3.5(a), the degree for the variable node (VN) is 2 and the degree for the parity-check node (PCN) is 3. For each parity-check node, all the variable nodes connected to it are constrained by a modulo-2 operation. The incidence matrix corresponding to this Tanner graph can be drawn, as in Fig. 3.5(b). The columns of the incidence matrix represent the variable nodes, and the rows of the incidence matrix represent the parity-check nodes. If there is a connection between the i th variable node to the j th parity-check node in the Tanner graph, there is a “1” at the j th row and the i th column of the matrix.

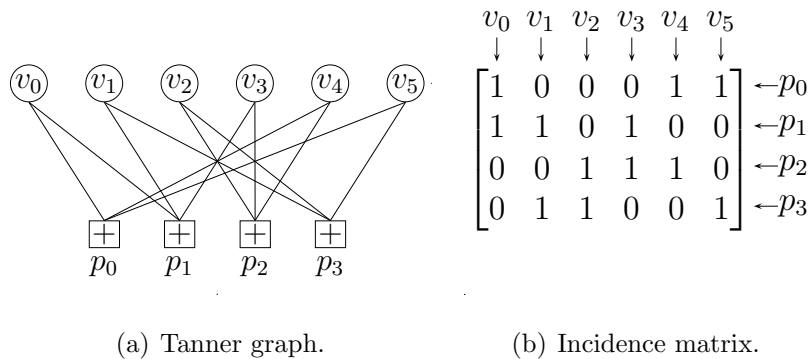


Figure 3.5: Tanner graph and incidence matrix of an LDPC code.

From the block diagram of the transmitter as is shown in Fig. 3.1, the code bits are mapped onto complex-valued symbols. To denote the mapping procedure, a so-called mapping node (MN) is introduced. For superposition mapping, these symbols are a weighted summation of the code bits. Thus, mapping nodes are sometimes called summation-check nodes for superposition mapping. The mapping node is constrained by the bits mapped to this symbol. Thus, the Tanner graph of such a structure is illustrated in Fig. 3.6(a), where a circle with s denotes a mapping node.

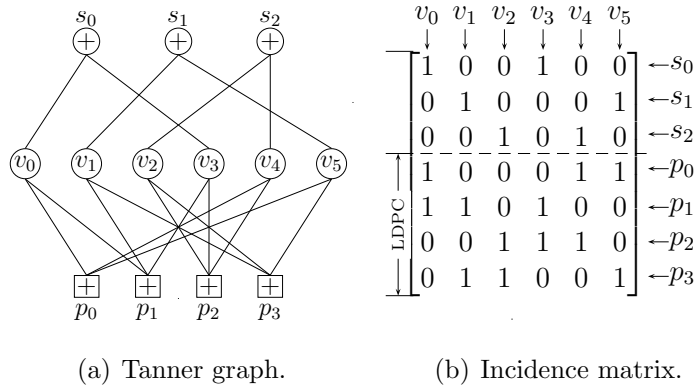


Figure 3.6: Tanner graph and incidence matrix of an LDHC code.

The lower part of the graph is the same as for LDPC codes. We focus more on the upper part. If the bits are mapped to one symbol, the corresponding variable nodes are all connected to the corresponding mapping node. The degree of the mapping node equals the modulation order N . Similarly, the incidence matrix of the LDHC code is shown in Fig. 3.6(b). The incidence matrix is separated into two submatrices by a dashed line in Fig. 3.6(b). The columns once again represent the variable nodes. As the lower part follows the same definition as for LDPC codes, the rows in the lower submatrix represent the parity-check nodes. The rows in the upper submatrix represent the mapping nodes, with their row weights equal to the degrees of the mapping nodes. For a constant modulation transmission, the degrees of the mapping nodes are all equal, meaning all the complex-valued symbols have the same modulation order.

After introducing the mapping nodes, the degrees of the variable nodes will be connected to two directions: One is to the mapping nodes and the other is to the parity-check nodes. Reflecting in the incidence matrix, the column weight is the overall degree of each variable node. For each column, the number of “1”s in the upper submatrix corresponds to the degree of the variable nodes with respect to (w.r.t) the mapping nodes and the number of “1”s in the lower submatrix corresponds to the degree of the variable nodes w.r.t the parity-check nodes. In Fig. 3.6(a), the degree w.r.t. the mapping nodes is one. This means that each code bit belongs to one symbol. If this degree is larger than one, as the Tanner graph illustrated in Fig. 3.7(a). In this case, this degree is two. The corresponding incidence matrix is shown in Fig. 3.7(b), where the column weight in the upper submatrix is also two.

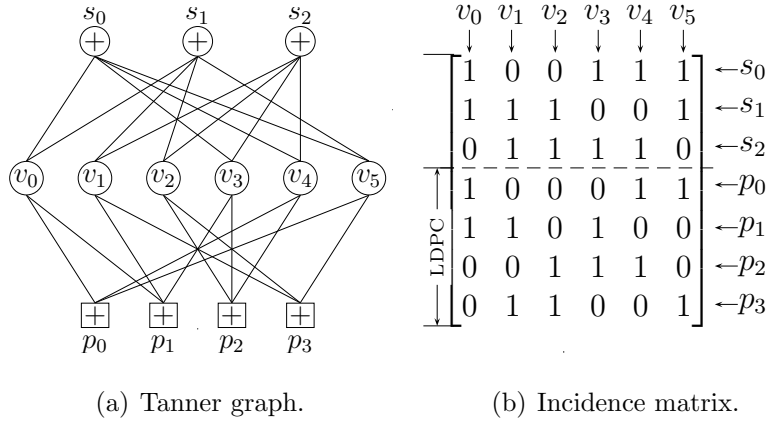


Figure 3.7: Tanner graph and incidence matrix of an LDHC code.

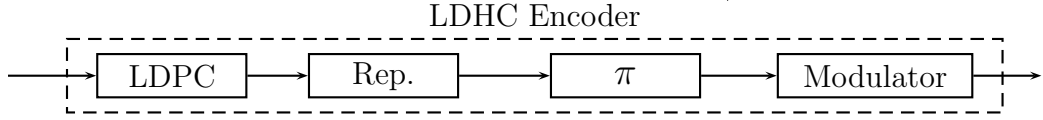


Figure 3.8: A BICM transmitter with a concatenated code.

It means that the code bits are repeated twice before mapping to a transmit symbol. Such a step equals operating a rate 1/2 repetition code. If seen from a BICM transmission point of view, this Tanner graph functions the same as adding a repetition encoder in between the LDPC encoder and the interleaver, as provided in Fig. 3.8. To make things clear, after the LDPC encoder, the sequence length is equivalent to the number of variable nodes. After the repetition encoder, the sequence length is tantamount to the number of connections between the variable nodes and the mapping nodes. The interleaver does not change the length. Finally, after the modulator, the sequence length is the same as the number of the mapping nodes.

It can be found that the Tanner graph of the LDHC code can truly reflect a BICM transmission where all the components are included. The connection between mapping nodes and variable nodes represent the modulation. How the variable nodes and the mapping nodes are connected is determined by the interleaver. The degree of the variable nodes shows the repetition code, if it exists. Then the connections between variable nodes and the parity-check nodes are for an LDPC code. How these two nodes are connected depends on the code design. Using such a general design, the optimization of the whole system can be performed in a joint way where the effects of each component related.

3.4 Code Design

Nowadays, many parity-check based code, such as convolutional codes, Turbo codes, and LDPC codes have been optimized extensively. Numerous research has also shown that these codes by far fit well for Gray-labeled square QAM [74, 75]. However, whether these parity-check-based codes fit for the superposition mapping lacks sufficient evidence. Due to the different properties between PSM and QAM, the channel codes for PSM might be quite different. In order to search for a channel code fits for PSM, the above mentioned LDHC code is applied herein.

3.4.1 Degree Distribution and Degree Allocation

Revisiting Fig. 3.7, the Tanner graph is composed of three different nodes. If the degree of each node is constant, the Tanner graph represents a regular LDHC code. If the degrees are different, there will be an irregular LDHC code. To denote the irregularity, the degree distribution polynomials are used. Here are the notations of the degree distribution polynomials for each kind of nodes. Starting from the bottom in Fig. 3.7, the parity-check node degree distribution polynomial is denoted by

$$\rho(X) = \sum_{d=1}^{D_p} \rho_d X^{d-1}, \quad (3.19)$$

where ρ_d denotes the fraction of all edges connected to the parity-check nodes with degree d , and D_p is the maximum parity-check node degree. Likewise, the mapping node degree distribution polynomial is given by

$$\zeta(X) = \sum_{d=1}^{D_s} \zeta_d X^{d-1}, \quad (3.20)$$

where ζ_d denotes the fraction of all edges connected to the mapping nodes with degree d , and D_s is the maximum mapping node degree. However, the situation for the variable nodes are a little bit complicated as the connections to the variable nodes are from two directions: the mapping nodes and the parity-check nodes. Hence, we define both

polynomials as follows:

$$\begin{aligned}\lambda_p(X) &= \sum_{d=1}^{D_{vp}} \lambda_{dp} X^{d-1} , \\ \lambda_s(X) &= \sum_{d=1}^{D_{vs}} \lambda_{ds} X^{d-1} .\end{aligned}\tag{3.21}$$

λ_p and λ_s are the fractions of all edges connected to the parity-check nodes and the mapping nodes, respectively. For instance, the degree distribution polynomials of the Tanner graph in Fig. 3.7 are as follows:

$$\begin{aligned}\rho(X) &= X^2 , & \zeta(X) &= X^3 , \\ \lambda_p(X) &= X , & \lambda_s(X) &= X .\end{aligned}\tag{3.22}$$

The values of the degrees are also important factors to determine the code rate of channel codes. For a regular code, the code rate is directly calculated using the degrees of the nodes. For an irregular code, there are different degrees for every type of nodes. Thus, it is necessary to average the degrees for different kinds of nodes and use these values to compute the code rate. To compute the code rate of an irregular LDHC code, we start from the lower part, which is also the LDPC code part. The code rate of a regular LDPC code is computed by

$$R_p = 1 - \frac{d_p}{d_{vp}} ,\tag{3.23}$$

with d_p being the degree of parity-check nodes and d_{vp} being the degree of variable nodes w.r.t parity-check nodes. The code rate of an irregular LDPC code R_p is obtained by

$$R_p = 1 - \frac{\sum_{d=1}^{D_p} \rho_d d}{\sum_{d=1}^{D_{vp}} \lambda_{dp} d} .\tag{3.24}$$

Consider the mapping node part. If the degree of mapping nodes is the same value d_s , the code rate is

$$R_s = \frac{1}{d_s} .\tag{3.25}$$

Similarly, if the degrees are not constant, the code rate is given by

$$R_s = \frac{1}{\sum_{d=1}^{D_{vs}} \lambda_{ds} d} .\tag{3.26}$$

The degree of a mapping node represents the modulation order of the mapping symbol. The LDHC codes provide a degree of freedom of choosing between a constant modulation scheme and a variable modulation scheme. If the degrees are not constant, we will have a variable modulation scheme, with its average modulation order

$$\bar{N} = \sum_{d=1}^{D_s} \zeta_d d. \quad (3.27)$$

The variable modulation transmission will not be a scope of this chapter, but will be elaborated in Chapter 4. Combining the above mentioned three parts, the overall bandwidth (BW) efficiency of the transmission system is

$$BW = \bar{N} R_s R_p. \quad (3.28)$$

Given a fixed bandwidth efficiency, a fixed code rate $R_s R_p$ is expected. There are two ways for the adaptation of the code properties. One is by changing the degree distribution polynomials. The other way is by determining the proportions between R_s and R_p . In the matrix representation, this can be controlled by allocating the proportions of “1”s between the upper submatrix and the lower submatrix. Such a way of adaptation is termed “degree allocation”. To find a channel code fit for superposition mapping, the analysis of both the “degree distribution” and the “degree allocation” will be considered.

3.5 EXIT Chart Analysis

To search for a suitable match between the coding and modulation, one way is by exhaustive search among various degree distribution combinations of codes. This procedure can be performed via large amounts of Monte Carlo simulations. However, the extremely long sequence length and a large number of iterations make such a method not very efficient. To speed up the search, several analysis tools have been proposed. The most commonly used analysis tools are the density evolution algorithm [72] and the EXtrinsic-Information-Transfer (EXIT)chart [39]. Density evolution has been proposed to predict the threshold for an irregular LDPC code. In case of AWGN channel, given a channel parameter (usually the noise variance), the update of LLR values between the variable

nodes and the parity-check nodes over iterations can be computed. Instead of the actual values, the density evolution algorithm only computes how the probability density function (pdf) of these LLR values are updated. Finally, the resulting bit error probability is obtained via the integration of the resulting pdfs. Applying this method, one can expect the noise threshold for a code with a fixed degree distribution polynomial by gradually increasing the noise parameter until a preset bit error probability is reached. Later on, a simplified version of this algorithm has been proposed in which the pdfs of the LLR values are approximated by mixture Gaussian variables [11]. This density evolution method, combining with a differential evolution algorithm [76], is able to find degree distribution polynomials which fulfill the required conditions. In this section, the other evaluation method, EXIT chart analysis, is of major consideration. Unlike the density evolution method, the EXIT chart analysis provides a graphical illustration of how the qualities of the LLR values are updated during iterations.

In the EXIT chart analysis, the mutual information between the LLR values and its corresponding code bits are considered. According to its definition, the mutual information evaluates the average quality of the LLR values with respect to the code bits. If the mutual information is zero, the LLR values and the code bits are considered irrelevant. If the mutual information is one, the LLR values and the code bits are very much related. In other words, the original information of the code bits can be perfectly reconstructed by the LLR values. Hence, the transfer curve can truly reflect the improvement of the output LLR values quality by manually setting the mutual information input. In a coded modulation transmission, transfer curves of the demodulator and the decoder are considered. Since the interleaver and the deinterleaver only shift the order of the code bits, the information those LLR values contained will not be changed after the interleaver and the

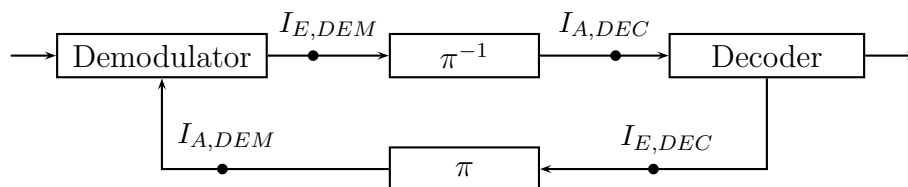


Figure 3.9: Mutual information for the demodulator and the decoder.

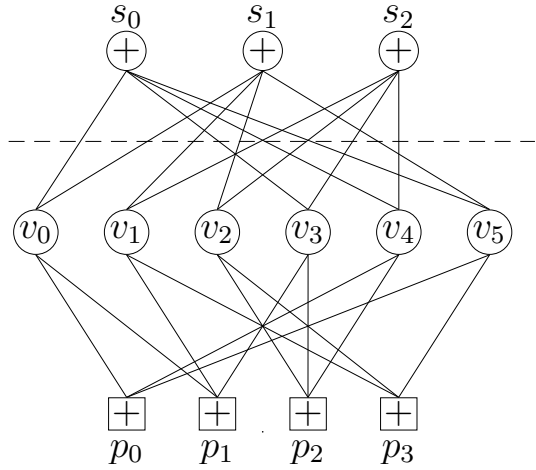


Figure 3.10: Tanner graph for an LDHC code.

deinterleaver. Following the definitions in Fig. 3.9, the mutual information at the output of the demodulator is served as the input for the decoder after the deinterleaver, and the decoder feedback is served as the input for the demodulator after the interleaver feedback. The following relationships hold true:

$$I_{E,DEM} = I_{A,DEC} , \quad I_{E,DEC} = I_{A,DEM} . \quad (3.29)$$

In the Tanner graph representation, as shown in Fig. 3.10, the mutual information curves are drawn between the dashed line. The upper part refers to the demodulator and the lower part refers to the decoder.

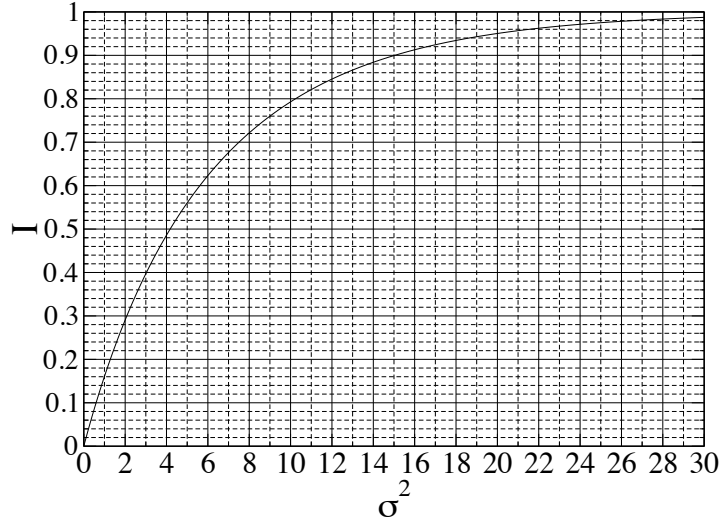
To obtain the mutual information, the LLR values are assumed to be a two-peak Gaussian distributed variable, which can be written as

$$\text{LLR}(d_k) = \frac{\sigma^2}{2} d_k + n_k , \quad (3.30)$$

where $d_k \in \{\pm 1\}$, and n_k is a Gaussian variable with zero mean and variance of σ^2 . Hence, the LLR values are a symmetric two-peak Gaussian variable with

$$\text{LLR} \sim \mathcal{N} \left(\pm \frac{\sigma^2}{2}, \sigma^2 \right) . \quad (3.31)$$

The advantage of this assumption is that the mean and variance of the Gaussian variable can be represented by only one parameter σ^2 . The goal is to use only one parameter to

Figure 3.11: I vs. σ^2 .

relate the mutual information. According to the mutual information definition, we have mutual information I computed as

$$\begin{aligned}
 I &= H(d) - H(d|\text{LLR}) \\
 &= 1 - \mathbb{E} \left\{ \log_2 \left(\frac{1}{p(d|\text{LLR})} \right) \right\} \\
 &= 1 - \sum_{d=\pm 1} \frac{1}{2} \int_{-\infty}^{+\infty} p(l|d) \log_2 \left(\frac{p(l|+1) + p(l|-1)}{p(l|d)} \right) dl . \quad (3.32)
 \end{aligned}$$

Following the Gaussian approximation, after a few derivations, the mutual information is of the form

$$I = 1 - \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{(l - \frac{\sigma^2}{2})^2}{2\sigma^2} \right) \log_2(1 + \exp(-l)) dl . \quad (3.33)$$

From the above equation, the mutual information can be treated as a function of σ^2 . However, the calculation of this equation is difficult as lots of integration and exponential operations are required. Thus, numerical simulations are conducted to obtain the relationship between I and σ , as illustrated in Fig. 3.11. Observed from Fig. 3.11, I increases with σ^2 increases. Since σ should always be positive, I is monotonic in σ . Hence, its inverse function exists. To relate I to σ^2 , $J(\cdot)$ and $J^{-1}(\cdot)$ are defined as

$$I = J(\sigma) , \quad \sigma = J^{-1}(I) . \quad (3.34)$$

Some approximation functions have been found to approximate this J function, as has been provided in [77]. Upon obtaining the functions of $J(\cdot)$ and $J^{-1}(\cdot)$, the mutual information curves for the demodulator (DEM) and the decoder (DEC) can be drawn, respectively.

3.5.1 DEM Curves

For any demapping algorithm, the computed LLR values from the demodulator are influenced by two parameters: the SNR value and the feedback a priori LLR values. Thus, the demodulator mutual information output $I_{E,DEM}$ can be seen as a function with both $I_{A,DEM}$ and SNR as the inputs:

$$I_{E,DEM} = f(I_{A,DEM}, \text{SNR}) . \quad (3.35)$$

In case of BPSK, σ is of the form

$$\sigma^2 = 4/\sigma_\omega^2 = 4E_s/N_0 = 4 \cdot \text{SNR} . \quad (3.36)$$

However, in higher-order modulations, it is not straightforward to get its output mutual information via equations. Besides, the LLR values will also be different if a different demapping algorithm is applied. Hence, the mutual information computation works as follows. By manually setting $I_{A,DEM}$, its corresponding σ_A is obtained via J^{-1} function. Based on the Gaussian assumption, LLR values with distribution $(\pm \frac{\sigma_A^2}{2}, \sigma_A^2)$ are randomly generated. These LLR values are used as a priori information to the demodulator. Hence, Monte Carlo simulations are carried out to compute the output LLR values. Given these output LLR values, the mutual information can be obtained following the mutual information definition mentioned in (3.32). By setting $I_{A,DEM}$ from 0 to 1, the corresponding $I_{E,DEM}$ is obtained accordingly. Hence, the curve can be drawn given a fixed SNR. Furthermore, there is a reduced complexity way of computing the mutual information given the LLR values is shown in [78]. This method has delivered the same transfer curve as from (3.32).

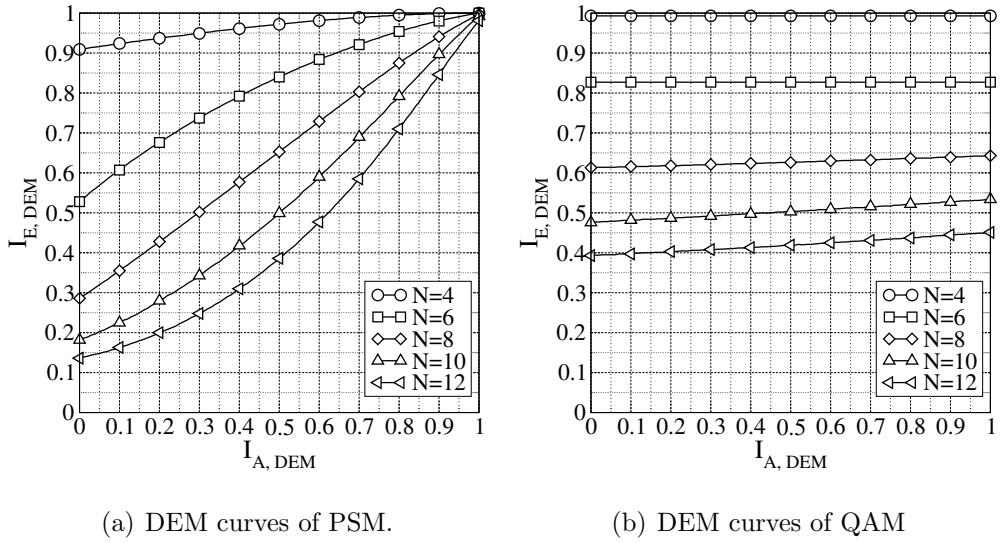
Figure 3.12: EXIT charts for PSM and QAM, $E_s/N_0 = 16$ dB.

Fig. 3.12 provides an example for the DEM transfer curves. The transfer function is compared for two modulation schemes, natural-labeled PSM and Gray-labeled square QAM. For both schemes, SNR is fixed to be 16 dB. The modulation orders are chosen from $N = 4$ to $N = 12$ at a step of 2. The transfer curves for PSM and QAM are illustrated in Fig. 3.12(a) and Fig. 3.12(b), respectively. Observed from the figures, the transfer curves for QAM are relatively flat for $I_{A,DEM}$ from 0 to 1. This indicates that QAM is not very sensitive to the input a priori information. Hence, iterative processing will not be helpful for the improvement of the LLR qualities. The other problem is that $I_{E,DEM}$ will not reach 1 even if the input LLR values are extremely good for higher-order modulations. At the same SNR, the ending point of the curve is dropping when N is increasing. As for PSM, the good point is the transfer curves are able to stop at (1, 1). However, every coin has two sides. The starting part is relatively low, especially for large N . The quality of the LLR values has a strong effect on the correctness of the detection. Thus, iterative processing is necessary to guarantee error-free detection. As the difference exists between the curves of two modulation schemes, the commonly known channel codes, which match QAM will no longer be suitable for PSM. Hence, a code design for PSM is necessary to match for this kind of “DEM” curves.

3.5.2 DEC Curves

Unlike “DEM” curves which are obtained via simulations, the “DEC” curves can be drawn directly using equations to be derived in this section. Observed from Fig. 3.10, the lower part of the Tanner graph is composed of the variable nodes and the parity-check nodes. The message update rules are predictable for each node. Based on the computation of the pdfs of LLR values, the mutual information update can be obtained accordingly. The LLR values are updated in the following orders: After receiving the messages from the mapping nodes, the messages are computed at the variable nodes and passed to the parity-check nodes. Afterwards, the messages are exchanged at the parity-check nodes, and passed back to the variable nodes. All the incoming LLR values at the variable nodes are again updated. Now there exist two possible iterations. One is that the updated LLR values can be transferred back to the parity-check nodes. We name such an iteration as the local iteration. The other possibility is that the updated LLR values can also be transported to the mapping nodes, which starts a so-called global iteration between the demodulator and the decoder. The mutual information update also follows the same order.

We start with a regular code where all the nodes have equal degrees. A certain value of mutual information $I_{A,DEC}$ is given as an input. According to the assumption, the LLR values from the mapping nodes to the variable nodes can all be treated as Gaussian variables with $\mathcal{N}\left(\pm\frac{\sigma_A^2}{2}, \sigma_A^2\right)$, where σ_A is computed by

$$\sigma_A = J^{-1}(I_{A,DEC}) . \quad (3.37)$$

As the output of LLR value of a message path at the variable nodes are purely the summation of messages from the rest incoming paths. The LLR values from the VNs to the PCNs are

$$L_{v \rightarrow p} = d_{vs}L_{s \rightarrow v} + (d_{vp} - 1)L_{p \rightarrow v} , \quad (3.38)$$

where d_{vs} denotes the degree of the VNs w.r.t. the MCNs and d_{vp} denotes the degree of the VNs w.r.t. the PCNs. According to the Gaussian approximation of LLR values, the summation of a number of Gaussian variables yields another Gaussian variable, with its mean equals the summation of the means of each components. Hence, we have

$$\sigma_{v \rightarrow p}^2 = d_{vs}\sigma_{s \rightarrow v}^2 + (d_{vp} - 1)\sigma_{p \rightarrow v}^2 , \quad (3.39)$$

with $\sigma_{s \rightarrow v}^2 = \sigma_A^2$ and $\sigma_{p \rightarrow v}^2 = 0$ for the first iteration. Therefore, the output mutual information is obtained by the J function:

$$I_{v \rightarrow p} = J(\sigma_{v \rightarrow p}) = J\left(\sqrt{d_{vs}\sigma_{s \rightarrow v}^2 + (d_{vp} - 1)\sigma_{p \rightarrow v}^2}\right). \quad (3.40)$$

Now that we have mutual information for LLR values from the VNs to the PCNs. Inside the parity-check nodes, the LLR values are updated following the box-plus operations. Again, a Gaussian distribution is assumed. The derivations for the mutual information update have been accomplished in [79, 80]. For a parity-check node with degree d_c , the mutual information output is given by

$$I_{p \rightarrow v} = 1 - J\left(\sqrt{(d_c - 1)[J^{-1}(1 - I_{v \rightarrow p})]^2}\right). \quad (3.41)$$

Later on, the obtained LLR values are returned to the variable nodes. If the iterations are conducted between the VNs and the PCNs, the output mutual information follows (3.40). If the iterations are performed between the VNs and the MNs, the output mutual information from VNs to the MNs is

$$I_{v \rightarrow p} = J(\sigma_{v \rightarrow p}) = J\left(\sqrt{(d_{vs} - 1)\sigma_{s \rightarrow v}^2 + d_{vp}\sigma_{p \rightarrow v}^2}\right). \quad (3.42)$$

Thus, the transfer curves can be drawn following equations (3.40), (3.41), and (3.42), and the final mutual information output is

$$I_{E,DEC} = I_{v \rightarrow p}. \quad (3.43)$$

This is the example for the computation of a regular code. In case of an irregular code, where the nodes have different degrees, the computations are a little more complicated. The output mutual information is first calculated for each degree. Then, according to its degree distribution, the overall mutual information is the weighted summation of each value. The weighting factor is determined by its degree distribution in the connection prospect. The above mentioned steps are general for both LDHC codes and LDPC codes.

Fig. 3.13 provides an example of the transfer curves of regular LDPC codes with code rate 1/2. Noted that the x -axis represents the output mutual information $I_{E,DEC}$, and the y -axis represents the input mutual information $I_{A,DEC}$. All the illustrated codes have the

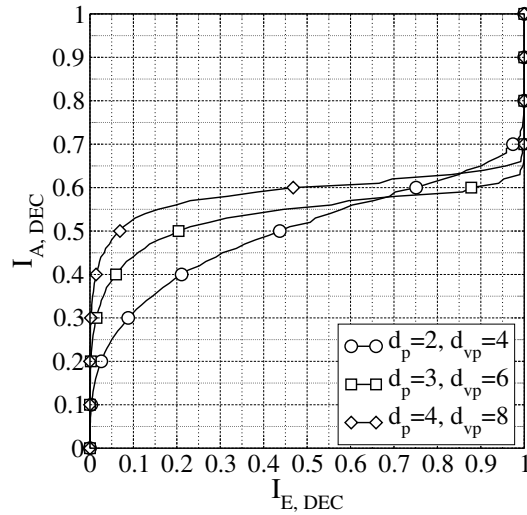


Figure 3.13: EXIT chart for LDPC codes, $R = 1/2$.

same code rate but different degree distribution polynomials. The number of iterations is 20. This is the local iteration (refers to the iteration between the VNs and the PCNs). To keep a fixed code rate, the degrees of both nodes are increased simultaneously. For $d_p = 2, d_{vp} = 4$, the curve in Fig. 3.13 starts lower but ends up higher compared to the other curves. Furthermore, $I_{A,DEC}$ is increasing with $I_{E,DEC}$ increasing. When the degrees are enhanced to $d_p = 3, d_{vp} = 6$, the curve becomes smoother. The increasing of the curve is relatively slow in the range from $I_{E,DEC} = 0.2$ to $I_{E,DEC} = 0.8$. Further increasing the degrees enables the curve to be smoother. However, the decoding complexity is also enhanced due to more connections between the nodes.

3.6 Coded Modulation Curves

In this section, the code design for superposition mapping will be considered. For analysis purpose, both the “DEM” curve and the “DEC” curve will be drawn on the same graph. Take $I_{A,DEM}$ on the x -axis and $I_{E,DEM}$ on the y -axis. Accordingly, $I_{E,DEC}$ will be on the x -axis and $I_{A,DEC}$ will be on the y -axis. Given the two curves are in the same EXIT chart, the trajectory between the two curves shows how the LLR values are improved during iterative processing. If the transmission errors can be corrected after a sufficient number of iterations, the following two conditions must be satisfied. First, there exists a tunnel

between the two curves from $I_{E,DEC} = 0$ to $I_{E,DEC} = 1$. Second, the “DEM” curve should be above the “DEC” curve. The important part is that the trajectory between the two curves can reach $I_{E,DEC} = 1$. Hence, the reliability of the decoder output is important since the final decision is made on it. In Sec. 3.5.2, it has been shown the “DEC” curve is purely determined by the code parameters. Given an LDHC code with known degree distribution and degree allocation, the “DEC” curve is fixed. The “DEM” curves are varied by many factors, but followed by certain pattern. Given a fixed modulation order, the “DEM” curves of different SNRs are going up parallelly. With SNR increases, the curves goes higher in the EXIT chart. Thus, by comparing the “DEM” curves with a fixed “DEC” curve, it is possible to predict at which SNR the BER curve starts to converge.

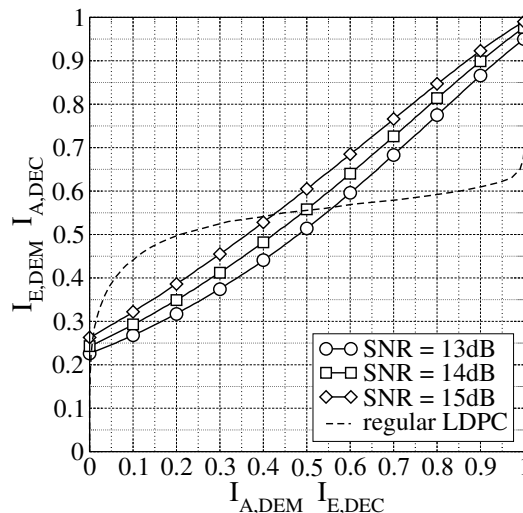


Figure 3.14: EXIT chart for a regular code and PSM with $N = 8$.

As has been observed from Fig. 3.12, the transfer curves for PSM and QAM deliver totally different shapes, especially when the modulation orders are high. For Gray-labeled square QAM, the transfer curves are almost flat from $I_{A,DEM} = 0$ to $I_{A,DEM} = 1$. The “DEC” curves provided by Fig. 3.13 deliver smooth shape for high degree LDPC codes. Hence, it is possible to have a tunnel in between these two curves so that the iterative systems can converge. Now PSM with $N = 8$ is considered. The “DEM” curves of PSM for different E_S/N_0 are illustrated in Fig. 3.14. As the figure shows, the transfer curves always start from a relatively low $I_{E,DEM}$ and end at a higher $I_{E,DEM}$. If the regular LDPC codes provided by Fig. 3.13 are connected, there will be a crossover of the two curves. The

trajectory of LLR improvement will also stop at the crossing point. The bit errors can not be fully corrected. Thus, a suitable channel code is required to be found so that it matches the transfer curve of superposition mapping in the EXIT chart. Since a regular LDPC code is not a suitable match, it is possible to have other codes as alternatives. In LDHC design, two ways of code adaptation will be considered: codes generated by degree distribution or by degree allocation.

Degree Distribution

First, adaptation via the degree distribution is considered. In this case, the degree of each VN w.r.t. the MNs is set to be one. In Fig. 3.15, the “DEM” curves for PSM with E_s/N_0 from 13 dB to 15 dB are illustrated. The “DEM” curves are approximately parallel. As has been mentioned before, regular LDPC codes yield smooth curves, but at different levels. For a long code word, the nodes can have different degrees. If wisely choosing the fractions of the nodes with different degrees, it is possible to obtain the desirable “DEC” curve. This is the motivation of using irregular code via the degree distribution adaption.

In Fig. 3.15, an irregular LDHC code with the following degree distribution polynomials are picked. The VN degree (w.r.t. PCNs) distribution polynomial is

$$\begin{aligned}\lambda_p(X) &= \sum_{d=1}^{D_{vp}} \lambda_{dp} X^{d-1} \\ &= 0.75X + 0.10X^2 + 0.12X^3 \\ &\quad + 0.01X^{15} + 0.01X^{19} + 0.01X^{35},\end{aligned}\tag{3.44}$$

and the PCN degree distribution polynomial is

$$\begin{aligned}\rho(X) &= \sum_{d=1}^{D_p} \rho_d X^{d-1} \\ &= 0.31X + 0.42X^3 + 0.23X^7 \\ &\quad + 0.01X^{15} + 0.01X^{23} + 0.01X^{45} + 0.01X^{99}.\end{aligned}\tag{3.45}$$

According to the degree distribution polynomials, the resulting code rate is 1/2. The number of iterations between the VNs and the PCNs is set to be 10. The transfer curve

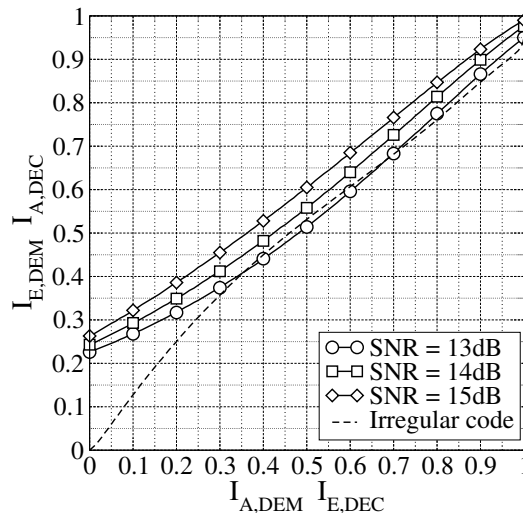


Figure 3.15: EXIT chart for an irregular code and PSM with $N = 8$.

of this irregular curve is meant to be below the required “DEM” curve for all the values on the x -axis. From Fig. 3.15, there is still a crossover between the “DEC” curve and the “DEM” curve at SNR=13 dB. The tunnel between the two curves open when SNR=14 dB, which suggests that the BER curve will converge at an SNR of 14 dB given long enough sequence length and enough iterations. It can also be observed from the figure that the “DEC” reaches $I_{E,DEC} = 1$ very late. This could make the crossing point not necessarily at the rightmost point. Hence, an error floor may occur.

Degree Allocation

The LDHC code can also be designed by the degree allocation. If the VN degree (w.r.t. MNs) is also taken into consideration, the overall code is composed of two parts: R_s and R_p . Based on their degree distributions, R_s and R_p is computed by (3.26) and (3.24), respectively. The overall rate of an LDHC code is given by

$$R = R_s R_p . \quad (3.46)$$

To keep consistent with the previous results, the overall code rate is again fixed to be 1/2. The proportions of the rates can be allocated between R_s and R_p . Normally an LDPC code with higher rate and a repetition code with lower rate is preferable. Hence, the following degree allocations are considered.

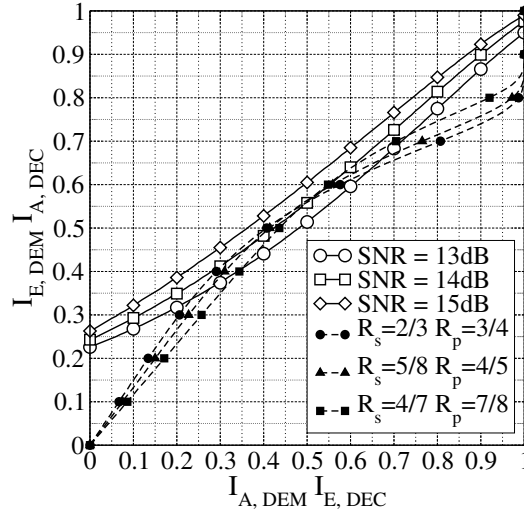


Figure 3.16: EXIT chart for LDHC codes with different degree allocations, $N = 8$.

In case $R_s = 2/3$ and $R_p = 3/4$, the degree distributions are:

$$\lambda_s(X) = 0.50 + 0.50X, \quad \lambda_p(X) = X, \quad \rho(X) = X^7. \quad (3.47)$$

In case $R_s = 5/8$ and $R_p = 4/5$, the degree distributions are:

$$\lambda_s(X) = 0.40 + 0.60X, \quad \lambda_p(X) = X, \quad \rho(X) = X^9. \quad (3.48)$$

In case $R_s = 3/5$ and $R_p = 5/6$, the degree distributions are:

$$\lambda_s(X) = 1/3 + 2X/3, \quad \lambda_p(X) = X, \quad \rho(X) = X^{11}. \quad (3.49)$$

In case $R_s = 4/7$ and $R_p = 7/8$, the degree distributions are:

$$\lambda_s(X) = 0.25 + 0.75X, \quad \lambda_p(X) = X, \quad \rho(X) = X^{15}. \quad (3.50)$$

In all of the above codes, the degrees between the VNs and the PCNs are all constant. However, the VN degrees (w.r.t. MNs) are varying. Hence, all of them can also be seen as irregular codes. It can also be observed from these equations that if R_p is high, the required PCN degree is also high. Sequentially, the decoding complexity is enormous.

We compare three concatenated combinations in Fig. 3.16. The same ‘‘DEM’’ curves as in the previous two figures are illustrated as well. The tendency of the concatenated codes is that the reduction of R_s means increasing the repetition code part, meanwhile the

parity-check code part is decreased. From the EXIT chart, when R_s becomes lower, the curves go down in the low $I_{A,DEM}$ regime. However, from $I_{A,DEM} = 0.5$ to $I_{A,DEM} = 1.0$, the curve with the lowest R_s is the highest in the chart. The addition of the repetition part will lead to a descent in the starting region and a rise in the ending region. When combining with the “DEM” curves, there are still crossovers between the “DEC” curves and the “DEM” curve at SNR of 14 dB. The tunnel is open for SNR=15 dB for all the “DEC” curves. Comparing with the transfer curve of irregular codes, the concatenated code converges 1 dB later. On the other hand, the “DEC” curves reach $I_{E,DEC} = 1$ quite late, which guarantees a crossing point at (1.0, 1.0).

3.7 Numerical Results

In the EXIT chart analysis, we have discussed two methods to find the code match to phase-shifted superposition mapping. One is by varying the degrees between the PCNs and VNs. The other method is by varying the degree of VNs (to the MNs). However, the EXIT chart analysis assumes that the code word length is extremely long so that the code bits are mutually independent. In practical applications, these conditions can not be completely fulfilled. How the two code designs perform with moderate burst length will be investigated in this section. The two ways of code adaptation are compared to search for an ideal combination of coding and modulation via simulations. The code rate is set to be 1/2. In the following results, QAM is Gray-labeled and PSM is natural-labeled.

3.7.1 Code Design by Degree Distribution

The picked irregular code here follows the EXIT chart analysis. The degree distribution polynomials given by (3.44) and (3.45) are chosen. In the EXIT chart, the tunnel opens at 14 dB so that the transmission shall be converge at 14 dB if iterative processing is successful. The simulation results are shown in Fig. 3.17. The codeword length is 6000000. The iterations inside the code and the iteration between the decoder and the demodulator are both 100. The capacity curve of 4 bits/symbol is also provided as a lower

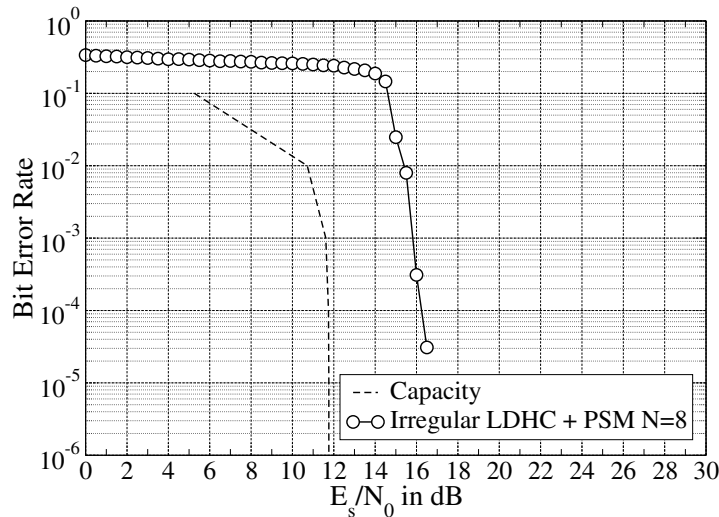


Figure 3.17: BER vs. SNR, irregular LDHC code with PSM.

bound. As the results show, the curve starts to converge at approximately 14.5 dB. The BER reaches 10^{-6} at 16.8 dB. Although the chosen burst length and the iteration number is extremely high, the simulation results are different from the EXIT chart analysis. As a matter of fact, adapting the degree distribution does help improve the performance. It has been shown in [36] that SM-GPA delivers a superior performance. In [81], combining an irregular convolutional code with PSM ($N = 12$) also approaches the capacity bound. However, in both cases, a very large code word length and a large number of iterations are mandatory. How this method will behave for a moderate burst length will be shown in the next simulation.

3.7.2 Code Design by Degree Allocation

In this section, the chosen number of code bits is 6000, which is not a very large number. Again, the chosen code is an LDHC code with rate $1/2$. In Fig. 3.18, the irregular LDHC code follows the degree distribution polynomials by (3.44) and (3.45). This is the same code as the one in the previous simulation. However, this time the BER curve starts to converge at 18 dB. An approximately 2 dB performance loss can be observed by reducing the sequence length. The problem of the irregular codes are the nodes with low degrees. For long sequences, it is possible to enable these low degree nodes to form a long cycle.

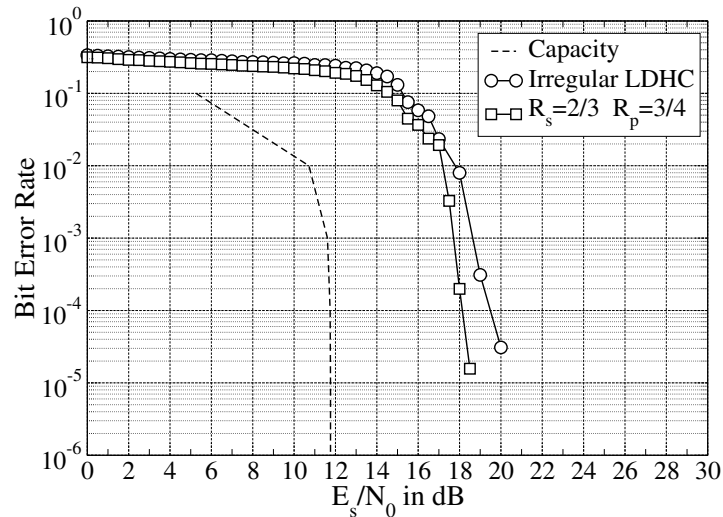


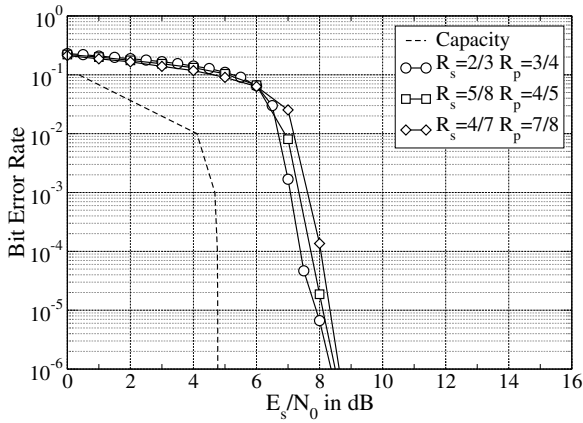
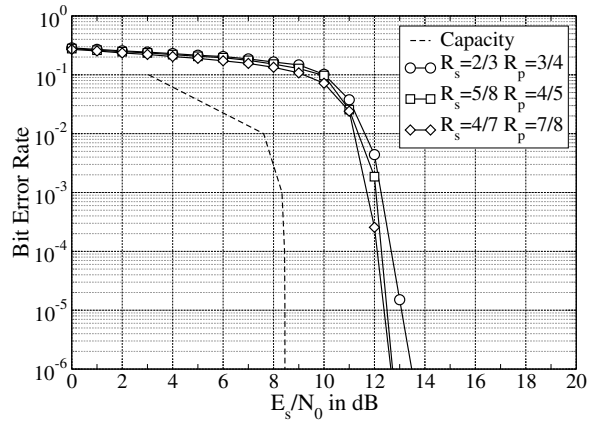
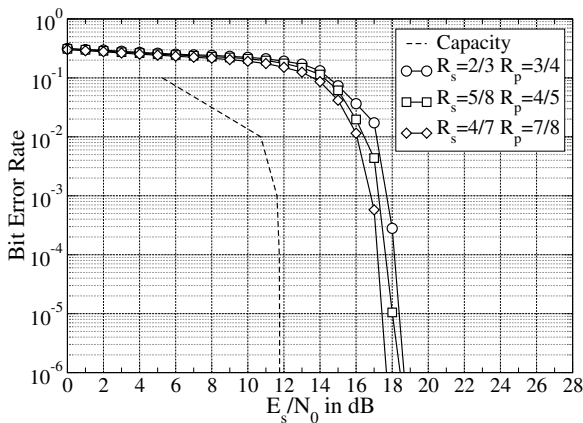
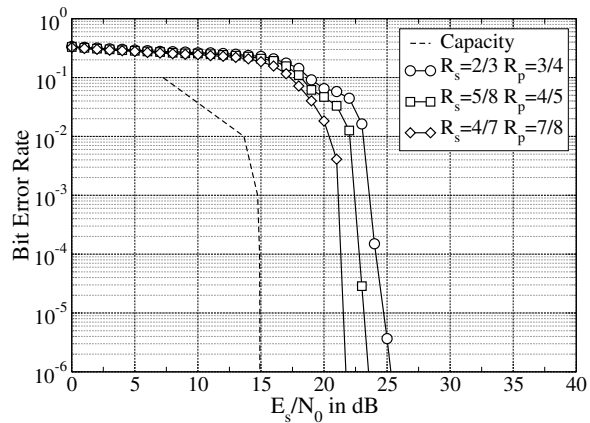
Figure 3.18: BER performance comparisons between two methods.

However, for short sequences, this condition is not easy to accomplish. Hence, trapping sets will occur and the transmission errors inside such trapping sets are not easy to be corrected. Fig. 3.18 also shows an LDHC code by degree allocation with $R_s = 2/3$ and $R_p = 3/4$. Surprisingly, the BER curve of this code deliver a more than 1 dB gain over the other case at the BER of 10^{-6} . It seems to contradictive to our EXIT chart analysis. However, in the EXIT chart analysis, a cycle-free code is presumed. In reality, when the effects of cycles can not be ignored, the code by degree allocation shows a slightly performance improvement. It also indicates that such a way of adaptation is not strongly affected by the short cycles. Due to this reason, in the following, the code design by degree allocation is of main focus.

3.7.3 Effects of Degree Allocations

In the previous simulation, we only take one possible degree allocation as an example for the analysis of the system. In this section, PSM with three possible degree allocations are provided for different modulation orders in Fig. 3.19. With an overall code rate of $1/2$, R_s and R_p are adapted, with their degree distribution polynomials provided by equations from (3.47) to (3.50). $N = 2$ is not compared because the constellations of square QAM and PSM are equivalent. For $N = 4$, the BER performance is reducing when the repetition

part is increasing. As has been shown in the EXIT chart, the “DEM“ curve of $N = 4$ is relatively smooth. Thus, a pure LDPC code would be more fit. When further increasing N , the curves are more gradient in the EXIT chart. Thus, a code with repetition part would be more suitable. The simulation result coincides with the EXIT analysis. For both $N = 6$ and $N = 8$, by increasing the repetition part, the performance is improving gradually. In case of $N = 8$, the performance improvement is even more obvious, as can be seen in Fig. 3.19(d). From the degree distribution polynomials c.f. (3.47) to (3.50), further increasing R_p will increase the degree of the parity-check nodes. As a result, the decoding complexity will be further increased.

(a) $N = 4, BW = 2$.(b) $N = 6, BW = 3$.(c) $N = 8, BW = 4$.(d) $N = 10, BW = 5$.Figure 3.19: BER vs. SNR for different degree allocations, $R = 1/2$.

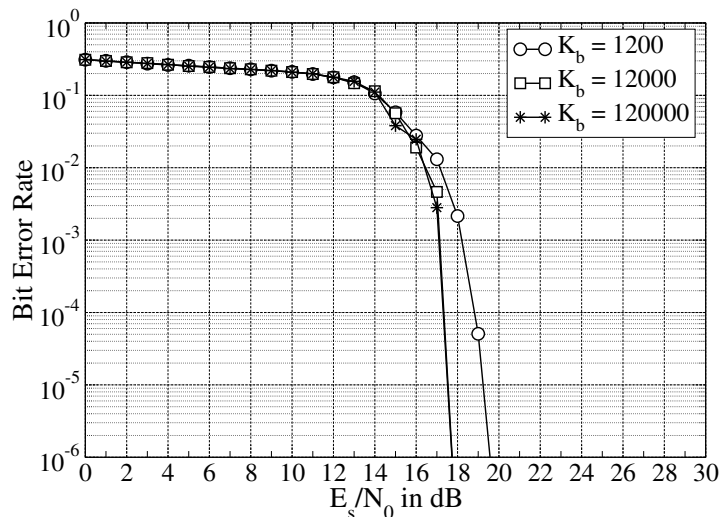


Figure 3.20: BER vs. SNR for different burst lengths.

3.7.4 Effects of Burst Length

As has been mentioned before, the burst length is crucial to the performance in case of designing by degree distributions. The reason is due to the large number of low degree nodes. These nodes easily generate short cycles. Hence, the trapped sets are formed and the error can hardly be mitigated. Thus, in the previous simulations, there is an obvious performance degradation by reducing the burst length given a fixed irregular LDHC code. In this section, we will see how the burst length affects the concatenated codes. In Fig. 3.20, the systems with different info word lengths K_b are tested. The chosen degree allocation is $R_s = 5/8$ and $R_p = 4/5$, with the degree distribution polynomials given by (3.48). The number of iterations for both global iterations and local iterations are 10. As the simulation results turn out, when the info bits length is short, the short cycles inside the code can barely be avoided. A higher SNR is required for the system to converge. However, when increasing K_b to 12000, the performance tends to be stable. There is a nearly 1.5 dB performance improvement. However, further increasing K_b to 120000, there is no obvious performance gain from the BER results. This indicates that 12000 would be sufficient and this way of code design is stable for a moderate burst length. The burst length does affect the performance of the codes by degree allocation, but only in a small regime. This could be seen as an advantage over the code design by degree distributions.

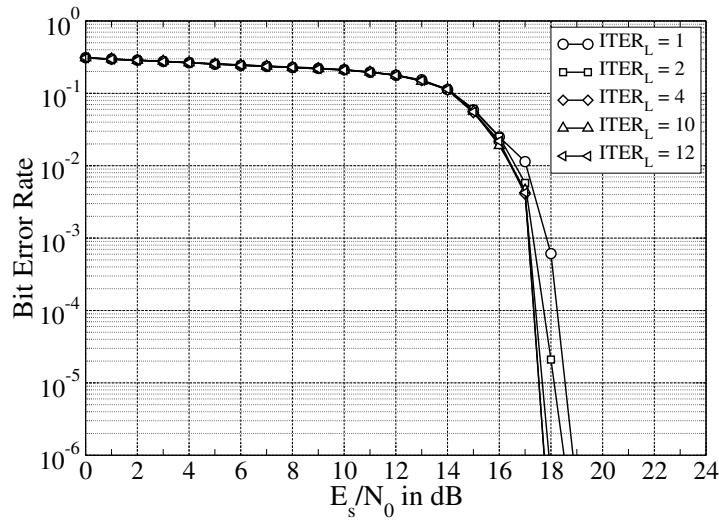
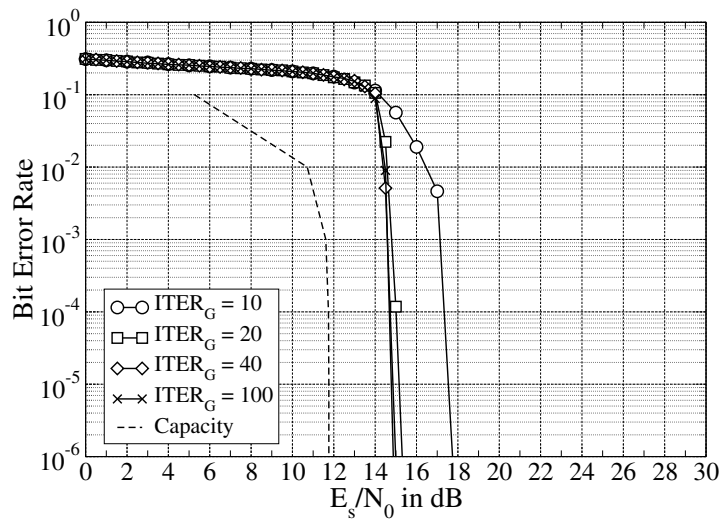
(a) Different local iterations, $ITER_G = 10$.(b) Different global iterations, $ITER_L = 4$.

Figure 3.21: BER vs. SNR for different local and global iterations.

3.7.5 Effects of Iterations

In iterative processing, a capacity achieving performance requires a long sequence length and a large number of iterations. The effects of sequence length has been investigated. The effects of iterations will be focused herein. There are two kinds of iterations in LDHC codes. The first one is termed local iterations, which are between the variable nodes and the parity-check nodes. Here, they are denoted by $ITER_L$. The second one is global iterations between the variable nodes and the mapping nodes. The global iterations are denoted by $ITER_G$. The global iterations are conducted after local iterations are finished.

The local iterations enable to flatten the “DEC” curves in the EXIT chart. The LLR values are improving as the global iterations between the decoder and the demodulator proceed. Hence, a proper choice of both iterations would be helpful for the convergence behavior of coded modulation transmission.

In the following, the code parameters and the modulation orders are the same as in Section 3.7.4. In Fig. 3.21(a), the number of global iterations $ITER_G$ is fixed to be 10 while the number of local iterations is varying from 1 to 12. Comparing with $ITER_L = 1$ and $ITER_L = 2$, a clear performance gain can be observed. Further increasing the iteration number provides a less than 1 dB gain. Once $ITER_L$ reaches 10, the performance will not be improved even more iterations are carried out. This is quite different from the common knowledge in LDPC codes that the iterations inside codes play an essential part in improving the SNR threshold. However, it can be observed here the effect of local iterations only imposes a slight performance improvement in LDHC codes.

On the other hand, the effect of global iterations is another story. The corresponding results are shown in Fig. 3.21(b). In these simulations, the local iteration number is fixed to be 4 while the other parameters are the same as the previous simulation. When the number increases from 10 to 20, there is more than 2 dB performance improvement. Further increasing the local iteration number, the performance gain can still be achieved, even when the number is as high as 100. This can be explained through the EXIT chart analysis. Revisiting Fig. 3.16, the tunnel between the two curves are rather narrow. Thus, in order to pass through the tunnel and reach the rightmost point, a large number of iterations between decoder and demodulator is required. The dashed curve indicating the performance bound is provided here as a reference. With the increased number of iterations, the BER curve can converge at 15 dB, which agrees with the EXIT chart result shown in Fig. 3.16. The EXIT chart analysis says that the concatenated codes open the tunnel at 15 dB. Thus, this reaches the limit for this coded modulation combination. However, there is still a gap to the capacity bound of 4 bits/symbol. In order to reach that limit, an irregular code with a very large number of sequence length is needed.

3.7.6 Coded Modulation Comparison

Square QAM has been a well-known standard modulation scheme for decades. Here, it can be used as a good benchmark. LDPC code can also be a good counterpart to the LDHC code. In Fig. 3.22, the LDHC coded PSM with natural labeling follows the setups as mentioned in the previous sections with $\text{ITER}_L = 4$ and $\text{ITER}_G = 20$. In case of LDPC coded square QAM, the parameters are as follows. The modulation is also $N = 8$. A regular LDPC code with variable node degree $d_v = 3$ and the parity-check node degree $d_c = 6$ is chosen. The resulting code rate is also $1/2$. Contrary to LDHC codes, more iterations should be put in the local iterations. Hence, $\text{ITER}_G = 4$ and $\text{ITER}_L = 20$ is picked for LDHC coded PSM. For LDPC coded square QAM, $\text{ITER}_G = 4$ and $\text{ITER}_L = 20$ is chosen. From the simulation results, the LDHC-PSM outperforms LDPC-QAM by approximately 1.5 dB. The capacity curve of 4 bits/symbol is illustrated here as a reference. The combination of LDHC coded PSM (with natural labeling) is optimum in terms of EXIT results. However, it is still 3dB away from the capacity, which implies a better code can be used for further improvement.

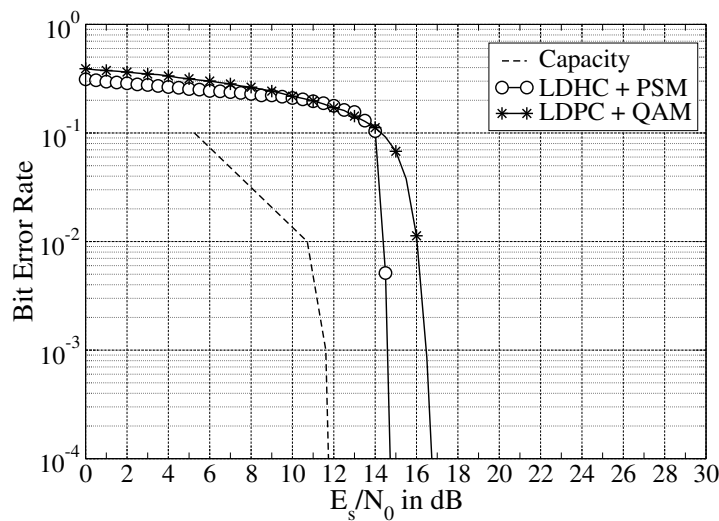


Figure 3.22: LDHC-PSM vs. LDPC-QAM.

3.8 Summary

In this chapter, a universal design strategy, LDHC coding, has been introduced. Utilizing an LDHC code, the coding and modulation can be designed jointly. By means of EXIT chart analysis, the transfer curves for PSM are compared with those of square QAM. The reasons why the conventional designed codes do not fit for PSM are provided. Later on, two methods have been used for finding a suitable coded modulation combination for LDHC coded PSM. It turns out the degree allocation provides a stable performance for moderate burst length. The results also show that more effects should be taken on global iterations instead of the local iterations, which is also different from the conventional acknowledgments.

Chapter 4

Superposition Mapping for OFDM

In the previous chapters, bit-interleaved coded modulation with superposition mapping has been introduced. From the EXIT chart analysis, conventional LDPC codes are not a perfect match for this modulation scheme. In order to match the transfer curves of superposition mapping, concatenated codes are more suitable. Based on this observation, a universal design method, dubbed LDHC code structure, has been applied to design the coding and the modulation jointly. Such a general way of designing systems is able to provide a performance improvement compared to the conventional coded modulation designed for the AWGN channel model. The AWGN channel model is more used for theoretical analysis. In practical applications, more sophisticated channels are considered.

In this chapter, the performance of BICM transmission through a time-invariant frequency-selective channel is considered. In order to mitigate the intersymbol interference in a frequency-selective channel, the orthogonal frequency-division multiplexing (OFDM) technique is often applied. By doing so, the frequency-selective channel will be treated as a number of frequency-flat subchannels with different channel coefficients. It would be theoretically beneficial to apply the near Gaussian distributed mapping scheme on these subchannels with water-filling. We have proposed a structure for superposition mapping with BICM-OFDM in [82]. In [83], the LDHC philosophy has shown to be utilized to this system structure. This will be the main focus of this chapter.

4.1 Theoretical Benefit

The considered channel throughout this chapter will be a frequency-selective channel. For simplification, the channel is also assumed to be constant in the time domain. OFDM is used to mitigate the intersymbol interference. Assume there are J symbols in one OFDM symbol. Hence, the channel is equivalent to J parallel subchannels, all of which are frequency-flat fading. Considering the j th subchannel, the relationship between the channel input x_j and channel output y_j is

$$y_j = h_j x_j + z_j, \quad 0 \leq j \leq J-1, \quad (4.1)$$

with j denotes the subchannel index. The channel coefficient h_j is different from subchannel to subchannel. It is also assumed that the noise in each subchannel follows a Gaussian distribution with the same mean and variance

$$z_j \sim \mathcal{N}(0, \sigma_z^2). \quad (4.2)$$

We assume that the transmit symbols are normalized to one. Hence, the j th subchannel capacity is given by

$$C_j = \log \left(1 + \frac{|h_j|^2 E_s}{N_0} \right) = \log \left(1 + \frac{|h_j|^2}{\sigma_z^2} \right). \quad (4.3)$$

Again, the capacity is achieved if and only if x_j is Gaussian distributed. The overall capacity is the average of the subchannel capacities, which is written by

$$C = \frac{1}{J} \sum_{j=0}^{J-1} C_j = \frac{1}{J} \sum_{j=0}^{J-1} \log \left(1 + \frac{|h_j|^2}{\sigma_z^2} \right). \quad (4.4)$$

From (4.4), the capacity is achieved on the condition that the transmit symbols of the subchannels are all Gaussian variables. It has previously been shown that superposition mapping is capable of generating a quasi Gaussian distributed transmit symbol. Hence, it is naturally to use this kind of modulation to approach the channel capacity for each parallel subchannel. The subchannels allow for two-dimensional modulation transmission. Thus, we continue to use PSM as the modulation scheme for investigation in the rest of this chapter.

4.2 Water-filling Algorithm

Although the quasi Gaussian distributed signal on each subchannel maximizes the overall capacity, the performance of the whole system still has room for improvement. Due to the difference in subchannel coefficients, the channel capacity varies from subchannel to subchannel. If the same setup (code rate, modulation order) is applied on each subchannel, each will deliver a different bit error rate under the same SNR. It is usually desirable to make all the subchannels have the same error probability. Otherwise, if some subchannels have error probabilities which are much higher than the other subchannels, these subchannels will dominate the BER performance. Given the channel state information (CSI) is known to both the transmitter and the receiver, the transmit power and the transmission rate for each subchannel can be adapted in order to match for the channel characteristic. This method is termed water-filling [84]. In the original definition, the so-called water-filling algorithm is applied to maximize the overall capacity under a given total energy and a target error probability. According to the system requirements, the water-filling algorithm can be separated into two categories [85]. One is to maximize the overall data rate. The other is to maximize the performance margin under a given fixed data rate. In either case, the water-filling algorithm needs to compute the transmit power and the data rate for each subchannel. In our system setup, the average transmit power of each subchannel is normalized to one. The code rate is controlled by the code parameters, which is also fixed. Hence, the factor that can be altered is the modulation order. Sequentially, this is a bit loading problem. The goal is to determine the suitable number of bits to be transmitted for each subchannel via water-filling.

4.2.1 Discrete Bit Loading

The water-filling solution provides a unique bit allocation, which is theoretical optimum. The optimum solution allows a real-valued bit allocation. However, in practical applications, the modulation orders are integer values. To solve this problem, many sub-optimum bit loading algorithms with discrete bit allocation have been proposed [86, 87, 88, 89, 90, 91, 82]. In 1987, the Hughes-Hartogs algorithm [86] has been

proposed which is based on the greedy optimization. The Hughes-Hartogs algorithm is performed by assigning an additional bit to the subcarrier which requires the smallest transmit power. However, this algorithm needs a large number of sorting and search procedures. Later, Campello de Souza [87] and Levin [88] developed the so-called Levin-Campello (LC) algorithm. The LC algorithm improves the original Huges-Hartogs method with a simpler complexity. In 1995, there comes an algorithm by Chow, Cioffi, and Bingham [89]. The Chow algorithm initializes the bit allocation by rounding of the bit distribution obtained by water-filling, which greatly reduces the sorting and search procedures. Later in 1996, Fischer [90] also proposed a low-complexity bit loading algorithm via measuring the error probabilities, which is especially suitable for square QAM. All of the above mentioned bit loading algorithms lay good foundations for our work. For a fixed condition, the bit allocations determined by the Hughes-Hartogs algorithm, the LC algorithm, and the Chow algorithm are similar. The performance difference is tiny while the computational complexity difference is significant. The Hughes-Hartogs algorithm offers a slightly better performance. The Chow algorithm delivers the lowest complexity. On the other hand, the Fischer algorithm is specified for square QAM. The other three algorithms are valid regardless of the modulation schemes. Taking all the factors into account, the Chow algorithm will be chosen for computing the bit allocation in our BICM-OFDM transmission with PSM.

Compared to the other algorithms, the complexity reduction of the Chow algorithm lies in its initialization stage. A rough value of the bit allocation is obtained by rounding the values determined by the water-filling. Consider a number of J flat fading subchannels. The transmit power for each subchannel is normalized. The target average number of bits is \bar{N} . The algorithm of Chow is summarized as follows:

1. The signal-to-noise ratio for each subchannel is computed by

$$\text{SNR}_j = |h_j|^2 / \sigma_z^2. \quad (4.5)$$

2. The subchannels with no bit transmitted are termed the unused subchannels. Thus, let J_u denote the used subchannels, where $J_u \leq J$, γ denote the current system performance margin. Set $J_u = J$ and $\gamma = 0$.

3. For the subchannels from $j = 0$ to $J - 1$, compute the following values:

$$b_j = \log \left(1 + \frac{\text{SNR}_j}{1 + \gamma} \right), \quad (4.6)$$

$$\hat{b}_j = \lfloor b_j \rfloor, \quad (4.7)$$

$$D_j = b_j - \hat{b}_j. \quad (4.8)$$

If $b_j = 0$ for a certain subchannel, this subchannel is set as unused, and

$$J'_u = J_u - 1. \quad (4.9)$$

4. Compute the new margin by

$$D_o = \sum_{j=0}^{J-1} \hat{b}_j - \bar{N}J, \quad (4.10)$$

$$\gamma' = \gamma + 10 \log_{10}(2^{D_o/J_u}). \quad (4.11)$$

5. If $\sum_{j=0}^{J-1} \hat{b}_j = \bar{N}J$, stop the iteration. Otherwise, go back to step 3.

The previous steps only finish the initialization stage. It is possible that after a few iterations, the number of the computed aggregate bits is not equal to the target number of bits. The following steps are considered in two situations:

- 1) If $\sum_{j=0}^{J-1} \hat{b}_j > \bar{N}J$, find the subchannel index j which has the smallest D_j , and subtract one bit for that subchannel. If the number of subchannels becomes zero, then stop reducing bits at this subchannel. When $\sum_{j=0}^{J-1} \hat{b}_j = \bar{N}J$, the procedure stops.
- 2) If $\sum_{j=0}^{J-1} \hat{b}_j < \bar{N}J$, find the subchannel with the largest D_j , and add one bit for that subchannel, also until the aggregate bit number reaches the target bit number.

At last, the final \hat{b}_j is the modulation order for the j th subchannel. In some cases that the modulation orders are required to be even numbers, such as square QAM, the procedure is slightly modified by setting the bit step to be two and forcing the bit rounding in (4.7) to be even values.

4.3 Mutual Information

The capacity of the channel given by (4.4) is the theoretical upper bound. If the channel is fixed, the capacity is also fixed regardless of the modulation schemes and the bit loading algorithms. In transmissions with finite symbol alphabets, the maximum achievable rate of a specific modulation scheme is represented by the mutual information. OFDM transmission supports J subchannels. If the modulation order varies over the subchannels, the overall mutual information \bar{I} is the average of the mutual information of all the subchannels, which is of the form

$$\bar{I} = \frac{1}{J} \sum_{j=0}^{J-1} I(x_j; y_j) , \quad (4.12)$$

where $I(x_j; y_j)$ is the mutual information of the j th subchannel. The calculation of the mutual information for an AWGN channel has been introduced in Chapter 2, by using (2.15). Here, it can be extended to the case of flat fading channels as a flat fading channel is equivalent to an AWGN channel with a channel coefficient. For the j th subchannel with channel coefficient h_j , according to the definition, we have

$$I(x_j; y_j) = h(y_j) - h(y_j|x_j) = h(y_j) - h(z) . \quad (4.13)$$

Taking the channel coefficient into account, the transition probability can be expressed by

$$p(y_j) = \sum_{x_j \in \mathcal{X}_j} P(x_j) \frac{1}{\sqrt{\pi\sigma_z^2}} \exp\left(-\frac{(y_j - h_j x_j)^2}{\sigma_z^2}\right) , \quad (4.14)$$

where \mathcal{X}_j is the cardinality of the transmitted symbols, which is determined by two factors: the modulation scheme and its modulation order. Applying a proper bit loading algorithm, \mathcal{X}_j will be different from subchannel to subchannel. With the help of numerical programming, the mutual information of each subchannel can be obtained. Sequentially, the mutual information (maximum achievable rate) of the whole channel is computed via summation and averaging.

In Fig. 4.1, the capacity curve and the mutual information curves for different cases are compared. The chosen channel is a time-invariant Rayleigh fading channel with memory length $L = 10$. The symbol length of each OFDM symbol is 256, meaning that the

number of subchannels is also 256. The overall rate is fixed. Since the modulation order is not constant over the subchannels, the overall rate is calculated by averaging over the bit loads of all the subchannels. If \bar{N} denotes the average number of bits per symbol, it is computed by

$$\bar{N} = \frac{1}{J} \sum_{j=0}^{J-1} N_j, \quad (4.15)$$

where N_j is the modulation order for the j th subchannel. Since no channel coding is combined, \bar{N} can also be seen as the bandwidth efficiency of the system.

In these figures, the two considered modulation schemes, square QAM and PSM, are compared. QAM is Gray-labeled. The performance with and without (w/o) bit loading (BL) is also illustrated. The capacity curves are obtained from (4.4) and the mutual information (MI) curves follow (4.12). From the simulation results, it can be observed that the transmission systems with bit loading have higher maximum achievable bandwidth efficiencies than the cases without bit loading. Also, in all the situations, the MI curves of PSM are closer to the capacity bounds than the curves of square QAM when they have not reached their bandwidth efficiency limits. This proves that applying both near Gaussian distributed modulation schemes and a bit loading algorithm have effects in approaching the channel capacity. Consequently, the BER performance will be improved, theoretically speaking. However, the impact of these two methods are different under different bits/symbol. For $\bar{N} = 4$, the gap between PSM and QAM curves is very small. For the same bandwidth efficiency, the gap between the case with bit loading and the case without bit loading is relatively large. This is due to the mismatch to the Gaussian distribution of PSM for lower modulation orders. For example, when $N = 2$, these two modulations are equivalent. The situation is just the opposite in higher-order modulation cases. For $\bar{N} = 8$, the gap between the two modulation schemes is large while the gap between the case with bit loading and the case without bit loading is minor. For higher-order modulations, the constellation between the two modulation schemes are quite different. For $\bar{N} = 6$, both of the two gaps are quite obvious. From this analysis, the effects of the signal distribution are stronger for the higher modulation orders. The effects of the bit loading are stronger for the lower modulation orders. In case of $\bar{N} = 6$, both effects have strong impacts.

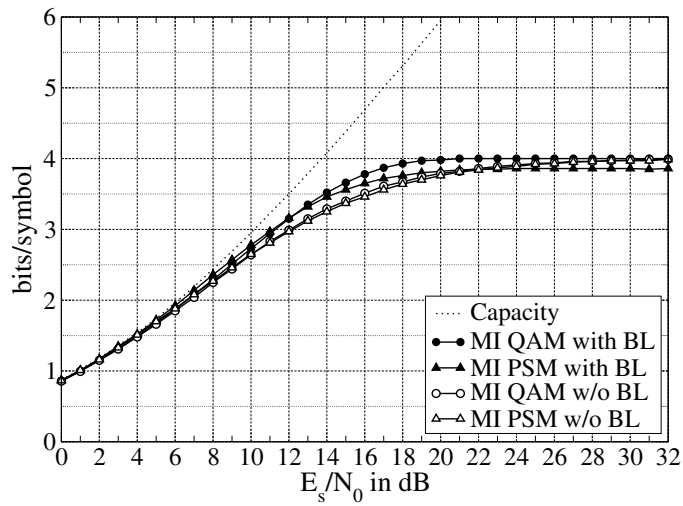
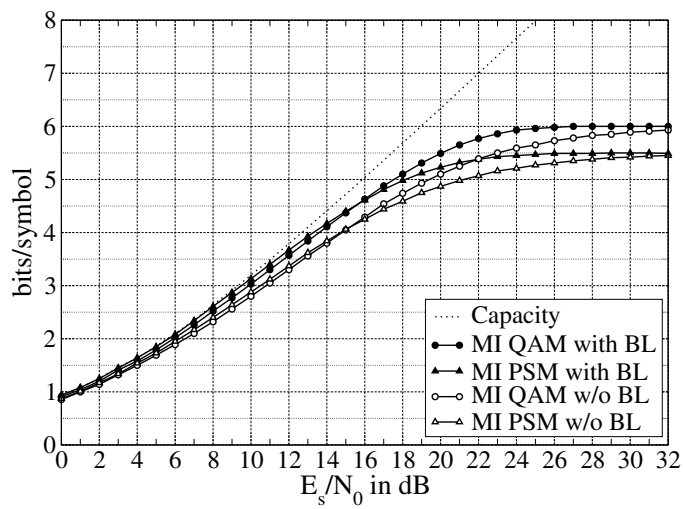
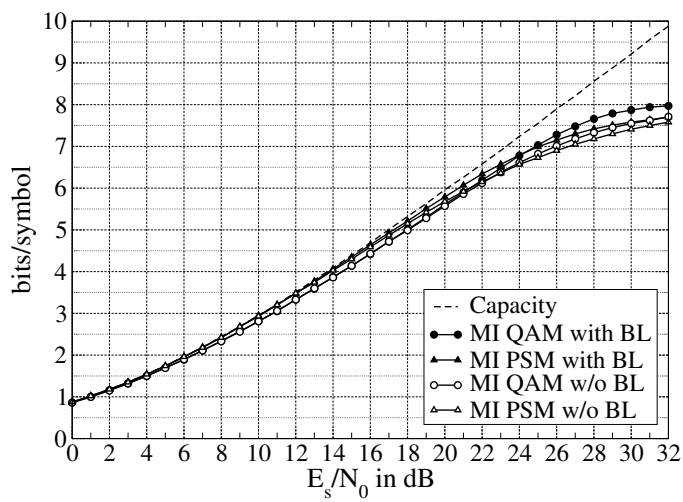
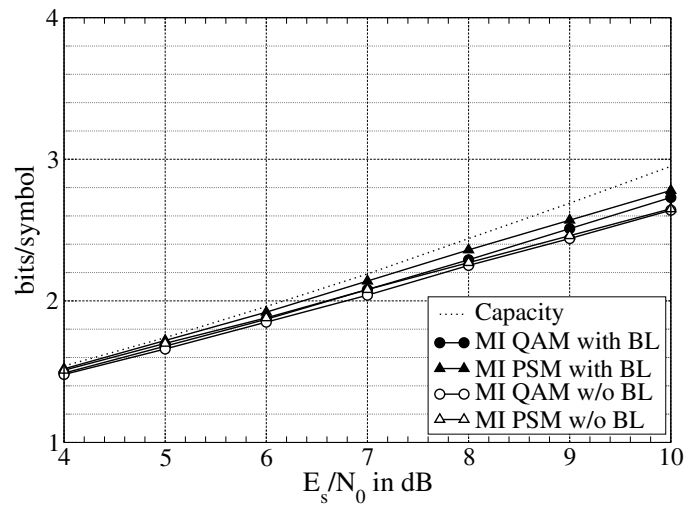
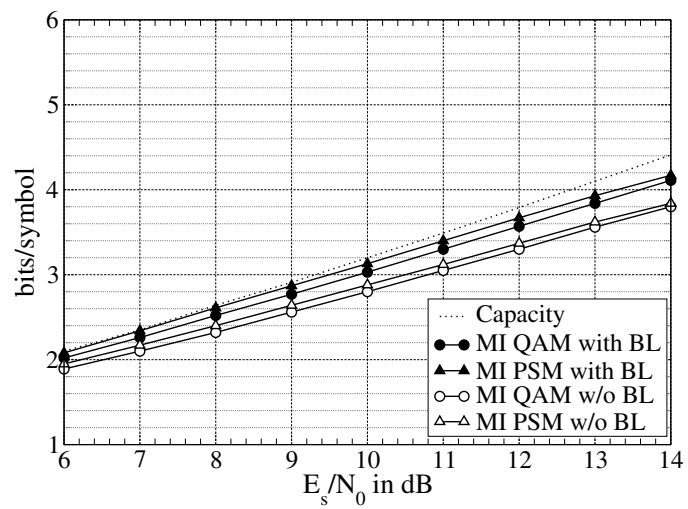
(a) $\bar{N} = 4$.(b) $\bar{N} = 6$.(c) $\bar{N} = 8$.

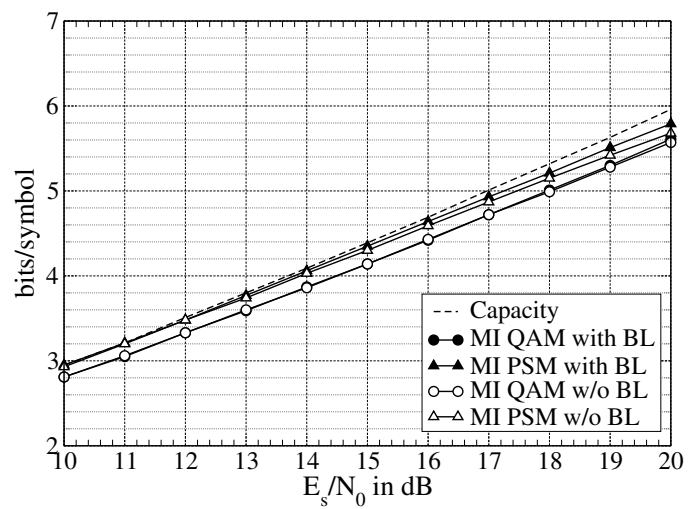
Figure 4.1: MI vs. SNR, QAM vs. PSM.



(a) $\bar{N} = 4$.



(b) $\bar{N} = 6$.



(c) $\bar{N} = 8$.

Figure 4.2: MI vs. SNR, QAM vs. PSM, a closer look.

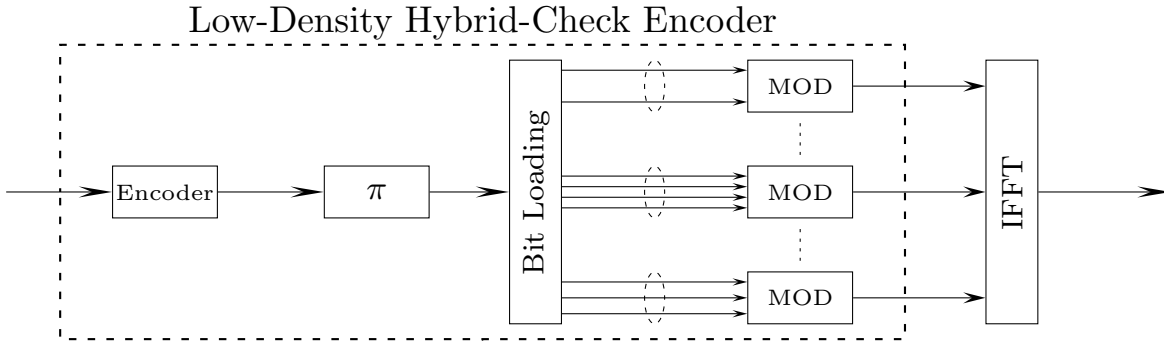


Figure 4.3: Transmitter structure of BICM-OFDM with bit loading.

4.4 LDHC Coded BICM-OFDM

Fig. 4.3 provides the transmitter of a BICM-OFDM transmission. The info bits are encoded and interleaved. Afterwards, the bit loading algorithm is applied in our transmission. Hence, a bit loading block is added before the modulation, as is illustrated in Fig. 4.3. The bit loading block actually has two functions. First, to determine the bit allocation by the Chow algorithm corresponds to their subchannel frequency responses. Second, to divide the interleaved code bits into small groups, with the group size according to the bit allocation. The bits in each group are mapped onto a complex valued symbol. Then the inverse fast Fourier transformation (IFFT) is performed to allocate each symbol onto the subcarrier for transmission.

At the receiver side, the channel observations on each subcarrier are first fast Fourier transformed (FFT). Assuming the CSI and the bit allocation for each subchannel are also known to the receiver, the demodulators are able to generate the a posteriori information of each bit. Here, the demodulators apply the tree-based APP demapping (mentioned in Section 3.2.2) and output the extrinsic LLR values for each code bits. Subsequently, these LLR values are parallel-to-serial (P/S) converted, deinterleaved to be served as the a priori information for the channel decoder. In this system setup, the iterative processing is also performed. Hence, the decoder outputs extrinsic LLR values of code bits. These extrinsic LLR values are interleaved, serial-to-parallel (S/P) converted, and fed back to the parallel demodulators for further detections. The whole block diagram of the receiver is depicted in Fig. 4.4.

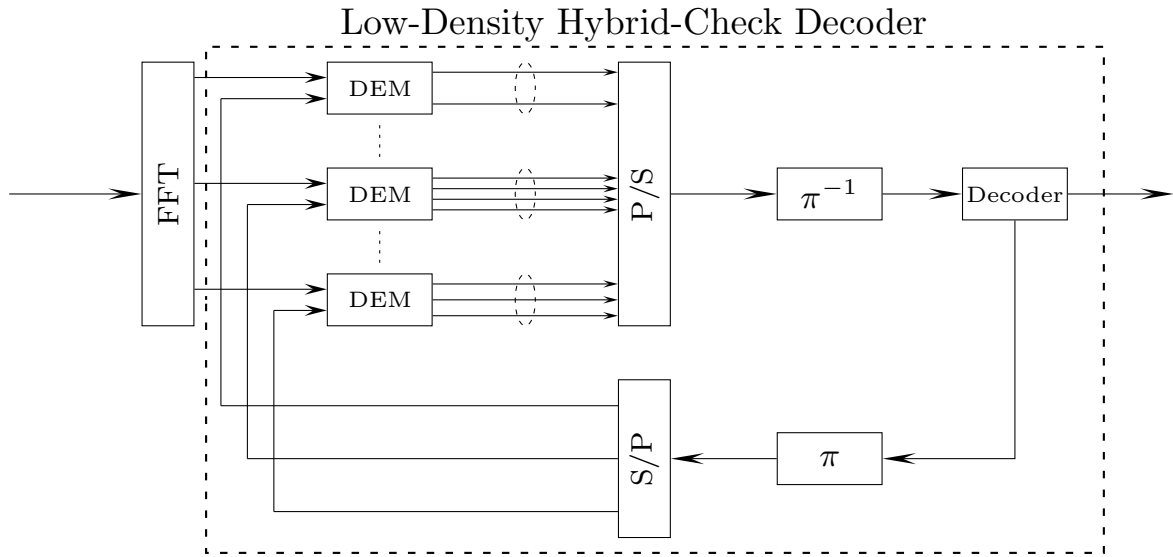


Figure 4.4: Receiver structure of BICM-OFDM.

The transmitter and the receiver structures of BICM-OFDM have been provided in Fig. 4.3 and Fig. 4.4, respectively. In Chapter 3, it has been shown that the LDHC structure can be applied in the BICM transmission for the AWGN channel. In this chapter, it will be shown that the LDHC code structure can also be extended to BICM-OFDM. So far BICM-OFDM has been considered for two cases: with or without bit loading. In case without bit loading, the number of bits to be transmitted for each subchannel is equivalent. The corresponding LDHC structure is similar to the case of an AWGN channel. The other case is with bit loading, which is different from the case of AWGN. In this thesis, an LDHC structure for BICM-OFDM with bit loading is proposed. In the AWGN channel model, the LDHC code is a structure that integrates both the coding and modulation. The difference lies in the bit loading procedure, which is also a part of the transmission. Hence, when considering the LDHC code, this block should also be integrated into the whole structure. Naturally, when designing the LDHC code, not only the coding and the modulation, but the effects caused by the bit loading should also be taken into account as well. As is illustrated by the dashed box in Fig. 4.3, the bit loading has been included inside the LDHC encoder. Likewise, at the receiver side, the P/S converter has been also included in the LDHC decoder. As the iterative processing is also a part of the receiver, the feedback blocks containing the interleaver and the S/P converter are all included inside the LDHC decoder.

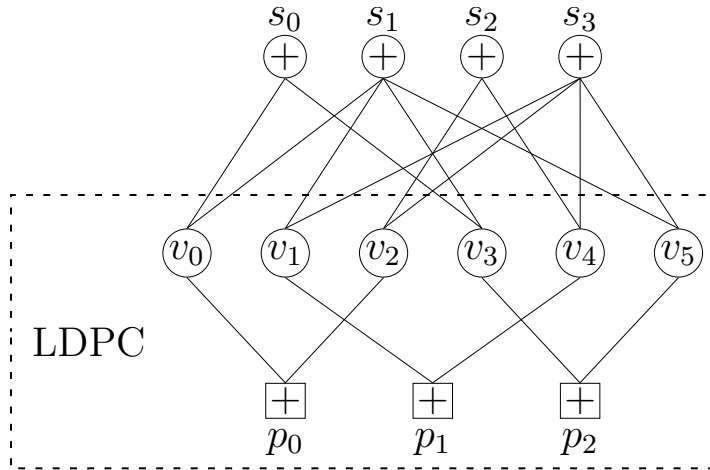


Figure 4.5: Tanner graph of a BICM-OFDM system with 4 subcarriers.

4.4.1 Graph and Matrix Representations

Like in Chapter 3, an LDPC code can be presented by a Tanner graph and a matrix in OFDM. The basic structure is similar. The Tanner graph has three different types of nodes. The complex symbols after modulation are represented by the mapping nodes (MNs). The code bits are denoted by the variable nodes (VNs). The parity-check nodes (PCNs) represent the constraints among the variable nodes. Since the channel is time-invariant, the variable nodes are only in frequency domain. The variable node index is the subchannel index. The bit loading makes the modulation orders of each subchannel different. It has been mentioned that the mapping is represented by the connection between the MNs and the VNs. The degree of one MN is its modulation order. Due to the bit loading, the MN degree is no longer constant as in case of AWGN channels. Fig. 4.5 gives an example of the Tanner graph for one OFDM symbol, where an “ s ” denotes a mapping node, a “ v ” denotes a variable node, and a “ p ” denotes a parity-check node. This OFDM symbol has four subcarriers. Let us assume the bit distribution is (2, 4, 2, 4) according to the bit loading algorithm. Correspondingly, the degree of s in the Tanner graph is also (2, 4, 2, 4). The bit loading makes the mapping nodes part different from the structures discussed in Chapter 3 because the MN degree is no longer constant. The lower part which represents an LDPC code is still the same.

$$\begin{array}{cccccc}
 & v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
 \left[\begin{array}{cccccc}
 1 & 0 & 0 & 1 & 0 & 0 & \leftarrow s_0 \\
 1 & 1 & 0 & 1 & 0 & 1 & \leftarrow s_1 \\
 0 & 0 & 1 & 0 & 1 & 0 & \leftarrow s_2 \\
 0 & 1 & 1 & 0 & 1 & 1 & \leftarrow s_3 \\
 \hline
 1 & 0 & 1 & 0 & 0 & 0 & \leftarrow p_0 \\
 0 & 1 & 0 & 0 & 1 & 0 & \leftarrow p_1 \\
 0 & 0 & 0 & 1 & 0 & 1 & \leftarrow p_2
 \end{array} \right] \\
 \left. \begin{array}{l} \downarrow \\ \downarrow \\ \downarrow \end{array} \right\} \text{LDPC} \left. \begin{array}{l} \uparrow \\ \uparrow \\ \uparrow \end{array} \right\} \\
 & u_0 & u_1 & u_2 & & & &
 \end{array}$$

Figure 4.6: Incidence matrix of a BICM-OFDM system.

The incidence matrix corresponding to this Tanner graph is illustrated in Fig. 4.6. The matrix can also be separately into two submatrices. The lower submatrix represents the LDPC code. Each row denotes a parity-check node, with row weight equals its degree. Likewise, each column denotes a variable node, with the column weight equals its degree. The upper submatrix contains both the modulation and the adaptive bit loading. In the upper submatrix, the j th row represents the symbol transmitted at the j th subchannel. The row weight equals to the degree of the corresponding mapping node. Due to the bit loading, the row weights in the upper part of the submatrix are varied. As in the graph representation, the bit loading is $(2, 4, 2, 4)$, accordingly, the row weight is also $(2, 4, 2, 4)$, as can be observed in Fig. 4.6.

In conventional code designs, many people only focus on the optimization of the LDPC submatrix. Based on the LDHC structure, the incidence matrix now contains not only the coding, but the modulation and bit loading as well. Following the LDHC philosophy, the problem is the design of the LDHC incidence matrix. By doing so, it provides a way of designing in a general way, where the effects of coding, modulation, and bit loading are all considered. Besides the degree distribution and the degree allocation issues, which have been discussed in Chapter 3, one another important factor are short cycles in the matrix.

4.4.2 Effects of Short Cycles

As a matter of fact, the design of an LDPC code is the design of the incidence matrix. From the principles of Bayesian theory [92, 93, 94, 95, 96], the iterative message propagation is able to deliver an optimum performance if the graph is cycle-free [97, 98, 99, 38]. However, due to the sequence length limitation, cycles are unavoidable in the factor graph. It has been known that the short cycles in the graph will lead to trapping sets for the iterative decoding. If an error occurs, it would be difficult to correct within the short cycles, especially for those length-4 cycles. For this reason, many algorithms have been proposed to avoid the short cycles for an LDPC code design. As the LDPC code always requires a long sequence, most of these algorithms are randomly generated by computers. Until now, many computer-based matrix design algorithms have been used to avoid short cycles inside the incidence matrix of an LDPC code. The original design techniques have been proposed by Gallager [9] and MacKay [100]. Later on, two computer-based design algorithms, the progressive-edge-growth (PEG) algorithm [73] and the approximate cycle extrinsic (ACE) message degree algorithm [101], become popular. In 1999, independent of Gallager's previous work, MacKay suggested a computer based design algorithm, in which the matrix is randomly generated while avoiding the overlapping of two columns. Afterwards, the PEG algorithm has been proposed which uses a greedy method to create a Tanner graph without short cycles. Then, noticing that the cycles containing all degree-2 VNs are suspicious to causing stopping sets during iterations, the ACE algorithm has been proposed to prevent these stopping sets in the design of LDPC code. In some approaches, the PEG algorithm and the ACE algorithm can also be combined.

The above mentioned code design algorithms all try to maximize the girth in the LDPC code. For an LDHC code, the matrix has both the coding and modulation part. The mentioned algorithms can also be applied to avoid short cycles in the LDHC incidence matrix. These cycles are not only the cycles inside the LDPC submatrix, but also the cycles across the two submatrices. The second kind of cycles are depicted by a dashed box in Fig. 4.6, which is to be avoided in LDHC code. In this thesis, the PEG algorithm is generalized to the design of LDHC code. The original PEG algorithm is performed in a greedy method. The goal is to construct the graph edge by edge. Each time, one

new edge is added to maximize the local girth. It has been found that the VNs with low degrees are more sensitive to errors. Hence, the edge is added from the VN with the lowest degree to the highest degree in the Tanner graph. For one variable node, the first edge is always connected to the PCN with the lowest degree that is currently available. Then the remaining edges are added to the PCN that will not form a closed cycle. If this kind of PCN is not available, then find the PCN that will form a cycle with the maximum cycle length. The edge connection continues to the next VN until all the edges of the current VN have been added.

In our system, the construction of the code is different in two ways. One difference is that the graph is combined of two parts, the edges between the MNs and the VNs and the edges between the PCNs and the VNs. The edges to the VNs are connected from two directions, but they will affect each other. The other difference is that the degree of the MNs are not constant, whereas the higher degrees are more susceptible to errors as they represent symbols with higher-order modulations. Based on these two observations, a modified PEG algorithm has been proposed in [83]. The proposed algorithm works as follows:

According to the system parameters, the number of nodes of each type along with their corresponding degrees are computed. Suppose d_s is used to denote the MN degree and d_p to denote the PCN degree. The graph starts from the upper part to the lower part.

1. Upper part

- (a) Sort all the MNs according to their degrees and find the MN with the highest degree d_s .

- (b) Add edges from 0 to $d_s - 1$.

If this is the first edge, add the edge to the VN with the minimum current connected edges.

Else, add the edge to the VN which will not form a closed cycle. If not possible, add it to the VN which will form a cycle with the longest length.

- (c) If all edges to the current MN have been added, go to the MN with the next

highest degree, follow step b) until all the edges between VNs and MNs are connected.

2. Lower part

(a) Sort all the PCNs according to their degrees and find the PCN with the lowest degree d_p .

(b) Add edges from 0 to $d_p - 1$

If this is the first edge, add the edge to the VN with the minimum current connected edges.

Else, add the edge to the VN which will not form a closed cycle. If not possible, add it to the VN which will form a cycle with the longest length. When comparing the cycle length, the cycle across the MNs are also taken into account.

(c) If all edges to the current PCN have been added, go to the PCN with the next lowest degree, following step b) until all the edges between VNs and PCNs are connected.

The above mentioned modified version of code design will be applied in the design of LDHC code in case of BICM-OFDM with bit loading.

4.5 Simulation Results

In this section, the performance of the above mentioned BICM-OFDM with superposition mapping with natural-labeling is tested by means of Monte Carlo simulations. The chosen channel is a Rayleigh fading channel with memory length $L = 10$. Each OFDM symbol has a length of 256, which means the corresponding number of subcarriers is also 256. The channel coefficients are constant within one OFDM symbol. From one OFDM symbol to the next one, the channel coefficients are randomly generated. As has been mentioned before, the chosen bit loading algorithm is the Chow algorithm. When the LDHC code is applied, the matrix is constructed following the modified PEG algorithm as has been introduced in Section 4.4.2. At the receiver side, similarly to the case of an AWGN

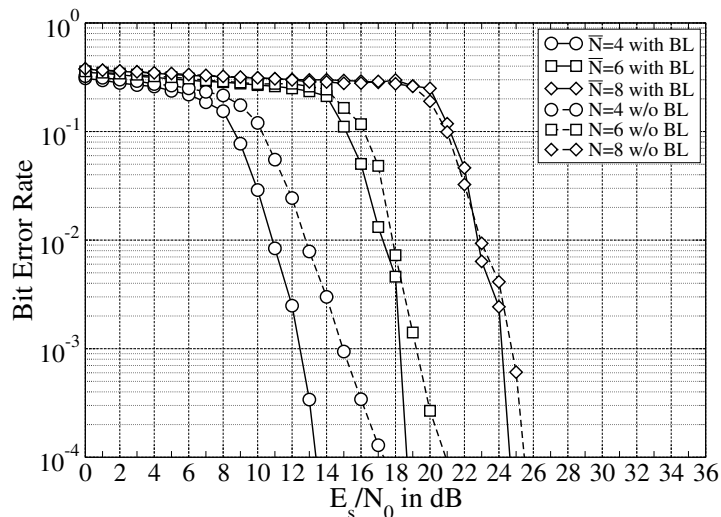


Figure 4.7: BER vs. SNR with and without bit loading.

channel, the iteration between the MNs and the VNs is set to be 20 and the local iteration between the VNs and PCNs is fixed to be 4.

4.5.1 Effects of Bit Loading

Now the performance of the coded transmission with and without bit loading are compared in Fig. 4.7. For simplicity, the combined channel code is a regular LDPC code with

$$d_{vp} = 2, d_p = 4, \quad (4.16)$$

where d_{vp} stands for the degree of the VNs w.r.t. the PCNs and d_p stands for the degree of the PCNs. The code rate is $R = 1/2$. As a coded modulation scheme, the average bandwidth efficiency \overline{BW} is computed by

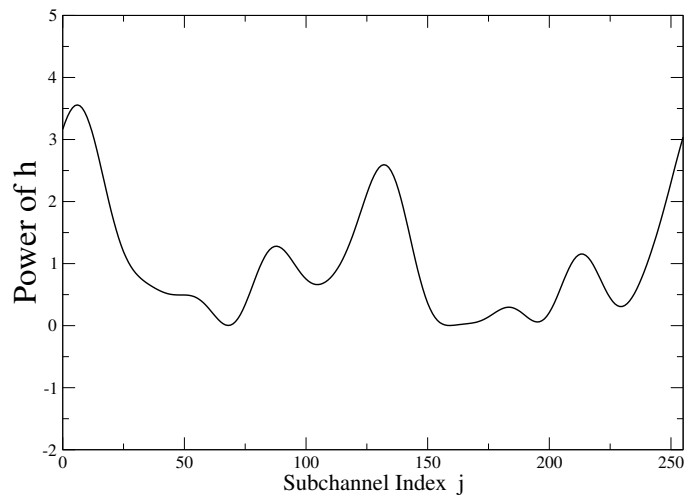
$$\overline{BW} = R\overline{N}. \quad (4.17)$$

Revisiting Fig. 4.1, the maximum achievable rates of the transmissions with bit loading are higher than the transmissions without bit loading for different bandwidth efficiencies, in uncoded cases. The gap between these two cases are significant when $\overline{N} = 4$ and $\overline{N} = 6$. However, this gap is not obvious in case of $\overline{N} = 8$. Such an observation is valid for both PSM and square QAM. The simulation results here agree with the above

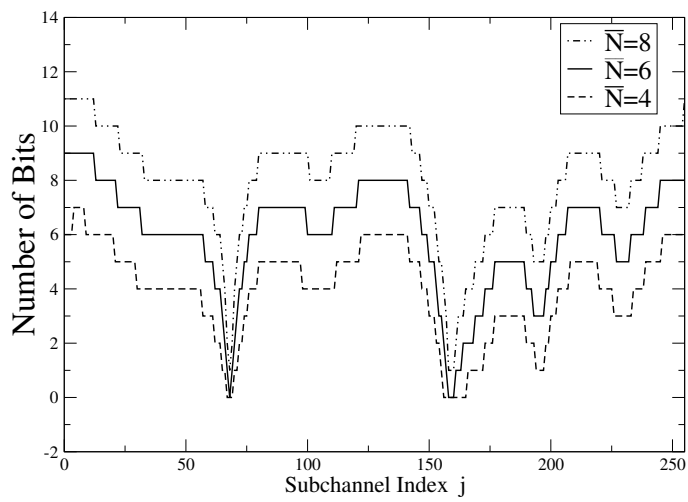
observations. For both cases of $\bar{N} = 4$ and $\bar{N} = 6$, the systems with bit loading deliver strong performance gain over the systems without bit loading. At the waterfall regions, the systems with bit loading drop much steeper than the systems without bit loading. However, in case of $\bar{N} = 8$, there is no obvious performance difference between the two systems. Apparently, if the bit loads are too high that its own capacity limit is exceeded, the effects of the bit loading will not provide any performance improvement. Due to this reason, the main focus should be put on $\bar{N} = 4$ ($\overline{BW} = 2$ bits/symbol), and $\bar{N} = 6$ ($\overline{BW} = 3$ bits/symbol).

4.5.2 The Bit Allocation

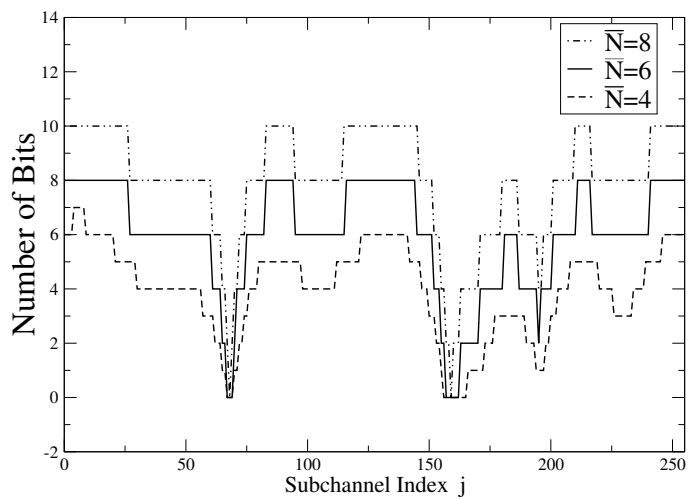
In this simulation, the channel coefficients and the bit allocations for one OFDM symbol are illustrated for example. The power of the channel coefficients in the subchannel is illustrated in Fig. 4.8(a). The number of the bits allocated to the subchannel determined by the bit loading algorithm is shown in Fig. 4.8(b) and Fig. 4.8(c). The step refers to the addition or the reduction of the bit unit. Due to some properties in the constellations, the orders of some modulation schemes, for example the square QAM, are preferably even numbers. Hence, the bit addition and reduction operations are performed at a step of two. In other modulation schemes, such as PSM, a bit allocation at the step of one is possible. This provides one more degree of freedom of choosing the modulation orders for PSM. On the other hand, in order to have a fair comparison between the two modulation schemes, the bit allocation with step equals two also needs to be considered. Fig. 4.8(b) and Fig. 4.8(c) provides the resulting bit allocations with a step size of one and a step size of two. The test SNR is 15 dB. In fact, given a fixed channel state information and the step size, the bit allocations are the same for all the SNRs because the subchannels only differ in the channel fading characteristics. In each of Fig. 4.8(b) and Fig. 4.8(c), the bit allocations for different \bar{N} are provided. The figures show that although the values are different, the bit allocations are changing at the same tendency. From $\bar{N} = 4$ to $\bar{N} = 6$, the number of the bits are increased by 2 at most of the subchannels except for those with no bit transmitted. The situation is the same from $\bar{N} = 6$ to $\bar{N} = 8$.



(a) Channel coefficients.



(b) Bit allocations, step = 1.



(c) Bit allocations, step = 2.

Figure 4.8: Channel coefficients and the bit allocations of subchannels.

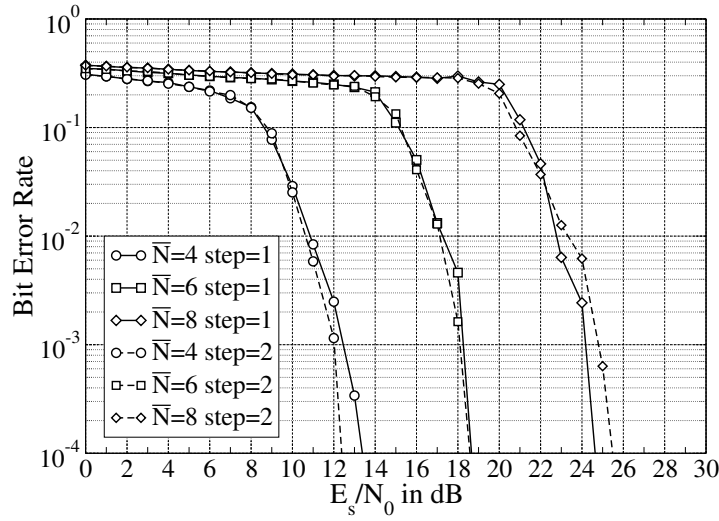


Figure 4.9: BER vs. SNR for different step sizes.

4.5.3 Effects of Step Size

Limited by the modulation schemes, the bit allocation step can be chosen between 1 and 2. The resulting bit allocations have been depicted in Fig. 4.8(b) and Fig. 4.8(c). For a step size of one, the number of modulation orders to be chosen is higher. The bit allocations can be selected so as to match for the subchannel capacities. However, the odd orders of PSM are more sensitive to errors, which degrade the performance. For a step size of two, the range of the orders are larger. This step size can have two problems. One is the possible chosen number of bits are farther from the actual subchannel capacities. The other is extremely high modulation orders can cause a tremendous computational complexity. These two effects also lead to the performance difference.

The corresponding BER performance is shown in Fig. 4.9. Again, the chosen encoder is an LDPC encoder with variable node degree of two and the parity-check node degree of four. The results show that the impact of the step size is not very strong on the system performance. When $\bar{N} = 4$, the small step size is slightly better than the system with a step size of two. However, for $\bar{N} = 8$, the systems with a step size of two are worse than the cases with a step size of one. This is due to the very high modulation orders. Since the performance difference of different step sizes is not very large, the step size of two is chosen for the bit loading algorithms, for both PSM and square QAM.

4.5.4 Effects of Degree Allocation

An LDHC code is composed of two parts: the modulation with the repetition code part, and the LDPC code part. The overall data rate can thus be separated into two parts: the rate for modulation with repetition code R_s , and the parity-check part R_p . Given a fixed bandwidth efficiency, how the overall rate is divided between these two parts is determined by the so-called degree allocation. In these simulations, the overall code rate $R_s R_p$ is fixed to be $1/2$. The performance is compared with different combinations of degree allocations for different \overline{N} . The parameters of the degree distributions are given in equations from (3.47) to (3.50). For the AWGN channel, it has been shown that a pure LDPC code is not a perfect match for superposition mapping. In order to circumvent the non-bijective properties of superposition mapping, a concatenated code composed of both the repetition code and the parity-check code is necessary. The LDPC code enables the BER curves to fast converge at a high enough SNR. On the other hand, the repetition code can make the BER curve start to converge early, but the curve is going down very slowly at the waterfall region. As two different types of codes have their own pros and cons. Hopefully, a concatenated code can be found to maintain both of their advantages while avoiding both of their disadvantages.

Fig. 4.10(a) provides the systems all with the average bandwidth efficiency $\overline{BW} = 2$. The degree allocations are different. Starting from a pure rate $1/2$ LDPC code, the fraction of the repetition code part is increasing until a pure rate $1/2$ repetition code. It can be seen that with the increase of the repetition part, the performance is decreasing, which shows that the repetition code will not be helpful for low modulation orders. When $\overline{BW} = 3$ as is illustrated in Fig. 4.10(b), the tendency is different. When R_s is decreased from $2/3$ to $5/8$, the BER starts to converge earlier, and it is also able to converge fast. By further reducing R_s to $4/7$, the performance is still improving. The system converges very slowly only when the combined channel coding is a pure repetition code. For $\overline{BW} = 4$ in Fig. 4.10(c), a more than 1.8 dB gain can be observed by reducing R_s from $2/3$ to $5/8$. Further reducing R_s , the performance gain just becomes smaller. Again, a rate $1/2$ regular repetition code fails to converge. From these results, the increasing parts of the repetition code while maintaining the LDPC code part, does improve the system performance, for

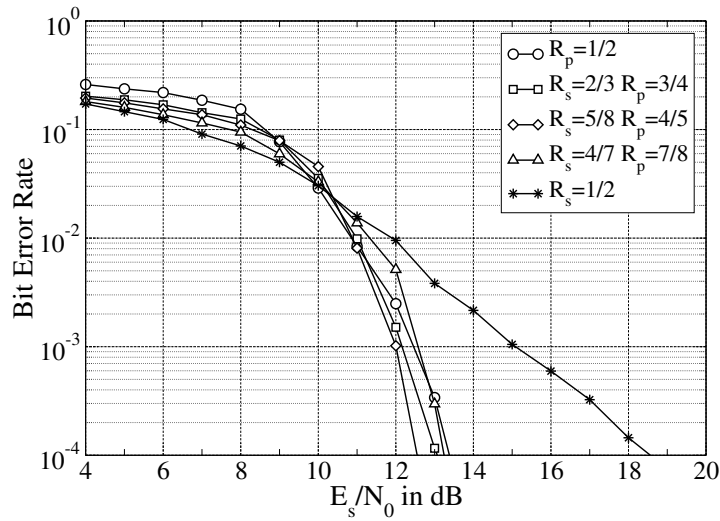
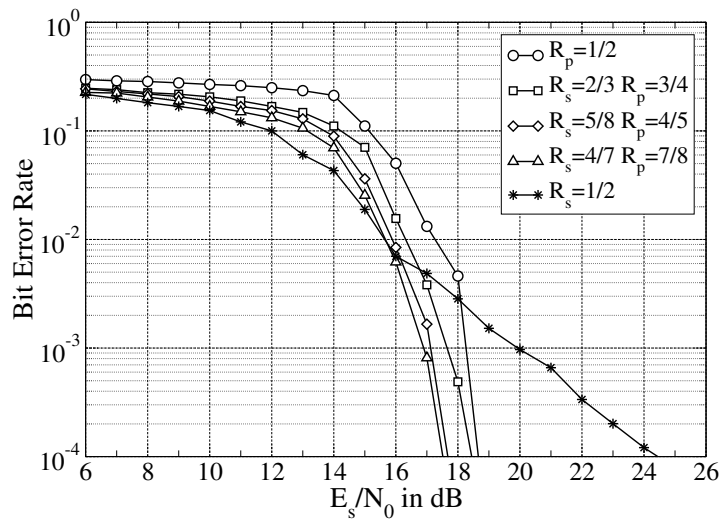
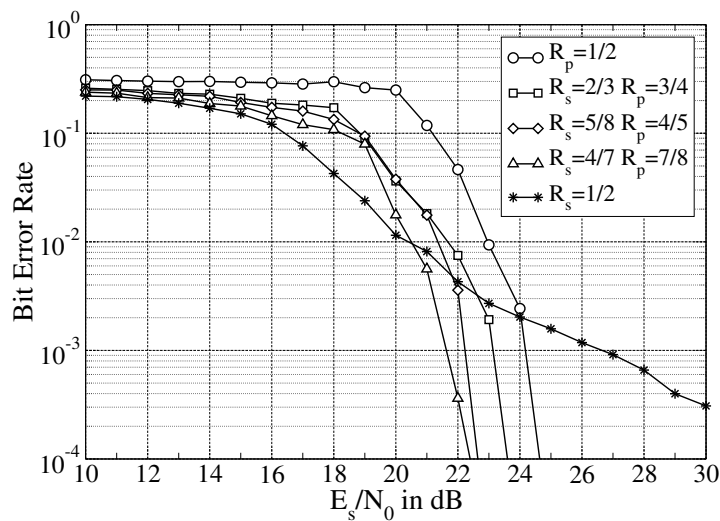
(a) $\bar{N} = 4$.(b) $\bar{N} = 6$.(c) $\bar{N} = 8$.

Figure 4.10: BER vs. SNR, different degree allocations.

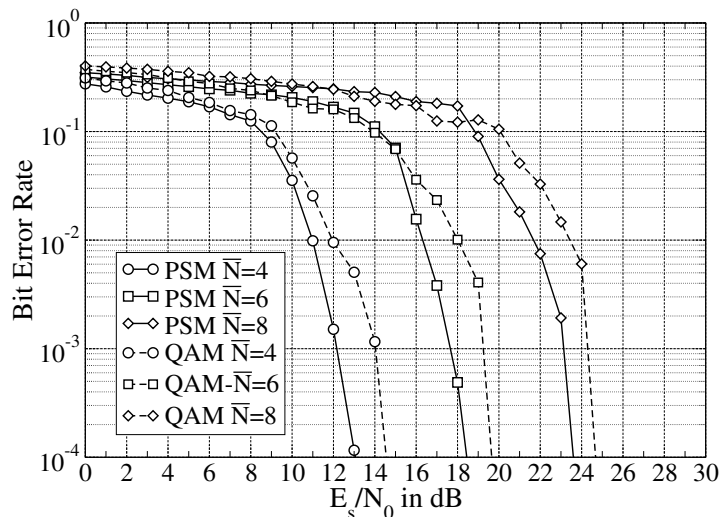


Figure 4.11: LDPC-QAM vs. LDHC-PSM

higher-order modulations such as $\bar{N} = 6$ and $\bar{N} = 8$, which agrees with our previous results in case of AWGN. Nevertheless, given a fixed overall code rate, increasing the repetition code part means the degrees of the parity-check nodes will also be enhanced. The higher the parity-check node degrees, the more difficulty in designing the code. Meanwhile, the complexity of the encoding and decoding will also grow significantly. Besides, the performance improvement is minor when the repetition part reaches a certain level. To maintain a trade-off between the performance and the complexity, the combination of $R_s = 2/3$ and $R_p = 3/4$ is chosen in the following simulations.

4.5.5 LDHC-PSM-OFDM vs. LDPC-QAM-OFDM

In Fig. 4.11, we compare the proposed LDHC-PSM with a conventional scheme. In modern transmission systems, it is well-known that square QAM is used as the standard modulation scheme and LDPC codes show desirable performance when combining with Gray-labeled square QAM for AWGN channels. The reason behind it is that both of their transfer curves match each other in the EXIT chart. For a comparison with our proposed combination, a rate $1/2$ LDPC code with $d_{vp} = 2$ and $d_p = 4$ is connected to square QAM with Gray labeling. As for PSM, we choose a combination of code with a degree allocation $R_s = 2/3$ and $R_p = 3/4$. Both systems use the Chow algorithm

for bit loading. From BER performance, it is clearly shown that the combination of LDHC code and PSM outperforms square QAM with LDPC code in terms of BICM-OFDM with adaptive bit loading. For different bandwidth efficiencies, the performance of LDHC-PSM-OFDM always delivers a 1 dB gain over LDPC-QAM-OFDM.

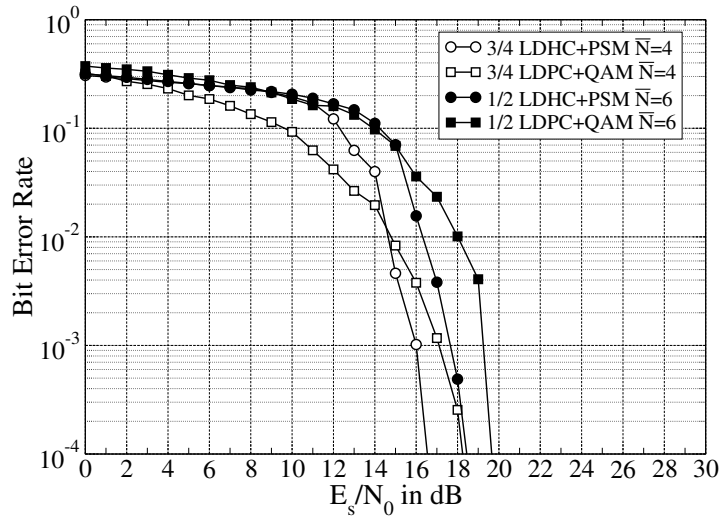
4.5.6 Bandwidth Efficiency

The previous performance comparisons are among systems with the same code rate $1/2$. Given a fixed bandwidth efficiency, there are also different combinations of code rate and modulation order. For example, 3 bits/symbol could be obtained by a rate $1/2$ code with a modulation order of $\bar{N} = 6$, and it could also be a rate $3/4$ code with a modulation order $\bar{N} = 4$. In this part of the simulation, systems with a same bandwidth efficiency are compared. In Fig. 4.12, the performance of these systems is compared at bandwidth efficiencies of 3 bits/symbol and 4 bits/symbol. The LDPC codes are connected with Gray-labeled square QAM and the LDHC codes are connected with natural-labeled PSM.

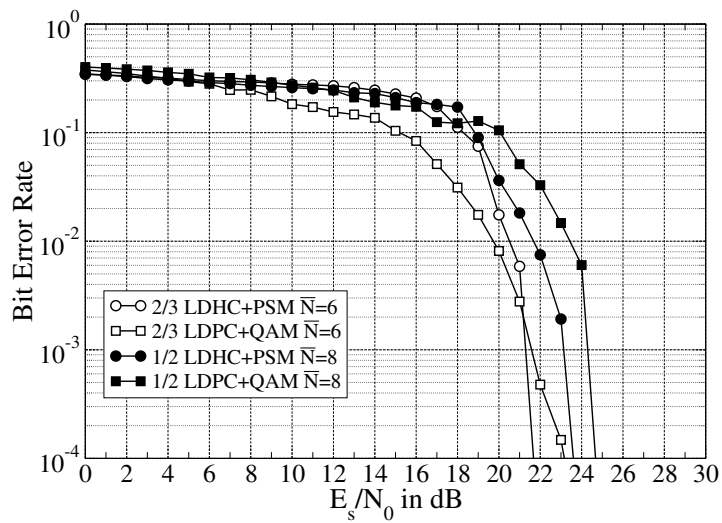
For $\overline{BW} = 3$ bits/symbol, two combinations of coding and modulation are compared. The first scheme is a rate $1/2$ code with a $\bar{N} = 6$ modulation. In this case, the code parameters are the same as in Section 4.5.5. The second scheme is a rate $3/4$ code with a $\bar{N} = 4$ modulation. Thus, the code parameter of the regular $3/4$ code is

$$d_{vp} = 2, d_p = 8, \quad (4.18)$$

for both modulations. It can be observed from Fig. 4.12(a), that the second scheme outperforms the first scheme for both QAM and PSM. This indicates that in OFDM systems with adaptive bit loading, the lower-order modulation cases are less sensitive to communication errors. Concerning the modulation in the second scheme, the curve of square QAM starts to converge earlier than that of PSM, but also goes down very slowly. This makes PSM a better choice.



(a) $\overline{BW} = 3$ bits/symbol.



(b) $\overline{BW} = 4$ bits/symbol.

Figure 4.12: BER vs. SNR, different coded modulation combinations.

For $\overline{BW} = 4$ bits/symbol, the similar performance is compared in Fig. 4.12(b). Here, the two schemes are chosen as follows. The first scheme is a rate 1/2 code with a $\overline{N} = 8$ modulation. The code parameters are also the same as in Section 4.5.5. The second scheme is a rate 2/3 code with a $\overline{N} = 6$ modulation, with the code parameters

$$d_{vp} = 2, d_p = 6, \quad (4.19)$$

for both modulations. The results are similar to that for the bandwidth efficiency of 3 bits/symbol.

4.6 Summary

In this chapter, we have studied superposition mapping in frequency-selective fading channels. By applying OFDM, the channel is equivalent to a number of frequency-flat fading subchannels. The theoretical benefit of using superposition mapping has been pointed out. It has also been mentioned that the subchannel diversity can be further exploited via water-filling. Hence, a structure of BICM-OFDM with adaptive bit loading is shown. An implementation of LDHC code to this structure has been proposed. The incidence matrix design is conducted by a modified PEG algorithm. From the simulation results, a concatenated code is again suitable for this system.

Chapter 5

Superposition Mapping for MIMO

Bit-interleaved coded modulation (BICM) has shown to be a pragmatic transmission scheme. Superposition mapping outperforms square QAM in single antenna systems, as has been proved in the previous chapters. These advantages motivate an extension of superposition mapped BICM transmission to multiple-input multiple-output (MIMO) systems. In fading conditions, utilizing multiple transmit antennas and multiple receive antennas provides an additional gain in the spatial domain. This spatial gain can be obtained via spatial multiplexing the independent data streams. By doing so, the channel capacity is increased.

Considering the receiver side, the received signal stream is composed of the signal streams from all the transmit antennas and the AWGN samples. This combination resembles the formation of a superposition mapped symbol. The code design strategy which fits for superposition mapping can also be applied herein.

In this chapter, main focus will be on the spatial multiplexing over frequency-flat fading MIMO channels. An extension to the frequency-selective MIMO channel is quite straightforward. Correspondingly, the LDHC code can be further applied by taking the spatial domain into account.

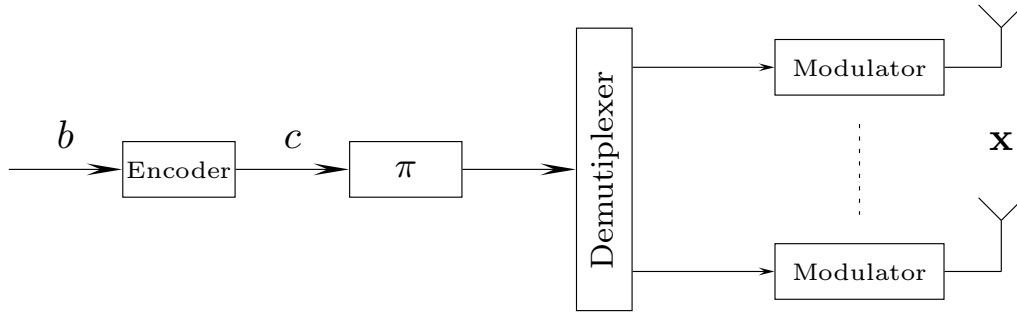


Figure 5.1: Block diagram of a BICM-MIMO transmitter.

5.1 MIMO Transmission Model

A general block diagram of a BICM-MIMO transmitter is illustrated in Fig. 5.1. Randomly generated info bits are encoded using an error-correcting encoder, and then passed through a bit-wise interleaver π . The coded and interleaved bits are demultiplexed onto N_T layers. On each layer, the bits are mapped onto a complex valued symbol via a certain modulation scheme before sending out. At the time index k , the transmitted symbol vector is defined as $\mathbf{x}[k] := (x_0[k], x_1[k], \dots, x_{N_T-1}[k])^T$. All the elements in \mathbf{x} belong to a certain symbol alphabet \mathcal{X} with size $M = 2^N$. Thus, N is the modulation order and \mathcal{X} is determined by the chosen modulation scheme. With normalization, the total average transmit power is defined as

$$E_s = E\{|\mathbf{x}[k]|^2\} = N_T. \quad (5.1)$$

Let us consider a MIMO channel model with N_T transmit antennas and N_R receive antennas. Assuming the received vector to be $\mathbf{y}[k] := (y_0[k], y_1[k], \dots, y_{N_R-1}[k])^T$, the channel relationship at time index k is provided by

$$\mathbf{y}[k] = \mathbf{H}[k]\mathbf{x}[k] + \mathbf{z}[k]. \quad (5.2)$$

The noise samples $\mathbf{z}[k] := (z_0[k], z_1[k], \dots, z_{N_R-1}[k])^T$ are zero mean Gaussian distributed with variance σ_z^2 . $\mathbf{H}[k]$ is a $N_R \times N_T$ channel matrix of the form

$$\mathbf{H}[k] = \begin{pmatrix} h_{00}[k] & h_{01}[k] & \cdots & h_{0N_T}[k] \\ h_{10}[k] & h_{11}[k] & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_R0}[k] & \cdots & h_{N_RN_T}[k] \end{pmatrix}. \quad (5.3)$$

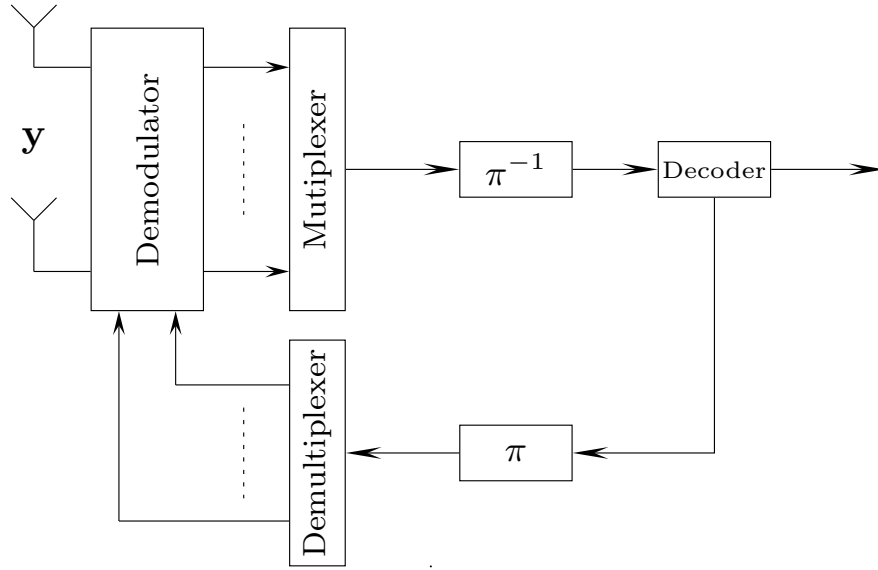


Figure 5.2: Block diagram of a BICM-MIMO receiver.

The receiver structure of BICM-MIMO is illustrated in Fig. 5.2. At the receiver side, the demodulator uses a certain demapping algorithm to compute the initial LLR values based on the received vector \mathbf{y} and the channel matrix \mathbf{H} . Now the time index k is dropped for simplification. Assume c_n is the n th code bit of one complex symbol. In case of non-iterative processing, the LLR of the code bit c_n computed by soft maximum likelihood detection is given by

$$\text{LLR}_n = \ln \frac{p(\mathbf{y}|c_n = 0, \mathbf{H})}{p(\mathbf{y}|c_n = 1, \mathbf{H})}, \quad (5.4)$$

where $p(\mathbf{y}|c_n, \mathbf{H})$ is the transition probability of the received symbol vector \mathbf{y} given c_n and \mathbf{H} . As there is no a priori information, (5.4) can be further written as

$$\text{LLR}_n = \ln \frac{p(\mathbf{y}|c_n = 0, \mathbf{H})}{p(\mathbf{y}|c_n = 1, \mathbf{H})} = \ln \frac{\sum_{x \in \mathcal{X}_n^0} \exp\left(-\frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\sigma_z^2}\right)}{\sum_{x \in \mathcal{X}_n^1} \exp\left(-\frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\sigma_z^2}\right)}, \quad (5.5)$$

where \mathcal{X}_n^0 and \mathcal{X}_n^1 denote the symbol sets where the code bit is $c_n = 0$ and $c_n = 1$, respectively. Then the obtained extrinsic LLR values are multiplexed, deinterleaved and passed to the decoder. The decoder tries to correct the transmission errors and to reconstruct the original info bits.

To improve the detection quality, iterative processing is almost always applied. In case of an iterative receiver, the decoder computes and outputs the extrinsic LLR values of the

code bits. These extrinsic LLR values are then interleaved, demultiplexed, and fed back to the demodulator. Now that they are treated as a priori information for the demodulator. Based on the relationship between the a priori LLR and the a priori probabilities

$$\text{LLR}_n = \frac{P(c_n = 0)}{P(c_n = 1)}, \quad (5.6)$$

the a priori probability of $c_n = 0$ and $c_n = 1$ is computed by

$$P(c_n = 0) = \frac{e^{\text{LLR}_n}}{1 + e^{\text{LLR}_n}}, \quad P(c_n = 1) = \frac{1}{1 + e^{\text{LLR}_n}}. \quad (5.7)$$

Based on the a priori probabilities, the received symbol vector \mathbf{y} , and the channel matrix \mathbf{H} , the demodulator can use the APP algorithm to compute the a posteriori information of the code bits and output their extrinsic LLRs, according to

$$\text{LLR}_n = \ln \frac{p(\mathbf{y}|c_n = 0, \mathbf{H})}{p(\mathbf{y}|c_n = 1, \mathbf{H})} = \ln \frac{\sum_{x \in \mathcal{X}_n^0} \exp\left(-\frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\sigma_z^2}\right) \prod_{j=0, j \neq n}^{N-1} P(c_j)}{\sum_{x \in \mathcal{X}_n^1} \exp\left(-\frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\sigma_z^2}\right) \prod_{j=0, j \neq n}^{N-1} P(c_j)}. \quad (5.8)$$

Then these LLR values are re-multiplexed, re-deinterleaved, and passed to the decoder. For the decoder, they serve as a priori information. This completes one iteration. The iteration is carried on until certain preset conditions are satisfied. The APP detection algorithm is optimum in terms of performance. However, its computational complexity is growing exponentially with NN_T , which is not affordable for systems with high modulation orders and large number of transmit antennas. To mitigate the complexity, a few low complexity detection algorithms have been proposed, which will be elaborated in the later sections.

5.2 MIMO Channel Capacity

Assume the channel state information is known at the receiver side. We consider two definitions of channel capacity, the ergodic capacity and the outage capacity.

5.2.1 Ergodic Capacity

The capacity derived by Shannon is also termed ergodic capacity. Here, we consider it in MIMO. Assuming an equal power allocation on each antenna, the capacity of a fast fading channel is

$$C = \max_{p(\mathbf{x})} I(\mathbf{x}; \mathbf{y}, \mathbf{H}) . \quad (5.9)$$

Since the channel fading is independent of the channel input,

$$I(\mathbf{x}; \mathbf{H}) = 0 . \quad (5.10)$$

From the chain rule of mutual information,

$$I(\mathbf{x}; \mathbf{y}, \mathbf{H}) = I(\mathbf{x}; \mathbf{H}) + I(\mathbf{x}; \mathbf{y}|\mathbf{H}) = I(\mathbf{x}; \mathbf{y}|\mathbf{H}) . \quad (5.11)$$

$I(\mathbf{x}; \mathbf{y}, \mathbf{H})$ can be calculated by the conditional mutual information $I(\mathbf{x}; \mathbf{y}|\mathbf{H})$, with the following relationship for a certain channel realization [102]:

$$\begin{aligned} I(\mathbf{x}; \mathbf{y}|\mathbf{H}) &= h(\mathbf{y}|\mathbf{H}) - h(\mathbf{y}|\mathbf{x}, \mathbf{H}) \\ &= h(\mathbf{y}|\mathbf{H}) - h(\mathbf{z}) \\ &= h(\mathbf{y}|\mathbf{H}) - N_R \log(\pi e \sigma_z^2) . \end{aligned} \quad (5.12)$$

For a complex vector, if all the variables are circular-symmetric jointly Gaussian variables, the entropy of the vector is maximized [1]. The corresponding maximum entropy value for \mathbf{y} is

$$\max h(\mathbf{y}) = \log(\det(\pi e \mathbf{K}_y)) , \quad (5.13)$$

where \mathbf{K}_y is the covariance of \mathbf{y} , which for a normalized power of \mathbf{x} is given by

$$\mathbf{K}_y = \sigma_z^2 \mathbf{I}_{N_R} + \frac{E_s}{N_T} \mathbf{H} \mathbf{H}^* . \quad (5.14)$$

\mathbf{H}^* denotes the transpose conjugate operation of \mathbf{H} . Upon substituting (5.13) and (5.14) into (5.12), we get

$$\begin{aligned} \max_{p(\mathbf{x})} I(\mathbf{x}; \mathbf{y}|\mathbf{H}) &= \log \left((\pi e)^{N_R} \det \left(\sigma_z^2 \mathbf{I}_{N_R} + \frac{1}{N_T} \mathbf{H} \mathbf{H}^* \right) \right) - N_R \log(\pi e \sigma_z^2) \\ &= \log \det \left(\mathbf{I}_{N_R} + \frac{E_s}{N_T \sigma_z^2} \mathbf{H} \mathbf{H}^* \right) \\ &= \log \det \left(\mathbf{I}_{N_R} + \frac{\text{SNR}}{N_T} \mathbf{H} \mathbf{H}^* \right) . \end{aligned} \quad (5.15)$$

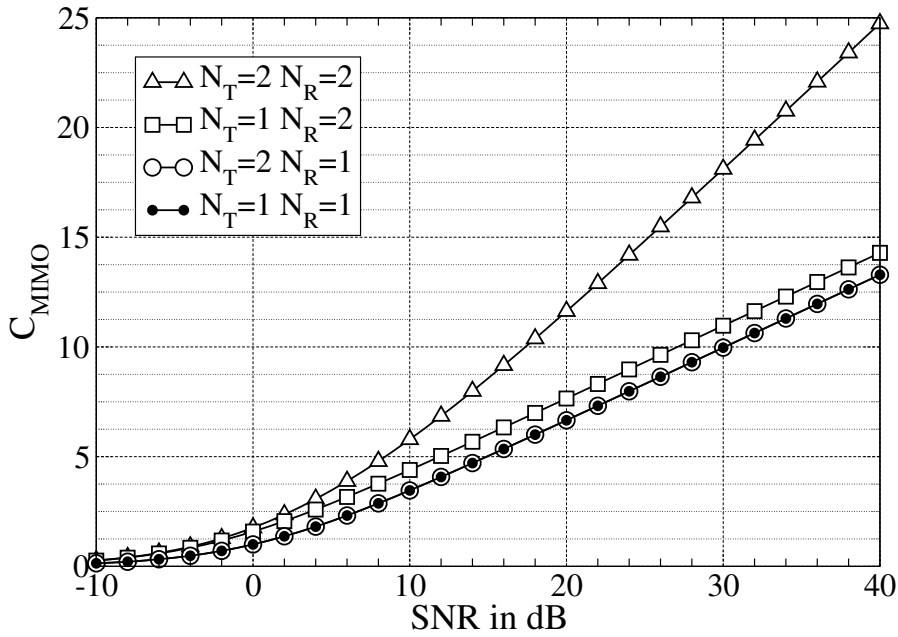


Figure 5.3: Capacity vs. SNR for different number of antennas.

Finally, the capacity of the fast fading channel is obtained by averaging $I(\mathbf{x}; \mathbf{y}|\mathbf{H})$:

$$C_{\text{MIMO}} = E_{\mathbf{H}} \left\{ \log \det \left(\mathbf{I}_{N_R} + \frac{\text{SNR}}{N_T} \mathbf{H}\mathbf{H}^* \right) \right\}. \quad (5.16)$$

Accordingly, the capacity curves with different number of transmit and receive antennas are illustrated in Fig. 5.3. As the curves show, only increasing the number of transmit antennas will not increase the capacity since the curves “ $N_T = 1, N_R = 1$ ” and “ $N_T = 2, N_R = 1$ ” overlap each other. Only increasing the number of receive antennas also enhances the capacity, but within a limited range. Only when both the transmit and receive antennas are increased simultaneously, a significant capacity increment can be observed.

Error Probability

The capacities in Fig. 5.3 are the maximum achievable information rates for a perfect transmission. It tells that the transmission rate must be lower or equal to the capacity so that the transmission is error-free. In the simulations, given a fixed transmission rate, the bit error ratios are computed for different SNRs. The capacity can only tell the minimum required SNR to achieve a quasi error-free transmission. For a certain transmission rate, if

the current SNR is lower than the minimum required SNR, transmission errors will occur. Thus, the bit error probability vs. SNR is always used to evaluate the transmission quality. Using the capacity function in (5.15) along with the rate distortion theory, the minimum required SNR can be computed given a finite error probability.

Consider binary transmission. The input source symbols are independent, identically distributed (i.i.d.) random variables with a Bernoulli($\frac{1}{2}$) distribution. The rate distortion function $R(D)$ of a Bernoulli($\frac{1}{2}$) distribution is given by

$$R(D) = \begin{cases} 1 - h(D), & 0 \leq D \leq \frac{1}{2} \\ 0, & \frac{1}{2} < D < 1 \end{cases}, \quad (5.17)$$

where D is the distortion. The rate distortion gives the minimum rate of the source code. Assume the number of info bits per symbol is R_t . According to information theory, if a perfect transmission is required,

$$C \geq R(D)R_t \quad (5.18)$$

must be fulfilled. Here the bit error probability P_e is equivalent to the distortion D . The bit error probability in the range $0 < P_e < \frac{1}{2}$ is of major concern. In case of MIMO coded modulation, the capacity C is obtained from (5.15). The overall rate is given by

$$R_t = RNN_T, \quad (5.19)$$

where R is the code rate and N is the modulation order. Hence, we have

$$E_{\mathbf{H}} \left\{ \log \det \left(\mathbf{I}_{N_R} + \frac{\text{SNR}}{N_T} \mathbf{H}\mathbf{H}^* \right) \right\} \geq (1 - h(P_e))RNN_T. \quad (5.20)$$

This equation gives the minimum required SNR for a finite bit error probability. On the other hand, we can also compute the minimum error probability given a fixed rate and SNR. After a few manipulations, (5.20) becomes

$$h(P_e) \geq 1 - \frac{E_{\mathbf{H}} \left\{ \log \det \left(\mathbf{I}_{N_R} + \frac{\text{SNR}}{N_T} \mathbf{H}\mathbf{H}^* \right) \right\}}{RNN_T}. \quad (5.21)$$

$h(P_e)$ is the entropy of P_e , which is computed by

$$h(P_e) = -P_e \log(P_e) - (1 - P_e) \log(1 - P_e). \quad (5.22)$$

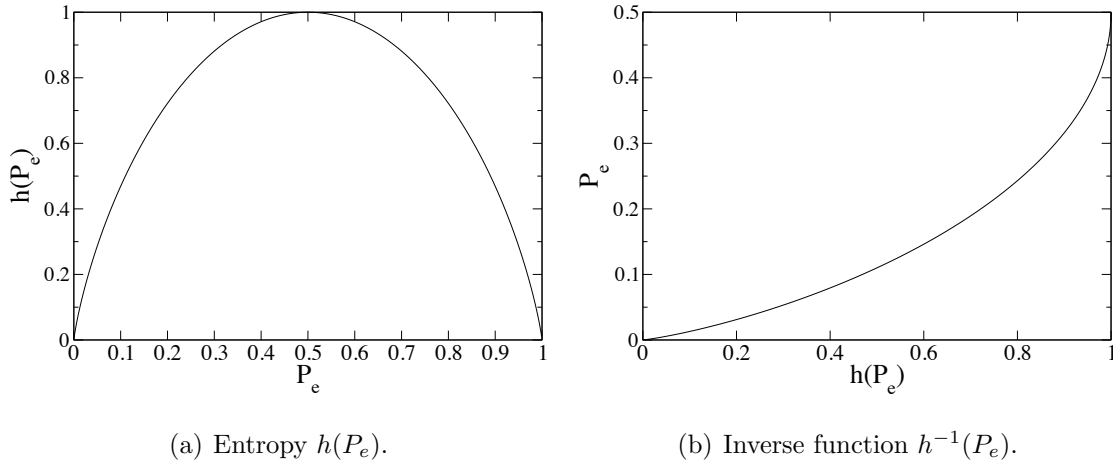


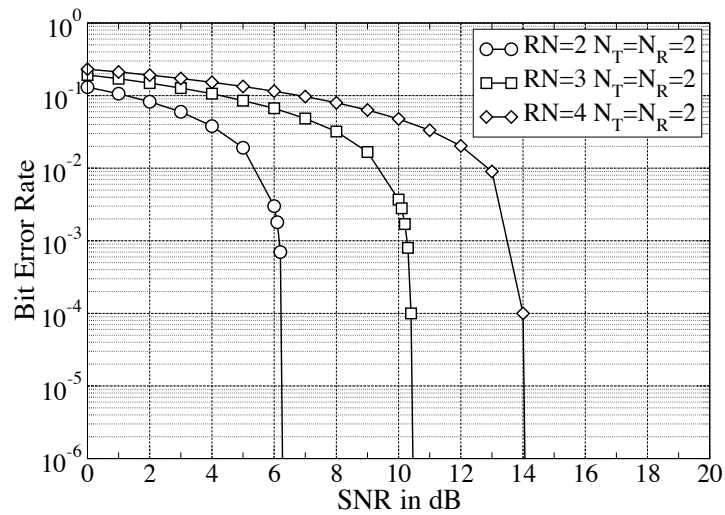
Figure 5.4: Entropy function and its inverse function.

The entropy function is illustrated in Fig. 5.4(a). For P_e in the range of $0 < P_e < 1$, the entropy function is not invertible, as P_e is not unique for a certain P_e . However, if we only take P_e in the range from 0 to $1/2$, this function is invertible. Since $h(P_e)$ is monotonically increasing for $0 < P_e < \frac{1}{2}$, we can obtain the inverse function via a look-up table, as drawn in Fig.5.4(b). Using the inverse function, (5.21) will turn into

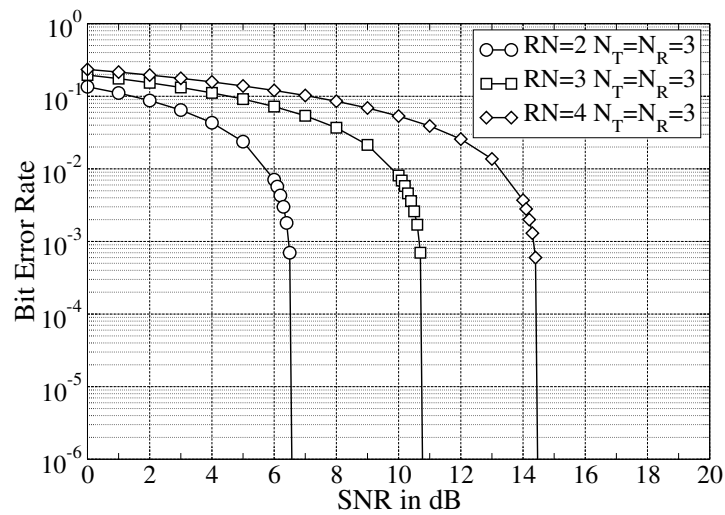
$$P_e \geq h^{-1} \left(1 - \frac{E_{\mathbf{H}} \left\{ \log \det \left(\mathbf{I}_{N_R} + \frac{\text{SNR}}{N_T} \mathbf{H}\mathbf{H}^* \right) \right\}}{RN N_T} \right). \quad (5.23)$$

This relationship provides lower bounds of BER curves for different SNRs. Using this formula, lower bounds on the BER for different rates are illustrated in Fig. 5.5.

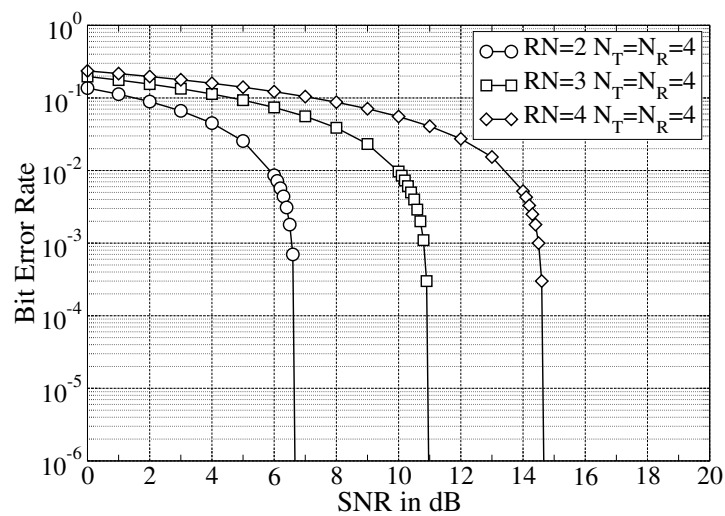
The figures show the capacities of MIMO systems with different transmit and receive antennas, respectively. The numbers of both antennas are changed simultaneously. However, the bit error probability bounds do not change much. If spatial multiplexing is applied, there is no doubt that when increasing the transmit antenna number, the bandwidth efficiency is enhanced without much performance degradation, assuming an equal receive antenna number is increased. The only factor that affects the performance is the overall code rate RN .



(a) $N_T = N_R = 2$.



(b) $N_T = N_R = 3$.



(c) $N_T = N_R = 4$.

Figure 5.5: BER lower bounds for different RN .

Mutual Information

The above discussed capacities are the theoretically achievable number of info bits per symbol. These capacities can be achieved if and only if the channel output samples are all Gaussian variables with the same mean and variance. In real transmissions, this condition is difficult to be fulfilled as the transmit symbols are taken from a finite symbol alphabet. Hence, the average mutual information between the transmit symbols \mathbf{x} and the received symbols \mathbf{y} is applied to measure the achievable number of bits per symbol use. For BICM-MIMO transmission, the mutual information given a particular channel realization is of the form [12, 103]

$$I(\mathbf{x}; \mathbf{y}|\mathbf{H}) = h(\mathbf{x}|\mathbf{H}) - h(\mathbf{x}|\mathbf{y}, \mathbf{H}) . \quad (5.24)$$

Since \mathbf{x} is independent of \mathbf{H} , for any bijective modulation scheme, such as square QAM or PSK, the entropy of the transmit symbols is given by

$$h(\mathbf{x}|\mathbf{H}) = h(\mathbf{x}) = N_T N . \quad (5.25)$$

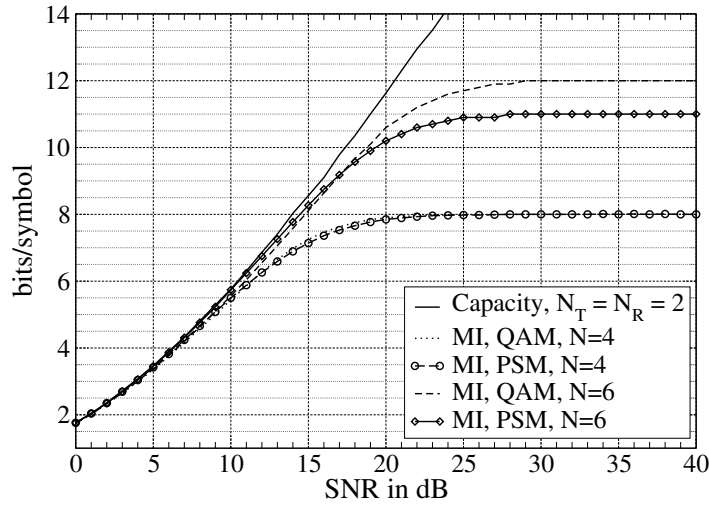
The conditional entropy is obtained from

$$h(\mathbf{y}|\mathbf{x}, \mathbf{H}) = E_{\mathbf{x}, \mathbf{y}} \left\{ \log \frac{\sum_{\mathbf{x} \in \mathcal{X}} p(\mathbf{y}|\mathbf{x}, \mathbf{H})}{p(\mathbf{y}|\mathbf{x}, \mathbf{H})} \right\} , \quad (5.26)$$

where $E\{\cdot\}$ denotes the expectation operation with respect to \mathbf{x} and \mathbf{y} and $p(\mathbf{y}|\mathbf{x}, \mathbf{H})$ is the transition probability:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{H}) = \frac{1}{(\pi\sigma_z^2)^{N_R}} \exp \left(-\frac{\|\mathbf{y} - \mathbf{x}\|^2}{\sigma_z^2} \right) . \quad (5.27)$$

This way of calculating the mutual information has been widely used in evaluating the capacity of BICM-MIMO systems. However, this method is based on one assumption: the applied modulation schemes are bijective. In our discussion, the superposition mapping is sometimes a non-bijective mapping. In the following, another way of computing the mutual information will be derived.

Figure 5.6: Mutual information, $N_T = N_R = 2$.

Revisiting (5.24), the mutual information can also be written by

$$\begin{aligned}
 I(\mathbf{x}; \mathbf{y}|\mathbf{H}) &= h(\mathbf{y}|\mathbf{H}) - h(\mathbf{y}|\mathbf{x}, \mathbf{H}) \\
 &= h(\mathbf{y}|\mathbf{H}) - h(\mathbf{z}) \\
 &= h(\mathbf{y}|\mathbf{H}) - N_R \log(\pi e \sigma_z^2). \tag{5.28}
 \end{aligned}$$

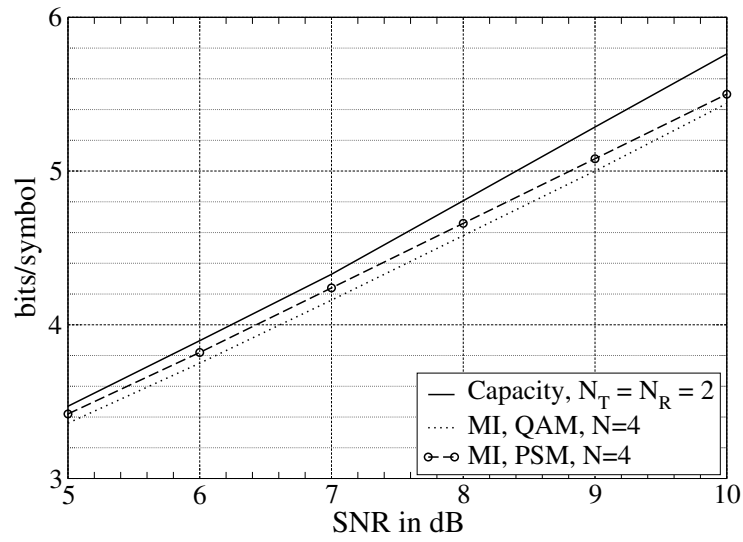
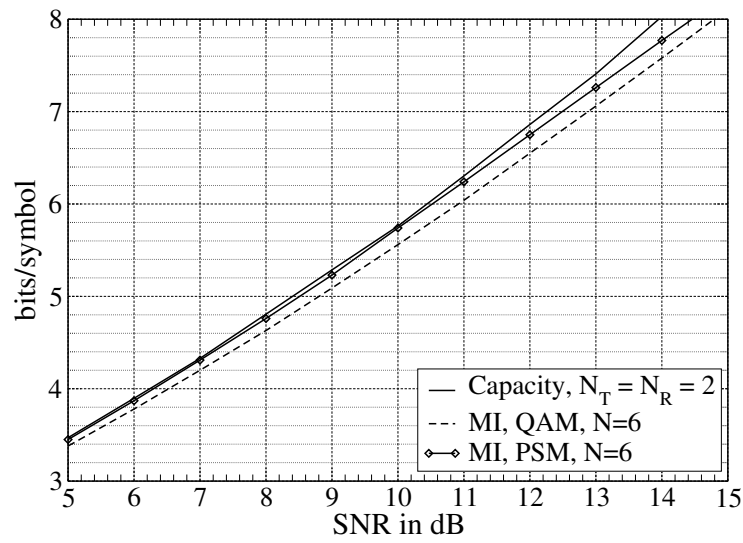
The differential entropy of the received vector, according to the definition, is given by

$$h(\mathbf{y}|\mathbf{H}) = - \int p(\mathbf{y}|\mathbf{H}) \log(p(\mathbf{y}|\mathbf{H})) . \tag{5.29}$$

The conditional probability density function of \mathbf{y} is obtained by

$$\begin{aligned}
 p(\mathbf{y}|\mathbf{H}) &= \sum_{\mathbf{x} \in \mathcal{X}} P(\mathbf{x}) p(\mathbf{y}|\mathbf{x}, \mathbf{H}) \\
 &= \sum_{\mathbf{x} \in \mathcal{X}} P(\mathbf{x}) \frac{1}{(\pi \sigma_z^2)^{N_R}} \exp\left(-\frac{\|\mathbf{y} - \mathbf{x}\|^2}{\sigma_z^2}\right) . \tag{5.30}
 \end{aligned}$$

The resulting conditional pdf is a summation of many Gaussian distributions with different means, which can be obtained via simulations. This way of computation is regardless of the bijectivity of the modulation scheme. Thus, in case of superposition mapping, this way of mutual information computation will always be applied. In the following, the curves using this method will be displayed.

(a) $N = 4$.(b) $N = 6$.Figure 5.7: Mutual information, a closer look, $N_T = N_R = 2$.

	bits/symbol	C	MI _{PSM}	MI _{QAM}
$N = 4$	4	6.25	6.40	6.60
	5	8.40	8.80	9.00
	6	10.40	11.30	11.30
$N = 6$	4	6.25	6.30	6.50
	5	8.40	8.50	8.80
	6	10.40	10.50	10.90

Table 5.1: SNR (in dB) comparison.

In this section, we focus on comparing the achievable rates for different modulation schemes in case of multiple antennas. In Fig. 5.6, the curve “capacity” is drawn following (5.15), and all the curves “MI” (mutual information) are obtained by (5.28). The results shown in Fig. 5.6 are different from those in the corresponding single antennas case. Considering $N = 4$, in other words 16-QAM and 16-PSM, the two “MI” curves are very close to each other. A zoom-in figure with SNR in the range of $5 \text{ dB} < \text{SNR} < 10 \text{ dB}$ (c.f. Fig. 5.7(a)) shows, both curves have gaps to the capacity curve. At rate 5 bits/symbol, the gap between “capacity” and “MI, PSM” is 0.4 dB, while the gap between “capacity” and “MI, QAM” is 0.6 dB. 16-PSM provides a 0.2 dB gain over 16-QAM. When N is increased to 6, which is for 64-QAM and 64-PSM, the gap between “MI, QAM” and “MI, PSM” is relatively large, as can be observed by a zoom-in between $5 \text{ dB} < \text{SNR} < 15 \text{ dB}$ (c.f. Fig. 5.7(b)). At 6 bits/symbol, the gap between “capacity” and “MI, PSM” is 0.1 dB, while the gap between “capacity” and “MI, QAM” is 0.5 dB. The gain of 64-PSM over 64-QAM is 0.4 dB, but is still not better than in SISO. However, the “MI, PSM” can not reach 12 bits/symbol at high SNRs. This is again due to the nonbijectivity property of PSM when $N = 6$. Sadly, the Gaussian constellation does not give significant improvement as in the single antenna case. The main reason is that at the receiver side, the observation can also be seen as the summation of symbols from various transmit antennas. The Gaussian shape can be generated via multiple transmit antennas. Further increasing the number of transmit antennas will make it more Gaussian, thus closer to the channel capacity.

5.2.2 Outage Capacity

In a fast fading channel, the channel capacity is computed by averaging inside one symbol block due to its time variant property. By applying channel coding, the code word error probabilities can also be averaged. In slow fading channels, the channel coefficients are randomly generated but they are almost constant over time, i.e. the channel coherence time is larger than the latency requirement. The maximum achievable rate is

$$R_{\mathbf{H}} = \log \det \left(\mathbf{I}_{N_R} + \frac{\text{SNR}}{N_T} \mathbf{H}\mathbf{H}^* \right) , \quad (5.31)$$

which depends on the channel realizations \mathbf{H} . If the overall transmission rate is R_t bits/channel use, in the case that it is impossible to make the decoding error probability arbitrarily small, the system is in outage. The outage probability is defined as

$$P_{\text{out}}(R_t) := P \{ R_{\mathbf{H}} < R_t \} . \quad (5.32)$$

For a slow fading MIMO channel, the capacity of the outage probability is

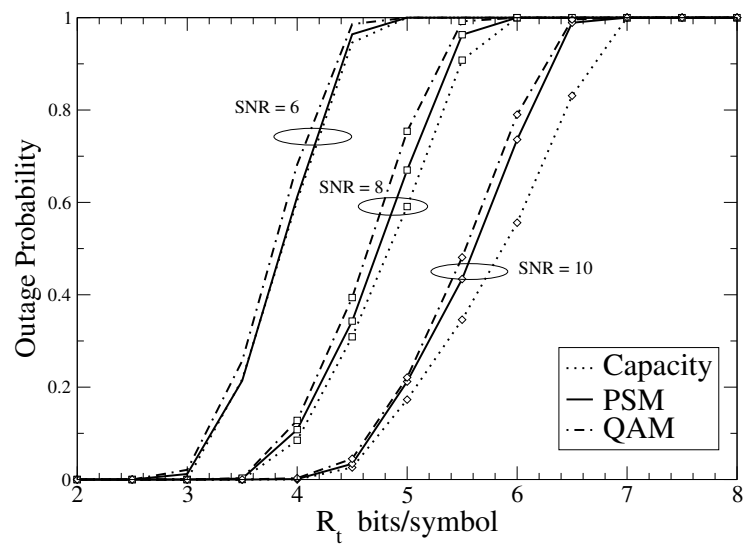
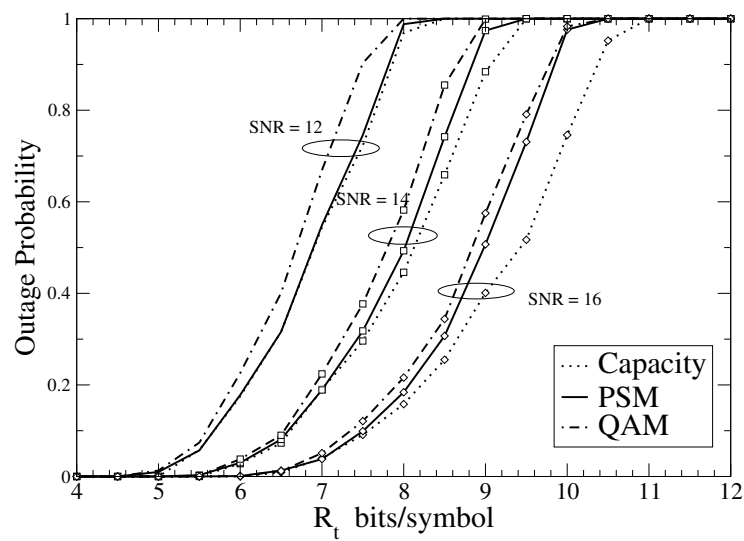
$$P_{\text{out}}(R_t) := P \left\{ \log \det \left(\mathbf{I}_{N_R} + \frac{\text{SNR}}{N_T} \mathbf{H}\mathbf{H}^* \right) < R_t \right\} . \quad (5.33)$$

If specific modulation schemes are utilized, $R_{\mathbf{H}}$ can be computed by

$$R_{\mathbf{H}} = E_{\mathbf{x}} \{ I(\mathbf{x}; \mathbf{y} | \mathbf{H}) \} . \quad (5.34)$$

Fig. 5.8 provides the outage probability for $N = 4$ and $N = 6$ by simulations. The simulations are tested for 10000 transmit blocks. Each transmit symbol block contains 1200 info bits. The channel coefficients are constant inside one block, but are randomly generated from block to block. SNRs are also defined the same as in the previous sections of this chapter.

From the simulation results, it can again be observed that at low SNRs, PSM is much closer to the capacity compared to square QAM. With increasing SNR, the curves of “PSM” still lie in between the “capacity” curves and the “QAM” curves. At higher SNRs, they both have gaps to the “capacity” curves due to the system limitation. From the capacity analysis, PSM schemes are closer to the capacity than square QAM, in both fast fading and slow fading channels.

(a) $N = 4$.(b) $N = 6$.Figure 5.8: Outage probability, $N_T = N_R = 2$.

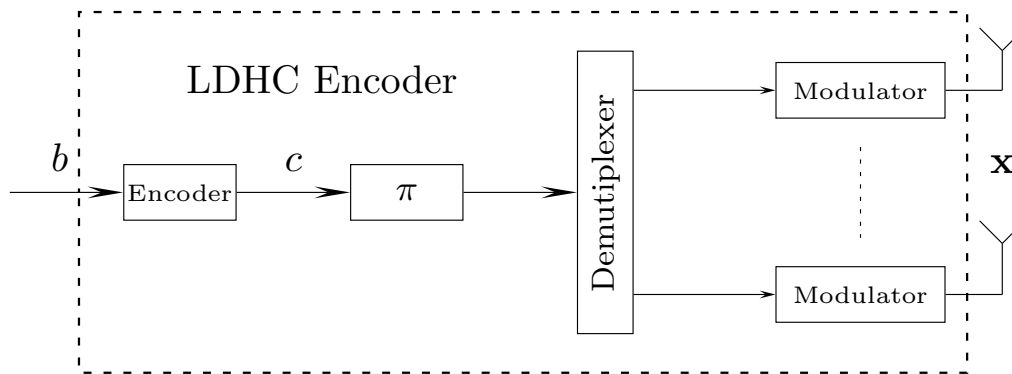


Figure 5.9: Block diagram of a BICM-MIMO transmitter using LDHC code philosophy.

5.3 LDHC-Code Based MIMO

So far the theoretically achievable rates for both modulation schemes have been analyzed in both fast and quasi static channels. To approach these channel capacities, bit-interleaved coded modulation transmission is applied. In this chapter, BICM transmission in case of multiple antennas is considered. For single antenna BICM transmission, the bit-wise interleaver manages to make the bits independent of each other in either the time domain and the frequency domain. Thus, a low-density hybrid-check (LDHC) code structure is applied to optimize coding and modulation. When multiple antennas are considered, the bit sequence is also transmitted via different antennas. The independence in the spatial domain should also be avoided. Hence, the LDHC code will be extended to BICM-MIMO, by taking the effects of multiple antennas into account. In the following, the concept of the LDHC based BICM-MIMO transmission will be proposed, along with different representations.

Fig. 5.9 provides a block diagram of the transmitter. In LDHC code design, both the encoder and the modulator are jointly designed. Hence, they have been included in the LDHC encoder. Besides, the code bits are also transmitted in the spatial domain via different transmit antennas. The transmit symbols from different antennas will be summed up at the receiver side. Hence, the various transmit antennas also need to be a part of the encoder. Correspondingly, at the receiver side, the receive antennas are included inside the LDHC decoder, which completes the whole block diagram of the

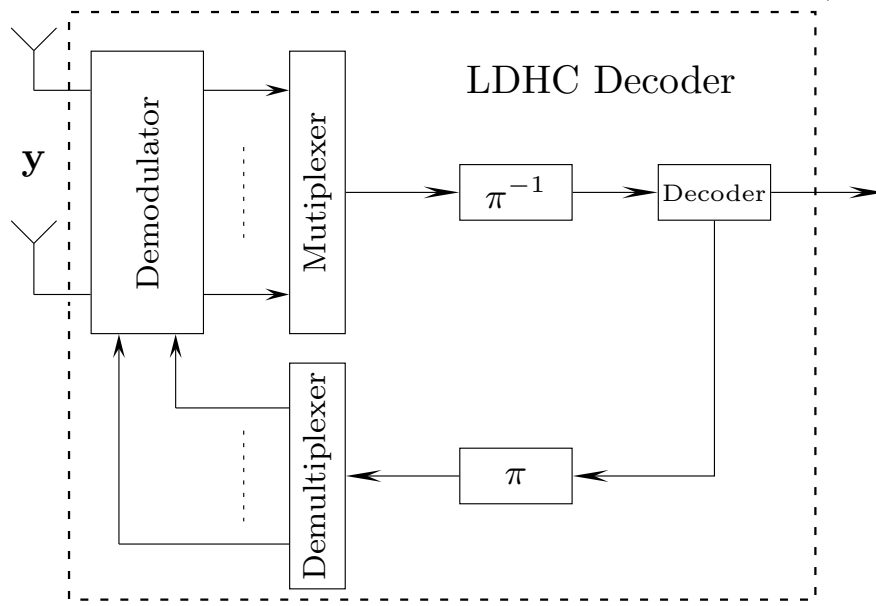


Figure 5.10: Block diagram of a BICM-MIMO receiver using LDHC code philosophy.

LDHC encoder and decoder. By doing so the LDHC code manages to take the spatial domain to be part of a universal design.

Consider one received symbol at time index k , which is the case of multiple-input and single-output (MISO), the received symbol is of the form

$$y[k] = \sum_{t=0}^{N_T-1} h_t[k]x_t[k] + z[k], \quad (5.35)$$

where $x_t[k]$ denotes the transmitted symbol from the t th antenna, and $h_t[k]$ represents the channel coefficient from the t th transmit antenna. $x_t[k]$ is a complex-valued symbol, whose symbol alphabet is determined by the modulation scheme and mapping rule. In case of superposition mapping, the modulated symbol is the weighted summation of BPSK mapped symbols, as is provided by

$$x[k] = \sum_{n=0}^{N-1} \alpha_n c_n[k], \quad (5.36)$$

where α_n stands for a constant weighting factor over time. From this relationship, a factor graph can be drawn accordingly, as is illustrated in Fig. 5.11. This graph provides an example of MISO with two transmit antennas. The modulation order is set to be two.

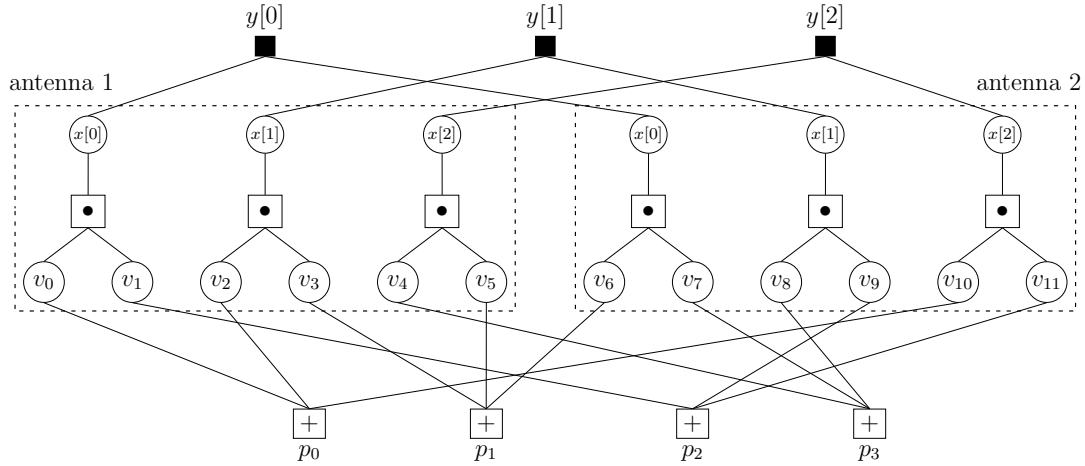


Figure 5.11: Factor graph of an LDHC code for MISO, $N_T = 2, N = 2, K = 3$.

For each antenna, a block length of three is considered. The whole graph is composed of mainly three parts. The upper part shows the multiple antenna combinations. The middle part illustrates the modulation. The bottom part reflects the coding. From the uppermost of Fig. 5.11, a \blacksquare denotes an observation node, which stands for the channel output at the receiver side. The observation nodes are connected to circles with x inside. A circle is a variable node, which stands for the transmit symbols. Since the number of transmit antennas is two, each observation node is connected to two variable nodes. Based on different transmit antennas, the variable nodes are separated into two groups (c.f. Fig. 5.11) by two dashed boxes. Inside the dashed boxes, there are also circles with v inside. These represent the code bits before modulation. The bit nodes and the variable nodes are connected via the so-called mapping nodes. The mapping nodes are denoted by boxes with \bullet inside in the Tanner graph. Here, a variable node is connected to two bit nodes because of the modulation order of two. At the bottom part, the boxes with $+$ inside are the parity-check nodes, which show the constraints of the associated code bits.

From the graph, it can clearly be seen that the code bits are not only across the time/frequency domain, but the spatial domain as well. The code design should also take the correlations in multiple domains into account. In this situation, the short cycles between the decoder and the demodulator need to be avoided not only in the time domain or frequency domain, but the spatial domain as well. The matrix to represent such a transmission system should also reflect the effects of the transmit antennas. Revisiting

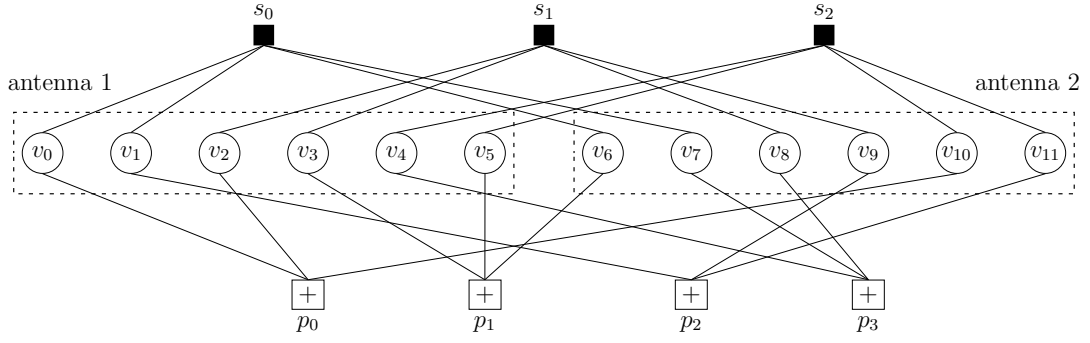


Figure 5.12: Simplified factor graph of the LDHC code.

	← antenna 1 →						← antenna 2 →						
	1	1	0	0	0	0	1	1	0	0	0	0	← s_0
	0	0	1	1	0	0	0	0	1	1	0	0	← s_1
	0	0	0	0	1	1	0	0	0	0	1	1	← s_2
↑	1	0	1	0	0	0	0	0	0	0	1	0	← p_0
↑	0	0	0	1	0	1	1	0	0	0	0	0	← p_1
↑	0	1	0	0	0	0	0	0	0	1	0	1	← p_2
↑	0	0	0	0	1	0	0	1	1	0	0	0	← p_3
		v_0	v_1	v_2	v_3	v_4		v_5	v_6	v_7	v_8	v_9	v_{10} v_{11}

Figure 5.13: Incidence matrix of an LDHC code for MISO, $N_T = 2, N = 2, K = 3$.

Fig. 5.11, the mapping nodes and the variable nodes can be merged. Then we obtain a simplified version of the factor graph, as is given by Fig. 5.12. The received symbols are replaced by the observation node representation “ s ”. For this simplified factor graph, its corresponding incidence matrix is provided by Fig. 5.13. The schedule of the message passing in the iterative decoding and demodulation in this Tanner graph is as follows. Based on the channel observations, the observation nodes use a certain algorithm to mitigate the multiple antenna interference and compute the extrinsic information of the code bits. Then, the output extrinsic LLR values are passed to the variable nodes. The messages are exchanged at the variable nodes and transferred to the parity-check nodes. The parity-check nodes also perform the message exchange and deliver them back the variable nodes. Afterwards, the LLR values are again updated at the variable nodes and passed back to the observation nodes for the next iteration.

The difference between this matrix from the previous incidence matrix is that the upper

part of the matrix is further divided into subblocks with number equivalent to the number of transmit antennas. Focusing on the upper matrix, each row denotes an observation node (received symbol). There will be a “1” in the position where the code bit belongs to the symbol which contributes to the summation of the received symbol. Hence, the row weight is actually equal to NN_T , where the row weight of each submatrix is the modulation order N . In each subblock, the time/frequency domain is reflected at each row. The spatial domain is reflected across the subblocks. Now the matrix is designed for the overall matrix. If the PEG algorithm is applied, the girth length of the whole LDHC code requires to be as large as possible. The dependence inside the LDPC code can be avoided. Adding the parity-check part, the dependence between coding and modulation is to be mitigated. In case of MIMO, the independences are not only in the time/frequency domain, but also in the spatial domain, if such an LDHC code structure is considered. Similarly, a concatenated code structure can also be realized if the variable node degrees w.r.t. the parity-check node degrees are set to be larger than one, as has been mentioned in Section 3.3.1.

5.3.1 Effects of Girth

It is known in LDPC code design that the girth length affects the convergence behavior of the system performance. Now that LDHC coded BICM-MIMO is considered, there exists many possible ways of cycles. The short cycles, especially the length-4 cycles, are harmful as a bit error will be difficult to correct in a trapping set. Assume one transmission error occurs inside such a short cycle, even if the remaining messages are error-free, it is still difficult to correct this error if the remaining messages are not strong enough. It is very likely that such a short cycle will stuck as an error cycle. The whole system performance will be hindered by the worst cases. If a longer cycle is formed, the chance that the whole cycles are trapped in an error propagation is much lower. In a longer cycle, the error message is in minority and the remaining correct messages will have a higher chance for correction given enough iterations. Hence, the cycles with length-4, or even length-6, are better to be avoided.

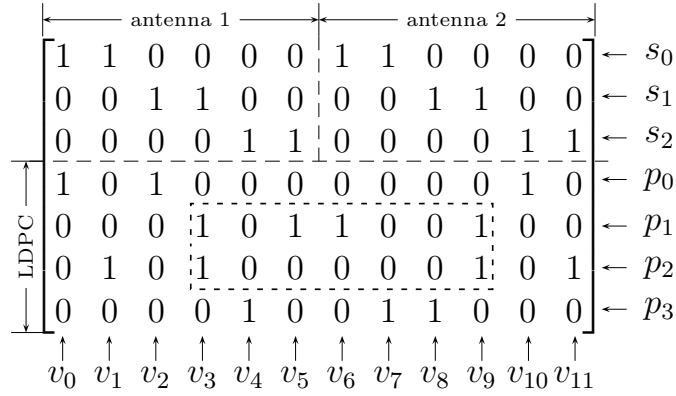


Figure 5.14: The first kind of short cycles.

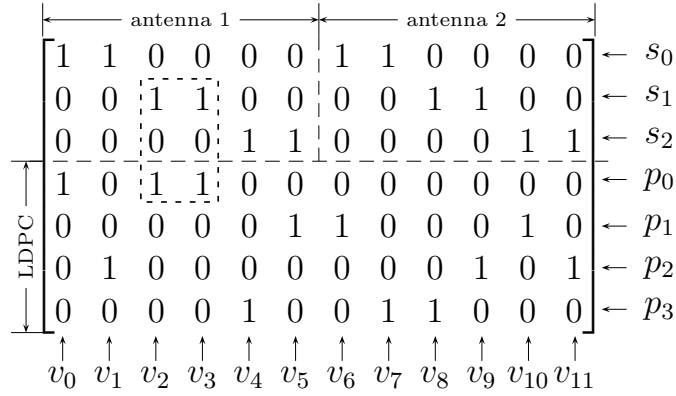


Figure 5.15: The second kind of short cycles.

The girth length refers to the minimum cycle length in the Tanner graph or in the incidence matrix. Here, we consider two kinds of code combination: a pure LDPC code and a concatenated code of both parity-check codes and repetition codes. First, the code with pure parity-check part is considered. The short cycles, e.g. the length-4 cycles, could appear as follows: The short cycles can occur between the variable nodes and the parity-check nodes if both the variable nodes and the parity-check nodes have degrees larger than one. In matrix representation, this kind of short cycles only exist inside the LDPC submatrix, as is depicted by a dashed box in Fig. 5.14. Considering in one antenna, the second kind of short cycles appear between the observation nodes and the parity-check nodes. An example of this kind of short cycle is illustrated by a dashed box in Fig. 5.15, where the cycle is across the LDPC submatrix and the antenna 1 submatrix. In previous chapters, we managed to maximize the length of this kind of cycles in single antenna

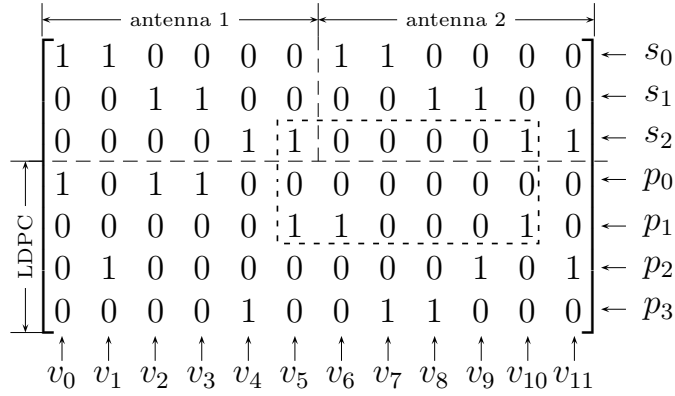


Figure 5.16: The third kind of short cycles.

scenarios. When multiple antenna technique is applied, there will be a third kind of short cycles. Consider the case when the two variable nodes are connected to a same parity-check node. Later on, these two bits are mapped to two different complex-valued symbols. However, these two symbols are obtained at the same observation at the channel output. Hence, in the Tanner graph, the two variable nodes are also connected to the same channel observation node. In this situation, it will also form a short cycle. This short cycle, in the matrix representation, is across the LDPC submatrix, the antenna 1 submatrix, and the antenna 2 submatrix, as is illustrated in Fig. 5.16.

Conventional computer-based code design methods have been used to mitigate the first kind of the short cycles only. In the previous chapter, we have proposed a modified PEG algorithm in order to enlarge the second kind of cycles. In case of MIMO, this algorithm can easily be applied to avoid the third kind of the short cycles. When constructing the incidence matrix of LDHC-based BICM-MIMO, the upper submatrices can be built easily, without considering the short cycles inside. After the upper matrices are finished, the PEG algorithm is thus utilized when constructing the LDPC submatrix while maximizing the second and the third kind of cycles.

In the previous matrix construction, the cycles across the upper submatrices will not appear because it is combined with a pure LDPC code. Then the column weight in the upper submatrices are always one. However, things are different when a concatenated code is considered. We have known that the concatenated code is helpful for the improvement of the systems, both in an AWGN channel and frequency-selective fading channels. The

$$\begin{array}{cccccccccccc}
& \xrightarrow{\text{antenna 1}} & & \xrightarrow{\text{antenna 2}} & & & & & & & & & & \\
\left[\begin{array}{cccccccccccc}
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0
\end{array} \right] & \leftarrow s_0 \\
& & & & & & & & & & & & & \leftarrow s_1 \\
& & & & & & & & & & & & & \leftarrow s_2 \\
\uparrow \text{LDPC} & & & & & & & & & & & & & \leftarrow p_0 \\
& & & & & & & & & & & & & \leftarrow p_1 \\
& & & & & & & & & & & & & \leftarrow p_2 \\
& & & & & & & & & & & & & \leftarrow p_3 \\
& \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \\
& v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} &
\end{array}$$

Figure 5.17: The fourth kind of short cycles.

problem of using concatenated codes is that it may cause a fourth kind of cycles. The matrix representation is provided by Fig. 5.17. As can be seen in the matrix, the column weights of the upper matrices are larger than one for some columns. In the Tanner graph, some variable nodes are connected to more than one observation nodes because of the repetition codes. Hence, the fourth kind of cycles may occur, as given by a dashed box in Fig. 5.17. This kind of short cycles, along with the above mentioned three other kinds of cycles will appear for a combination of the concatenated codes. The appearance of the fourth kind of cycles will increase the difficulty in the matrix design. In the construction of such a matrix, when adding ones in the upper matrices, care should be taken so that the cycle length of the fourth kind should be as large as possible. This makes it even more complicated if the number of transmit antennas is larger. After finishing the design of the upper submatrices, the lower LDPC submatrix is generated while using the modified PEG algorithm to avoid the first, the second, and the third kind of the short cycles. Although concatenated codes may provide a performance gain over a pure parity-check code for higher-order modulations, the efforts in avoiding the short cycles are also higher.

5.4 Detection Algorithms

In this section, possible detection algorithms for MIMO with PSM will be considered and compared. It has been mentioned in Section 5.1 that the APP detector is optimum in terms of performance. However, its computational complexity is exponentially growing

with the number of transmit antennas and the modulation order. Hence, many researchers have tried to find some alternative detection algorithms with a reduced complexity but hopefully less performance penalty. In the following, only iterative detection algorithms are considered. Thus, the posteriori information computed by the detection algorithm shall include both the channel observations and the a priori information from the decoder feedback. Let us assume that the soft information from the decoder is LLR^a . Sequentially, the a priori probabilities of code bit c are computed by

$$P(c = 0) = \frac{e^{\text{LLR}^a}}{1 + e^{\text{LLR}^a}}, \quad P(c = 1) = \frac{1}{1 + e^{\text{LLR}^a}}. \quad (5.37)$$

This a priori information is general for all following mentioned detection algorithms. However, for the first iteration,

$$P(c = 0) = P(c = 1) = 0.5, \quad (5.38)$$

as there are no feedback information. The considered MIMO detector is separated into two steps. The first step is to cancel the multi-antenna interference. Then based on the detected symbols, the demodulation algorithm is used in the next step to calculate the extrinsic information of the code bits. For the demodulation of a complex-valued symbol, a tree-based APP algorithm has shown to provide a lower complexity compared to the standard APP detection without performance penalty. Thus, the following algorithms will only differ in the way how the multi-antenna interference is eliminated.

5.4.1 Gaussian Approximation Detection

The Gaussian approximation detector is a low complexity detection algorithm. The original type of this detection has been proposed in [104, 105]. The main idea of the detector is by approximating the multi-antenna interference as a Gaussian distributed random variable. According to the central limited theorem, the summation of a large number of independent random variables naturally delivers a Gaussian distribution. The Gaussian approximation detector is valid for BPSK [105], especially when the number of transmit antennas is high. However, it does not work well in case of square QAM due to the mismatch of the interference to the Gaussian approximation [106]. If we apply superposition

mapping, whose signal constellations are quasi Gaussian distributed, the summation of the interference components shall match to a Gaussian distribution. The details of the Gaussian approximated detection are as follows. Here, we only approximate the multi-antenna interference instead of the bit interference in an PSM symbol as a Gaussian distribution.

Consider a MIMO scenario. One channel observation at the r th receiver antenna is of the form

$$y_r = \sum_{t=0}^{N_T-1} h_{rt}x_t + z_r, \quad (5.39)$$

where h_{rt} denotes the channel coefficient of the subchannel from the t th transmit antenna to the r th receive antenna and z_r represents the AWGN noise sample. This equation can be further divided into several parts, the symbols to be detected, the multiple antenna interference, and the AWGN noise sample:

$$y_r = h_{rt}x_t + \sum_{i=0, i \neq t}^{N_T-1} h_{ri}x_i + z_r = h_{rt}x_t + v_t, \quad (5.40)$$

where v_t is the summation of the multi-antenna interference and the AWGN noise sample. For PSM, each x_i is the summation of the bipolar symbols multiplied with the weighting factors:

$$x_i = \sum_{n=0}^{N-1} \alpha_{i,n}c_{i,n}. \quad (5.41)$$

By replacing (5.41) into (5.40), we have the interference part of the form

$$v_t = \sum_{i=0, i \neq t}^{N_T-1} h_{ri}x_i + z_r = \sum_{i=0, i \neq t}^{N_T-1} h_{ri} \sum_{n=0}^{N-1} \alpha_{i,n}c_{i,n} + z_r. \quad (5.42)$$

In this detector, v_t is approximated by a Gaussian distributed variable, with the parameter

$$v_t \sim \mathcal{CN}(\mu_{v_t}, \sigma_{v_t}^2), \quad (5.43)$$

with μ_{v_t} being its mean and $\sigma_{v_t}^2$ being its variance. According to (5.40), the transition probability $p(y_r|x_t)$ can thus be written as

$$p(y_r|x_t) = \frac{1}{\pi\sigma_{v_t}^2} \exp\left(-\frac{|y_r - \mu_{v_t} - h_{rt}x_t|^2}{\sigma_{v_t}^2}\right). \quad (5.44)$$

μ_{v_t} and $\sigma_{v_t}^2$ are obtained using the a priori probabilities. Revisiting (5.40), to get the parameters for v_t , the mean and variance for each x_i also need to be calculated first. According to the definitions, the mean and variance of each $c_{i,n}$ are computed as follows:

$$\begin{aligned}\mu_{c_{i,n}} &= P(c_{i,n} = 0)(+1) + P(c_{i,n} = 1)(-1) = P(c_{i,n} = 0) - P(c_{i,n} = 1) , \\ \sigma_{c_{i,n}}^2 &= P(c_{i,n} = 0)(+1)^2 + P(c_{i,n} = 1)(-1)^2 - \mu_{c_{i,n}}^2 = 1 - \mu_{c_{i,n}}^2 .\end{aligned}\quad (5.45)$$

Both $P(c_{i,n} = 0)$ and $P(c_{i,n} = 1)$ are a priori probabilities. As x_i is composed of independent of bits, its mean and variance are the weighted summations of the mean and variance of each component,

$$\begin{aligned}\mu_{x_i} &= \sum_{n=0}^{N-1} \alpha_{i,n} \mu_{c_{i,n}} , \\ \sigma_{x_i}^2 &= \sum_{n=0}^{N-1} |\alpha_{i,n}|^2 \sigma_{c_{i,n}}^2 .\end{aligned}\quad (5.46)$$

Similarly, given AWGN noise terms with variance σ_z^2 , the mean and variance of v_t are

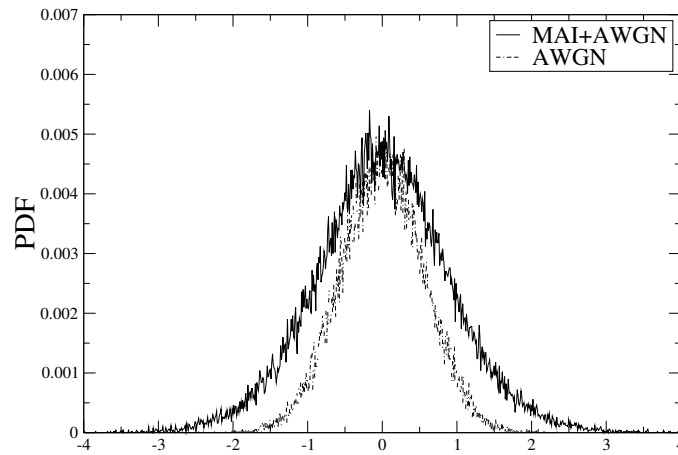
$$\begin{aligned}\mu_{v_t} &= \sum_{i=0, i \neq t}^{N_T-1} h_{ri} \mu_{x_i} , \\ \sigma_{v_t}^2 &= \sum_{i=0, i \neq t}^{N_T-1} |h_{ri}|^2 \sigma_{x_i}^2 + \sigma_z^2 .\end{aligned}\quad (5.47)$$

Now that with the mean and the variance of the interference in hand, the LLR of each bit is computed by

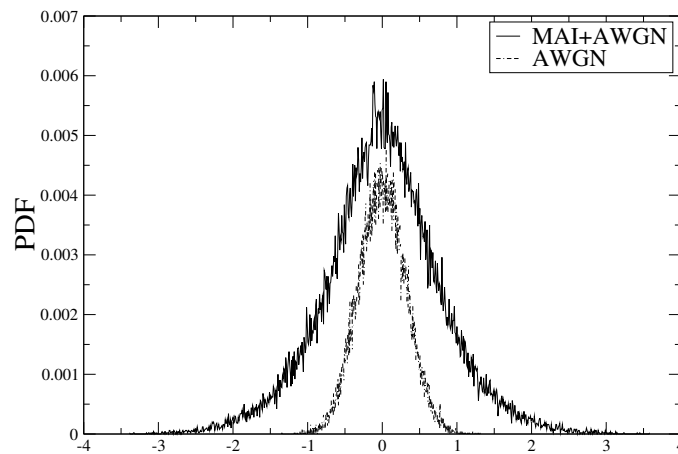
$$\text{LLR}_{r,t,n} = \ln \frac{\sum_{x \in \mathcal{X}_n^0} p(y_r | x_t) \prod_{j=0, j \neq n}^{N-1} P(c_{t,j})}{\sum_{x \in \mathcal{X}_n^1} p(y_r | x_t) \prod_{j=0, j \neq n}^{N-1} P(c_{t,j})} .\quad (5.48)$$

This is the extrinsic LLR value computed from one observation at the r th antenna. For multiple receive antennas, the observations of the same code bit will be obtained also from other receive antennas. These values shall also be combined before they are passed to the decoder. The final output LLR values are the summation of the LLR values from all the receive antennas, as is provided by

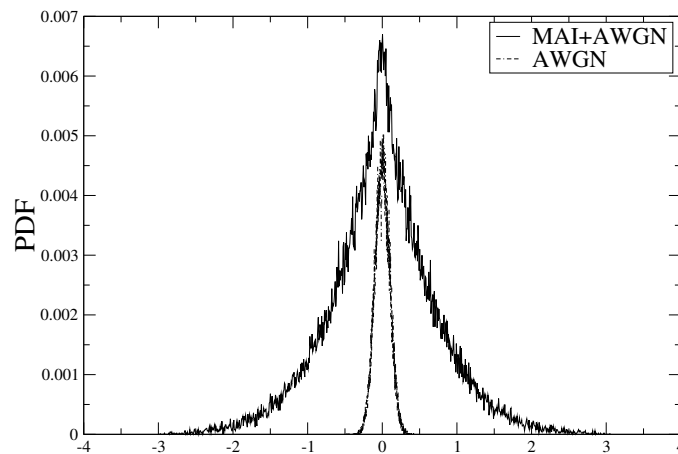
$$\text{LLR}_{t,n} = \sum_{r=0}^{N_R-1} \text{LLR}_{r,t,n} .\quad (5.49)$$



(a) SNR =5 dB.



(b) SNR=10 dB.



(c) SNR=20 dB.

Figure 5.18: Pdf of interference and noise, $N = 6$, $N_T = N_R = 2$.

The key point of this algorithm is the approximation of the interference. If the distribution of the interference is truly Gaussian distributed, this method should work properly. Otherwise, the correctness of the interference cancellation is questionable. For BPSK, or 4-QAM, the imperfectness of the approximation is not very critical because the symbol cardinality is small. However, for higher-order modulations, the magnitude of the symbols have different values. Hence, even one bit error might cause a huge difference between the estimated interference and the exact interference. The interference contains the multi-antenna interference (MAI) and the AWGN noise. Fig. 5.18 provides the probability density functions (pdfs) of the interference for different SNRs, with $N = 6$ and $N_T = N_R = 2$. The pdfs of only the AWGN noise are also illustrated. From these three figures, we have the following conclusions. First, the MAI is indeed Gaussian distributed. This is due to the Gaussian distribution of the PSM signal constellations. Second, this kind of detector only works well in the low SNR regime. It can be observed in the low SNR regime that the shape of the two pdfs are similar. Both the peak value and the width of the two pdfs are very close. Hence, if a system can converge at a low SNR value, the Gaussian approximation is valid. In cases of large number of receive antennas with BPSK, where the systems converge at low SNRs, the Gaussian approximation will work. However, if a system converges at a relatively high SNR, the approximation comes with a problem. As can be observed by Fig. 5.18, when SNR=20 dB, there is a big difference between the two pdfs. They differ both in the peak value and the width of the pdf. These differences can not be compensated by further increasing the SNR values. As a result, the systems which converge at high SNRs can not be applied by the Gaussian approximation detector. As a matter of fact, MIMO with higher-order modulation converge normally at higher SNRs, another detection algorithm is required in such situations. The complexity comparison of these detection algorithms will be mentioned after the next detector has been described.

5.4.2 MMSE Detection

In this section, we take a look at minimum mean-square error (MMSE) detection. The MMSE demodulation has initially been proposed in [107, 108, 109, 110]. This demod-

ulator is an improved version of the zero-forcing (ZF) demodulator [111]. Later on, it has been extended to iterative processing in [112]. The idea is to multiply the received values by certain filter coefficients so that the original transmit symbols can be retrieved. Nowadays, the MMSE equalizer with subsequent LLR calculations has been widely utilized for detection of BICM-MIMO systems due to its low complexity and near optimum performance.

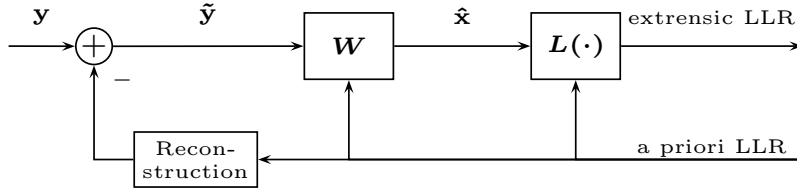


Figure 5.19: Block diagram of MMSE detector.

A structure of an MMSE detector is illustrated in Fig. 5.19. There are mainly three steps for an MMSE detector. The a priori information is required for each step for iterative processing. Upon receiving a priori LLRs, the a priori probabilities are computed following (5.37). In the first step, the a priori probabilities of bits are used for the cancellation of the interference. Sequentially, the symbol probabilities of the t th transmit antenna can also be obtained by

$$P(x_t) = \prod_{n=0}^{N-1} P(c_{t,n}) . \quad (5.50)$$

With $P(x_t)$ at hand, the mean and variance of the symbol are computed by definitions:

$$\begin{aligned} \mu_{x_t} &= \sum_{x \in \mathcal{X}} x P(x_t = x) \\ \sigma_{x_t}^2 &= \sum_{x \in \mathcal{X}} |x|^2 P(x_t = x) - |\mu_{x_t}|^2 , \end{aligned} \quad (5.51)$$

where \mathcal{X} is the symbol cardinality. Then these values can be used to compute and cancel the estimated multi-antenna interference from the received symbol, as the first step of MMSE detection. Assume y_r denotes the received observation at the r th antenna. After interference subtraction:

$$\tilde{y}_r = y_r - \sum_{t=0}^{N_T-1} h_{rt} \mu_{x_t} , \quad (5.52)$$

In order to further reduce the residual error, $\tilde{\mathbf{y}} = [\tilde{y}_0, \dots, \tilde{y}_{N_R-1}]$ is passed through an MMSE filter that provides the symbol estimates $\tilde{\mathbf{x}} = [\hat{x}_0, \dots, \hat{x}_{N_T-1}]$, which is the second step. The goal of the filter is to minimize the mean square error $E(|x_t - \hat{x}_t|^2)$. The t th estimate is computed by

$$\hat{x}_t = \mathbf{W}_t^H \tilde{\mathbf{y}}. \quad (5.53)$$

The filtering coefficient matrix is given by

$$\mathbf{W}_t = (\mathbf{H}\mathbf{V}_t\mathbf{H}^H + \sigma_z^2\mathbf{I})^{-1}E_s\mathbf{h}_t, \quad (5.54)$$

with

$$\mathbf{V}_t = \text{diag}(\sigma_{x_0}^2, \dots, \sigma_{x_{t-1}}^2, E_s, \sigma_{x_{t+1}}^2, \dots, \sigma_{x_{N_T-1}}^2), \quad (5.55)$$

where E_s is the average transmit power, $(\cdot)^H$ denotes the Hermitian operation and \mathbf{h}_t represents the t th column of \mathbf{H} .

As the third step, the extrinsic LLR values are computed using the estimated symbols. According to the definition, the extrinsic LLR is computed by

$$\text{LLR}_{t,n} = \ln \frac{\sum_{x \in \mathcal{X}_n^0} p(\tilde{x}_t|x_t) \prod_{j=0, j \neq n}^{N-1} P(c_{t,j})}{\sum_{x \in \mathcal{X}_n^1} p(\tilde{x}_t|x_t) \prod_{j=0, j \neq n}^{N-1} P(c_{t,j})}. \quad (5.56)$$

As the a priori probabilities have been obtained, the key point is to obtain $p(\hat{x}_t|x_t)$. It has been shown in [113] that the MMSE estimate approximates a Gaussian distribution.

Hence, we have

$$p(\hat{x}_t|x_t) \approx \frac{1}{\pi\sigma_{\eta_t}^2} \exp\left(-\frac{|\hat{x}_t - \mu_{\eta_t}|^2}{\sigma_{\eta_t}^2}\right), \quad (5.57)$$

with

$$\begin{aligned} \mu_{\eta_t} &= \mathbf{W}_t^H \mathbf{h}_t x_t \\ \sigma_{\eta_t}^2 &= \mathbf{W}_t^H (\mathbf{H}\mathbf{V}_t\mathbf{H}^H + \sigma_z^2\mathbf{I})^{-1} \mathbf{W}_t \end{aligned} \quad (5.58)$$

as the mean and variance of the estimate \hat{x}_t . The mean is affected by the channel coefficients and the filtering coefficients. The variance consists of the AWGN and the remaining interference. This completes a soft MMSE based demapping.

5.4.3 Comparison

The above two detection algorithms, along with the standard APP detection algorithm only differ in the computation of the transition probabilities. The demodulation of the symbols are the same. Hence, the comparison of their complexity is by calculating the transition probabilities.

In terms of computational complexity, for the standard APP, the distances between the received vector and all the possible combinations of bits on different antennas need to be compared. Thus, the complexity would be

$$2^{NN_T} . \quad (5.59)$$

Consider a Gaussian approximation detector. For computing the parameters of the interferences, $N_T N$ operations are necessary. After the interference cancellation, the LLR computations for the code bits follows the standard APP algorithm, each with the complexity 2^N . As a result, the overall complexity is

$$N_T N 2^N . \quad (5.60)$$

Concerning MMSE detection, the matrix inversion operation requires a complexity of N_T^3 , and the LLR computations are still 2^N . Consequently, the overall complexity is

$$N_T^3 2^N . \quad (5.61)$$

From these equations, the complexity of the three algorithms can be compared, apart from the LLR computations of the code bits. The complexity also differs in the mitigation of the multi-antenna interference. The complexity of the standard APP grows exponentially with the number of transmit antennas N_T . The Gaussian approximation detector grows linearly with NN_T . The MMSE detector grows linearly with N_T^3 . It is clear that Gaussian approximation detector provides the lowest complexity and the standard APP detector has the highest complexity. MMSE detection lies in between, but much lower than standard APP.

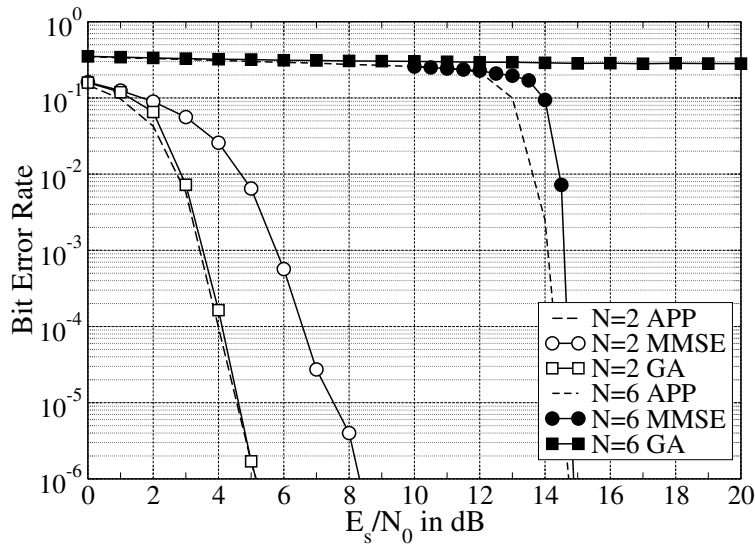


Figure 5.20: BER vs. SNR, different detections, $N_T = N_R = 2$.

The performance difference between the three different detectors are shown in Fig. 5.20. Both the number of transmit and receive antennas is set to be 2 and an LDPC code with rate 1/2 is chosen for the channel coding. APP detection serves as a lower bound for the optimum performance. When $N = 2$, the Gaussian approximated (GA) detector is very close to the lower bound. This is because the convergence happens at about 5 dB. In this region, the GA detector works perfectly. This agrees with our previous analysis. On the other hand, the MMSE detector is 2 dB worse in this case. However, things changes totally for higher-order modulation schemes. When $N = 6$, MMSE shows only a slight performance degradation compared to the standard APP detector. However, the GA detector completely fails to converge, due to the mismatch of the Gaussian approximation in the high SNR regime. If we are interested in investigating higher-order PSM modulation such as $N = 6$ or $N = 8$, the GA detector fails to work, unfortunately. The complexity of standard APP detection is much higher. Hence, to have a better trade-off between the performance and the complexity, the MMSE detector will be chosen for the detection of BICM-MIMO with PSM in the upcoming simulations. Afterwards, the tree-based APP detection algorithm is used for the LLR computations. As the curves suggest, the performance loss by using MMSE detector in case of $N = 6, N_T = N_R = 2$ is only 2dB worse than the optimum detection.

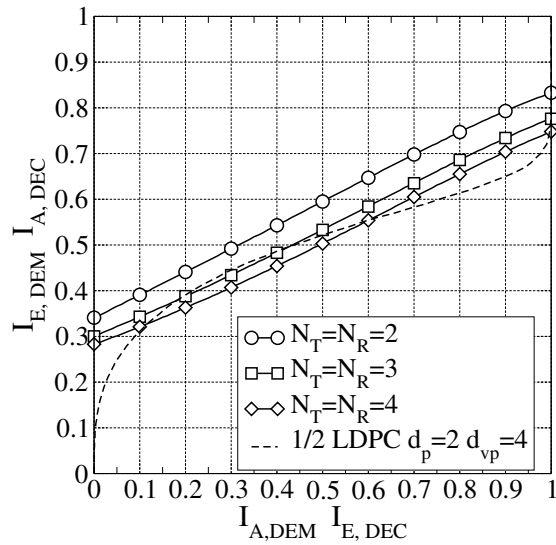


Figure 5.21: EXIT charts for MIMO-PSM with MMSE detection, SNR=15dB, N=6.

5.5 EXIT Chart Analysis

In previous chapters, EXIT chart analysis has been utilized for investigating transfer functions of iterative processing. For PSM with multiple antennas, the performance difference caused by different antenna numbers can also be observed via EXIT chart analysis. Fig. 5.21 provides an example of a PSM demodulator with different antenna numbers. The chosen detection algorithm is MMSE. The simulations are run at $E_s/N_0 = 15$ dB for PSM with $N=6$. The antenna numbers are chosen from 2 to 4. It can clearly be seen that the curves are almost parallelly descending with the increment of both transmit and receive antenna numbers. The slope of the curves will not change with different antenna numbers. Besides, the EXIT chart analysis can also tell us at which SNR a coded modulation system will start to converge combining a specific code. In Fig. 5.21, a transfer curve of a regular LDPC code is also drawn using a dashed line. The chosen degree distributions of this code is $d_p = 2$ and $d_{vp} = 4$, which is exactly the same as the code used in Fig. 5.20. From the BER results of Fig. 5.20, a system with the same parameters delivers a Turbo cliff at about 15 dB. In the EXIT chart, there exists a tunnel between the demodulator and decoder curves. The two results agree with each other. However, if continuing increasing the transmit and receive antenna numbers, the crossover will occur between the two curves, meaning a higher SNR is required for that system starts to converge.

5.6 Numerical Results

In this section, numerical results are provided to observe the performance of LDHC-coded based MIMO transmission. In the single antenna case, it has been shown that a proper chosen degree allocation is vital for the improvement of the system performance. Besides, more focus should be taken on the global iterations for superposition mapping. In case of MIMO, the effects of girth are extremely important due to the dependence in both the time domain and the spatial domain. How the number of antennas affects the overall performance will also be considered.

In the following simulations, a Rayleigh fading channel is assumed. The channel is assumed to be time-varying but not frequency-selective. The modulation scheme is again PSM. At the receiver side, due to the high computational complexity of the standard APP detector, a reduced complexity receiver is applied. MMSE detection is used to eliminate the multi-antenna interference. Then, the detection of the code bits follows the tree-based APP algorithm. According to the results in the single antenna case, the global iteration number is set to be 20 and the local iteration number is chosen to be 4. Again, a moderate number of code bits are assumed. The number of transmit symbols is 1000.

5.6.1 Effects of Degree Allocation

The degree allocation is controlled by setting the degrees of the variable nodes w.r.t. the mapping nodes to a value which is larger than 1. Doing so, there will be more information from the demodulation feedback for the improvement of the iterative detection. It has also been shown that in the SISO case, by allocating part of the code rate for R_s from R_p helps to make the BER curves of PSM converge earlier. However, it is not happening in MIMO.

Fig. 5.22 provides a comparison between the systems with and without R_s part. In these simulations, both the transmitter and the receive antenna numbers are chosen to be 2. The modulation order is $N = 6$. The degree polynomials for the codes are as follows. For

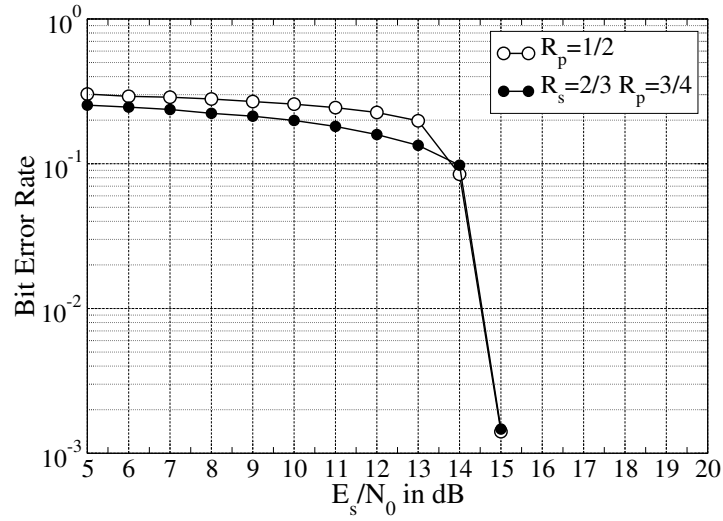


Figure 5.22: BER vs. SNR, different degree allocations, $N_T = N_R = 2, N = 6$.

the code $R_p = 1/2$, we have a regular code with

$$d_{vp} = 2, d_p = 4. \quad (5.62)$$

For the code $R_s = 2/3$ and $R_p = 3/4$, we have an irregular concatenated code with the degree distributions

$$\begin{aligned} \lambda_s(X) &= 0.50 + 0.50X, \\ \lambda_p(X) &= X, \\ \rho(X) &= X^7. \end{aligned} \quad (5.63)$$

As the simulation results show, taking the code rate from R_p to R_s will not enable the system to converge earlier. The reason is because of the multiple receive antennas. In SISO cases, the repetition codes provide more observations of one code bit, which help for a correct detection of this code bit. This function can be realized by the receive antennas. With more receive antennas, more observations will be obtained for one code bit. Hence, by adding the receive antennas, the degree allocation adaption will provide no obvious performance improvement. Besides, with a concatenated code, it will also bring extra efforts in eliminating the fourth kind of short cycles in the matrix design. Due to these two reasons, a pure parity-check code would be sufficient for this system.

5.6.2 Effects of Girth

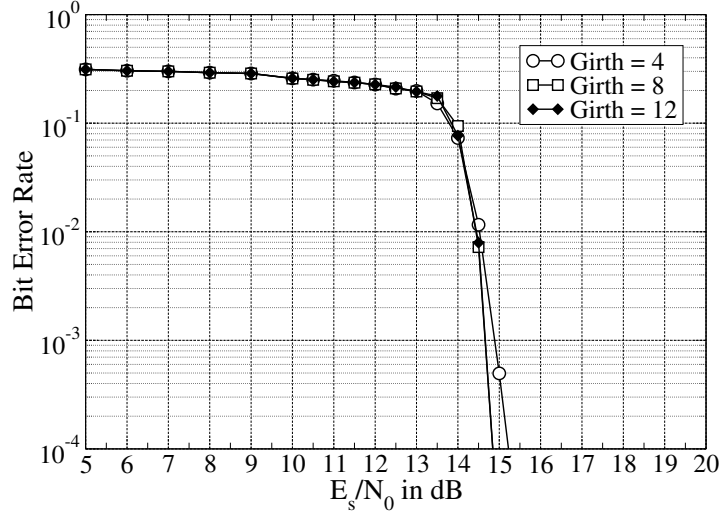


Figure 5.23: BER vs. SNR, different girth lengths, $N_T = N_R = 2$, $N = 6$.

In this part of the simulation, the effects of the girth length is considered. As has been mentioned before, the girth length refers to the minimum cycle length of the incidence matrix. In terms of LDHC code structure, the cycles include three types of cycles. They are the cycles having only variable nodes and parity-check nodes, the cycles containing only mapping nodes and variable nodes, and also the cycles with all of three kinds of nodes. In order to avoid error propagation inside the cycles, the girth length should be as large as possible. However, the cycle length is limited by the matrix size, and also the degree distributions. Besides, it is more likely to have a large girth for a regular LDHC code. It is because for irregular cases, the large column weight or row weight will appear. Hence, length-4 and length-6 cycles are not easy to avoid given the incidence matrix is not large enough. To maintain a large girth length, we use a regular code for illustration purpose.

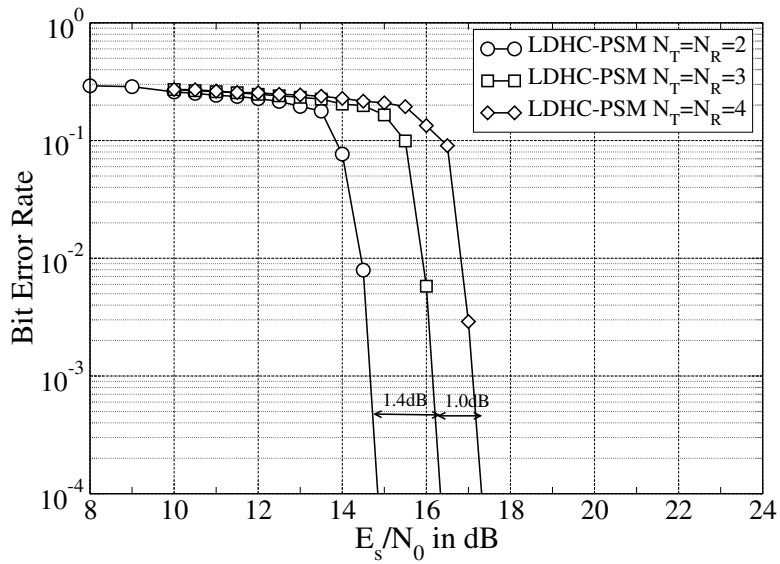
In Fig. 5.23, the same parameters as in Section 5.6.1 have been picked. The LDHC incidence matrix is obtained via the PEG algorithm. The resulting BER performance is illustrated in Fig. 5.23 with different girth lengths. The minimum girth length is 4. For the simulation setup, the info word length is 3000 and the code rate is 1/2. Concerning the LDHC incidence matrix, the column number is 6000 and the row number for the lower part is 3000. With $N_T = 2$ and $N = 6$, in the upper part, the row number is

$6000/N_T/N = 500$. The resulting matrix has dimension 3500×6000 . The maximum girth length is 12. Observed from Fig. 5.23, the case of girth= 4 means a random interleaver. It is known that because both the length-4 and length-6 cycles are easier to cause an error propagation during iterations, they should be mitigated. If the girth length is increased to 8, an approximately 0.5 dB performance gain can be observed compared to the random interleaver. However, further increasing the girth length will not improved the performance accordingly, even at the maximum achievable girth length of 12. From these comparisons, it is shown that the effects of girth length is helpful for the code design.

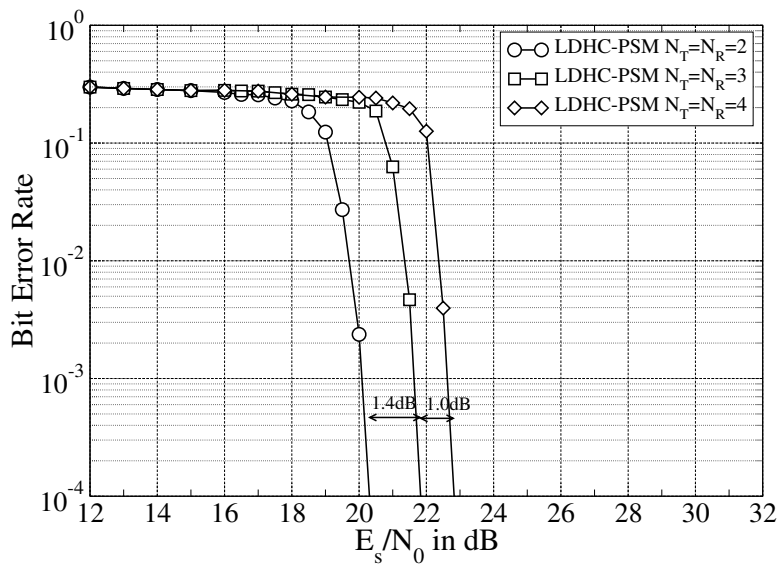
5.6.3 Effects of Antenna Numbers

In case of MIMO, the transmit antennas are applied to increase the channel capacity. From the derivations in Section 5.2.1, the theoretical lower bounds for BER performance do not affect much by the antenna numbers. The curves for the lower bounds are also illustrated in Fig. 5.24(a) and Fig. 5.24(b). Given a fixed modulation order, the performance degradation by increasing the number of antennas is only less than 0.2 dB. However, for practical transmissions, the gaps between the performance difference are larger.

Now the performance is compared in Fig. 5.24(a) and Fig. 5.24(b). The info word length is chosen so that the number of transmit symbol is 1000 for each transmit antenna in all the cases. The code parameters are the same as the previous simulation with the girth lengths requiring to be 8. As both Fig. 5.24(a) and Fig. 5.24(b) show, the case $N_T = N_R = 3$ is 1.4 dB worse than the case $N_T = N_R = 2$, whereas the case $N_T = N_R = 4$ is approximately 1 dB worse than the case $N_T = N_R = 3$. Such a large performance difference is partially due to the MMSE detection. The MMSE detection is here to cancel the multi-antenna interference. When the antenna number is increased, it is more difficult to cancel the multi-antenna interference perfectly. Due to this reason, a larger performance gap can be obtained when increasing the number of antennas for both transmitter and receiver sides. Besides, these results also proves the EXIT chart results in Fig. 5.21. The BER curves converge at around 15dB for $N_T = N_R = 2$. For higher number of antennas, a higher E_s/N_0 value for convergence is required.



(a) $N = 6$.



(b) $N = 8$.

Figure 5.24: BER vs. SNR for different antenna numbers.

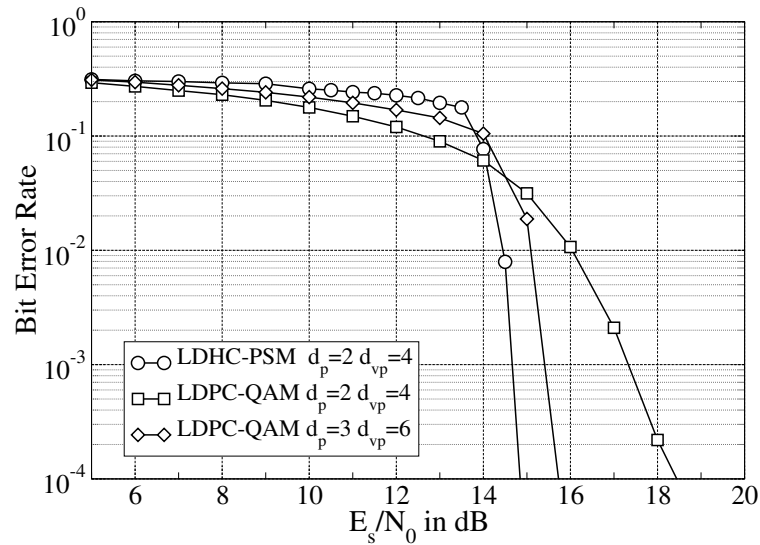


Figure 5.25: BER, LDHC-PSM vs. LDPC-QAM, $N_T = N_R = 2$, $N = 6$.

5.6.4 Performance Comparison

In this section, the performance of LDHC coded PSM is compared with LDPC coded square QAM. The modulation order is $N = 6$. The antenna numbers for both transmit and receive side are two. For both cases, a regular code with rate $1/2$ is applied and the cycle length is set to be 8 for both cases. In case of square QAM, LDPC coding with different degree distributions are compared. Given a fixed code rate, the degrees of the variable nodes and the parity-check nodes increase at the same time. The simulation results are provided in Fig. 5.25. From the BER results, the code with higher degrees of both nodes are better fit for square QAM. However, higher degrees requires more message exchanges during the decoding procedures, which leads to a higher complexity and a longer delay in the stage of decoding. On the other hand, a code with lower degrees are better fit for PSM, which can also be proved from the transfer curves for PSM. Hence, from the complexity point of view, the combination of LDHC-PSM provides a lower decoding complexity than LDPC-QAM. On the other hand, LDHC-PSM also converges earlier than the LDPC-QAM with higher degrees, which also makes it a better candidate of these two combinations.

5.7 Summary

In this chapter, an analysis for superposition mapping with multiple antennas has been conducted. Although in both time-varying and time invariant channels, the achievable bandwidth efficiency of PSM is better than that of square QAM, the gain is only within 1 dB. Different detection algorithms for the detections are compared. The GA detector has the lowest complexity and is fit for the case of lower-order modulations. The MMSE is more suitable for higher-order modulations as it can provide a better trade-off between complexity and performance. Then, the LDHC philosophy is utilized for the design of BICM-MIMO with PSM. In this incidence matrix design, four different kinds of short cycles need to be eliminated. Thus, the performance is improved, which is approved by the BER results. It is also shown in the BER results that concatenated codes are not necessary for this kind of system as the function of repetition can be compensated by multiple receive antennas.

Chapter 6

Conclusions

It has been provided by many previous contributions that superposition mapping is able to deliver a quasi Gaussian distributed signal, thus having the potential to approach the channel capacity in the high SNR regime. In [35, 37], the theoretical benefits of using superposition mapping have been analyzed. An LDHC code was proposed in search for suitable channel codes matched to superposition mapping. A capacity achieving performance has been reported for the AWGN channel. However, the performance is achieved requiring extremely long burst lengths and tremendous number of iterations. In this thesis, LDHC coding is implemented in OFDM and MIMO transmission. The main focus lies on finding a suitable coded modulation combination for a system with moderate burst length and adequate number of iterations.

In previous works, it has been shown by the EXIT chart analysis that the capacity is approached by irregular coded superposition mapping. Those irregular codes are obtained by finding a code with suitable degree distribution polynomials. Nevertheless, the following two conditions are required. One is an extremely long sequence length so that the chance that the occurrence of trapping sets is low. The other is a sufficient number of iterations so that the trajectory can go through the tunnel between the two curves in the EXIT chart. In this thesis, the performance of the considered system is first investigated for the AWGN channel given a moderate burst length. It has been shown that adaptation by degree distribution is very sensitive to the burst length. The channel codes obtained

by degree allocation adaptation delivers more stable performance regardless of the burst length. Applying LDHC coding, there are two iterations: local iterations between the variable nodes and the parity-check nodes, and global iterations between the demodulator and the decoder. It has also been observed that in an LDHC code design, increasing the number of global iterations has a stronger effect on improving the system performance than that of local iterations, for superposition mapping. Thus, a desirable performance with adequate overall iteration number can be found by using more global iterations.

In case of OFDM, the channel capacity can be fully exploited by assigning suitable bits to subchannel subject to its frequency response. Thus, LDHC coding is utilized in BICM-OFDM with adaptive bit loading. In terms of superposition mapping, the degree distributions are varied for different bandwidth efficiencies if a capacity achieving performance is required. It is challenging to find a universal code for the non-constant modulation scheme. Thus, it is not suitable to apply the degree distribution in this type of system. On the other hand, the adaptation by degree allocation is stable for different bandwidth efficiencies. Hence, it is again used here. Besides, superposition mapping provides more degree of freedom in choosing the step of bit loading. The results show that LDHC coded PSM outperforms LDPC coded square QAM in BICM-OFDM.

The generation of a superposition mapped symbol is similar to the formation of a multiple-antenna output. It is straightforward to apply LDHC coding to BICM-MIMO with superposition mapping. From the theoretical analysis, the capacity of multiple antenna superposition mapping is also higher than the capacity of conventional modulation schemes. However, the gap is less than that in the case of single antennas. The reason is that MIMO can also provide a Gaussian distribution when the transmit antenna number is high. In this case, LDHC coding can still be extended to MIMO. The corresponding structure and the representations are proposed in this thesis. In the matrix design of this code, short cycles are avoided inside codes, between demodulators and decoders, and also among different transmit antennas. By taking all these affects into account, the performance of the system can thus be improved. In MIMO cases, it is also shown that the adaptation by degree allocation is not necessary as the repetition part can be replaced by multiple receive antennas.

Bibliography

- [1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.
- [2] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, Jul., Oct., 1948.
- [3] —, “Communication in the presence of noise,” in *Pro. IRE*, vol. 37, 1949, pp. 10–21.
- [4] J. M. Wozencraft and I. M. Jacobs, *Principle of Communication Engineering*. New York: Wiley, 1965.
- [5] H. Imai and S. Hirakawa, “A new multilevel coding method using error correcting codes,” *IEEE Trans. Inf. Theory*, vol. 23, pp. 371–377, May 1977.
- [6] J. Leech and N. J. A. Sloane, “Sphere packing and error-correcting codes,” *Canad. J. Math.*, vol. 23, pp. 718–745, 1971.
- [7] G. Ungerböck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Inf. Theory*, vol. 28, pp. 55–67, Jan. 1982.
- [8] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes,” in *Proc. IEEE Int. Conf. Commun. (ICC '93)*, Geneva, Switzerland, May 23–26, 1993, pp. 1064–1070.
- [9] R. G. Gallager, “Low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

- [10] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low-density parity-check codes," vol. 33, no. 6, pp. 457–458, Mar. 1997.
- [11] S.-Y. Chung, J. D. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.
- [12] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.
- [13] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding using soft feedback," *Electron. Lett.*, vol. 34, no. 10, pp. 942–943, May 1998.
- [14] G. D. Forney, Jr. and G. Ungerboeck, "Modulation and coding for linear Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2384–2415, Oct. 1998.
- [15] G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Quershi, "Efficient modulation for band-limited channels," *IEEE J. Sel. Areas Commun.*, vol. SAC-2, no. 5, pp. 632–646, Sep. 1984.
- [16] G. D. Forney, Jr. and L.-F. Wei, "Multidimensional constellations—Part I: Introduction, figures of merit, and generalized cross constellations," *IEEE J. Sel. Areas Commun.*, vol. 7, pp. 877–892, Aug. 1989.
- [17] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable signaling on the Gaussian channel," *IEEE Trans. Inf. Theory*, vol. 36, pp. 726–740, Jul. 1990.
- [18] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 913–929, May 1993.
- [19] D. Raphaeli and A. Gurevitz, "Constellation shaping for pragmatic turbo-coded modulation with high spectral efficiency," *IEEE Trans. Commun.*, vol. 52, no. 3, Mar. 2004.
- [20] B. K. Khoo, S. Y. Le Goff, B. S. Sharif, and C. C. Tsimenidis, "Bit-interleaved coded modulation with iterative decoding using constellation shaping," *IEEE Trans. Commun.*, vol. 54, no. 9, Sep. 2006.

- [21] S. Y. Le Goff, B. K. Khoo, C. C. Tsimenidis, and B. S. Sharif, “Constellation shaping for bandwidth-efficient turbo-coded modulation with iterative receiver,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 6, Jun. 2007.
- [22] R. F. Fischer, *Precoding and Signal Shaping for Digital Transmission*. New York: John Wiley & Sons, Inc., 2002.
- [23] G. R. Lang and F. M. Longstaff, “A Leech lattice modem,” *IEEE J. Sel. Areas Commun.*, vol. 7, pp. 968–973, Aug. 1989.
- [24] P. Fortier, A. Ruiz, and J. M. Cioffi, “Multidimensional signal sets through the shell construction for parallel channels,” *IEEE Trans. Commun.*, vol. 40, pp. 500–512, Mar. 1992.
- [25] A. K. Khandani and P. Kabal, “Shaping multidimensional signal spaces—Part I: Optimum shaping, shell mapping,” *IEEE Trans. Inf. Theory*, vol. 39, pp. 1799–1808, Nov. 1993.
- [26] R. Laroia, N. Farvardin, and S. Tretter, “On optimal shaping of multidimensional constellations,” *IEEE Trans. Inf. Theory*, vol. 40, pp. 1044–1056, Jul. 1994.
- [27] G. D. Forney, Jr., “Trellis shaping,” *IEEE Trans. Inf. Theory*, vol. 38, pp. 281–300, Mar. 1992.
- [28] R. F. Fischer, J. B. Huber, and U. Wachsmann, “On the combination of multilevel coding and signal shaping,” in *ITG-Fachbericht: Codierung für Quelle, Kanal und Übertragung*, Aachen, Germany, Mar. 1998, pp. 273–278.
- [29] L. Duan, B. Rimoldi, and R. Urbanke, “Approaching the AWGN channel capacity without active shaping,” in *Proc. IEEE Int. Symp. Inform. Theory (ISIT’97)*, Ulm, Germany, Jun./Jul. 1997, p. 374.
- [30] X. Ma and Li Ping, “Coded modulation using superimposed binary codes,” *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3331–3343, Dec. 2004.

- [31] H. Schoeneich and P. A. Hoeher, "Adaptive interleave-division multiple access - A potential air interface for 4G bearer services and wireless LANs," in *Proc. International Conference on Wireless and Optical Communications and Networks (WOCN'04)*, Muscat, Oman, Jun. 2004.
- [32] P. A. Hoeher, H. Schoeneich, and J. C. Fricke, "Multi-layer interleave-division multiple access: Theory and practice," *European Transactions on Telecommunications (ETT)*, vol. 19, no. 5, Aug. 2008.
- [33] T. Wo and P. A. Hoeher, "Superposition mapping with application in bit-interleaved coded modulation," in *Proc. 8th International ITG Conference on Source and Channel Coding (SCC'10)*, Siegen, Germany, Jan. 18–21, 2010.
- [34] T. Wo, M. Noemm, D. Hao, and P. A. Hoeher, "Iterative processing for superposition mapping," *Journal of Electrical and Computer Engineering – Special Issue on Iterative Signal Processing in Communications*, vol. 2010, Article ID 706464, 13 pages, 2010.
- [35] P. A. Hoeher and T. Wo, "Superposition modulation: Myths and facts," *IEEE Commun. Mag.*, vol. 49, no. 12, Dec. 2011.
- [36] T. Wo, "Superposition mapping & related coding techniques," Ph.D. dissertation, University of Kiel, 2012.
- [37] T. Wo and P. A. Hoeher, "A universal coding approach for superposition mapping," in *Proc. 6th International Symposium on Turbo Codes & Iterative Information Processing*, Brest, France, Sep. 6–10, 2010.
- [38] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [39] S. ten Brink, "Convergence of iterative decoding," vol. 35, no. 10, pp. 806–808, May 1999, reprinted in vol. 35, no. 13, pp. 1117–1118, Jun. 1999.

- [40] F. Schreckenbach, N. Gortz, J. Hagenauer, and G. Bauch, "Optimization of symbol mappings for bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 7, pp. 593–595, Dec. 2003.
- [41] R. Schreckenbach, "Iterative decoding of bit-interlaved coded modulation," Ph.D. dissertation, Technical University of Munich, Germany, 2007.
- [42] K. Fukawa, S. Ormsub, A. Toelli, K. Anwar, and T. Matsumoto, "EXIT-constrained BICM-ID design using extended mapping," *EURASIP Journal on Wireless Communications and Networking*.
- [43] Q. Xie, Z. Yang, J. Song, and L. Hanzo, "EXIT-chart-matching-aided near-capacity coded modulation design and a BICM-ID design example for both Gaussian and Rayleigh channels," *IEEE Trans. Veh. Technol.*, vol. 62, pp. 1216 – 1227, Mar. 2013.
- [44] S. Pfletschinger and F. Sanzi, "Error floor removal for bit-interleaved coded modulation with iterative detection," *IEEE Trans. Commun.*, vol. 5, pp. 3174–3181, Nov. 2006.
- [45] S. ten Brink, "Rate one-half code for approaching the Shannon limit by 0.1 dB," *Electron Lett*, vol. 36, pp. 1293–1294, 2000.
- [46] D. Zhao, A. Dauch, and T. Matsumoto, "Modulation doping for repetition coded BICM-ID with irregular degree allocation," in *Proc. International ITG Workshop on Smart Antennas (WSA)*, Berlin, Germany, Feb. 2009.
- [47] M. L. Doelz, E. T. Heald, and D. L. Martin, "Binary data transmission techniques for linear systems," in *Proc. IRE*, vol. 45, May 1957, pp. 656–661.
- [48] P. A. Bello, "Selective fading limitations of the Kathryn modem and some system design considerations," *IEEE Trans. Commun. Technol.*, pp. 320–333, Sep. 1965.
- [49] G. C. Porter, "Error distribution and diversity performance of a frequency-differential PSK HF modem," *IEEE Trans. Commun. Technol.*, pp. 197–205, Apr. 1967.
- [50] I. Kalet, "The multitone channel," *IEEE Trans. Commun.*, pp. 119–124, Feb. 1989.

- [51] J. A. C. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," *IEEE Commun. Mag.*, pp. 5–14, May 1990.
- [52] T. J. Willink and P. H. Wittke, "Optimization and performance evaluation of multicarrier transmission," *IEEE Trans. Inf. Theory*, vol. 43, Mar. 1997.
- [53] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.
- [54] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using mult-element antennas," *Bell Labs Tech. J*, pp. 41–59, 1996.
- [55] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, pp. 311–335, 1998.
- [56] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communications: performance criterion and code construction," *IEEE Trans. Inf. Theory*, vol. 44, pp. 744–765, Mar. 1998.
- [57] S. M. Alamouti, "A simple transmit diversity techniques for wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.
- [58] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc., 2006.
- [59] P. A. Hoeher and H. Schoeneich, "Interleave-division multiple access from a multiuser point of view," in *Proc. Int. Symp. on Turbo Codes & Related Topics in conjunction with Int. ITG Conf. on Source and Channel Coding*, Munich, Germany, Apr. 2006.
- [60] H. Schoeneich, "Adaptiver Interleave-Division Mehrfachzugriff (IDMA) mit Anwendung in der Mobilfunkkommunikation," Ph.D. dissertation, University of Kiel, Germany, 2007.

- [61] P. A. Hoeher and W. Xu, “Multi-layer interleave-division multiple access for 3GPP long term evolution,” in *Proc. IEEE Int. Conf. Commun. (ICC’07)*, Glasgow, Scotland, Jun. 2007.
- [62] J. Tong, Li Ping, and X. Ma, “Superposition coded modulation with peak-power limitation,” *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2562–2576, Jun. 2009.
- [63] Li Ping, J. Tong, X. Yuan, and Q. Guo, “Superposition coded modulation and iterative linear MMSE detection,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 995–1004, Aug. 2009.
- [64] D. Hao and P. A. Hoeher, “Iterative estimation and cancellation of clipping noise for multi-layer IDMA systems,” in *Proc. Int. ITG Conf. on Source and Channel Coding*, Ulm, Germany, Jan. 2008.
- [65] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2001.
- [66] J. Hagenauer, “The turbo principle: Tutorial introduction and state of art,” in *Proc. International Symposium on Turbo Codes & Iterative Information Processing*, Brest, France, Sep. 1997, pp. 1–11.
- [67] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.
- [68] W. Koch and A. Baier, “Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference,” in *Proc. IEEE GLOBECOM 1990*, Dec. 1990, pp. 1679–1685.
- [69] P. Robertson, E. Villebrun, and P. A. Hoeher, “A comparison of optimum and sub-optimum MAP decoding algorithms operating in the log domain,” in *Proc. IEEE Int. Conf. on Communications (ICC ’95)*, Jun. 1995, pp. 1009–1013.
- [70] J. Hagenauer and P. A. Hoeher, “A Viterbi algorithm with soft-decision outputs and its applications,” in *Proc. IEEE GLOBECOM 1989*, Nov. 1989, pp. 1680–1686.

- [71] G. Bauch and V. Franz, "A comparison of soft-in/soft-out algorithms for 'turbo-detection'," in *Proc. Int. Conf. Telecomm.*, Jun. 1998, pp. 259–263.
- [72] T. J. Richardson, M. A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular LDPC codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [73] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [74] J. Hou, P. H. Siegel, L. B. Milstein, and H. D. Pfister, "Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2141–2155, Sep. 2003.
- [75] B. Lu, G. Yue, and X. Wang, "Performance analysis and design optimization of LDPC-coded MIMO OFDM systems," *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 348–361, Feb. 2004.
- [76] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, Dec. 1997.
- [77] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.
- [78] I. Land, P. A. Hoeher, and S. Gligorević, "Computation of symbol-wise mutual information in transmission systems with LogAPP decoders and application to EXIT charts," in *Proc. 5th International ITG Conference on Source and Channel Coding (SCC'04)*, Erlangen-Nürnberg, Germany, Jan. 14–16, 2004.
- [79] E. Sharon, A. Ashikhmin, and S. Litsyn, "EXIT functions for the Gaussian channels," in *Proc. 40th Annu. Allerton Conf. on Communications, Control, Computers*, Allerton, IL, Oct. 2003, pp. 972–981.

- [80] —, “EXIT functions for binary input memoryless symmetric channels,” *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1207–1214, Jul. 2006.
- [81] M. Noemm, A. Mourad, and P. A. Hoeher, “Superposition modulation with irregular convolutional coding,” in *Proc. IEEE GLOBECOM 2012*, Anaheim, USA, Dec. 2012.
- [82] Z. Shi, T. Wo, and P. A. Hoeher, “Superposition mapping with adaptive bit loading for BICM-OFDM systems,” in *Proc. International Symposium on Turbo Codes & Iterative Information Processing*, Brest, France, Sep. 2010.
- [83] Z. Shi and P. A. Hoeher, “Low-density hybrid-check coded superposition mapping in BICM-OFDM,” in *Proc. International Symposium on Turbo Codes & Iterative Information Processing*, Gothenburg, Sweden, Aug. 2012.
- [84] R. G. Gallager, *Information Theory and Reliable Communication*. Wiley, New York, 1968.
- [85] Y. G. Li and G. L. Stüber, Eds., *Orthogonal Frequency Division Multiplexing for Wireless Communications*. Springer, 2006.
- [86] D. Hughes-Hartogs, “Ensemble modem structure for imperfect transmission media,” U.S. Patent 4,679,227 (July, 1987), 4,731,816 (Mar. 1988), 4,833,706 (May 1989).
- [87] J. Campello, “Practical bit loading for DMT,” in *Proc. IEEE International Conference on Communications (ICC '99)*, Vancouver, June 1999.
- [88] H. Levin, “A complete and optimal data allocation method for practical discrete multitone systems,” in *Proc. IEEE GLOBECOM 2001*, vol. 1, Nov. 2001, pp. 369–374.
- [89] P. S. Chow, J. M. Cioffi, and J. A. Bingham, “A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels,” *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 773–775, Feb. 1995.
- [90] R. F. Fischer and J. B. Huber, “A new loading algorithm for discrete multitone transmission,” in *Proc. IEEE GLOBECOM 1996*, London, Nov. 1996.

- [91] Y. Li and W. E. Ryan, “Mutual-information-based adaptive bit-loading algorithms for LDPC-coded OFDM,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, May 2007.
- [92] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [93] R. D. Shachter, “Probabilistic inference and influence diagrams,” *Operations Research*, vol. 36, no. 4, pp. 589–604, Aug. 1988.
- [94] G. Shafer and P. Shenoy, “Probability propagation,” *Ann. Mat. Art. Intell.*, vol. 2, pp. 327–352, 1990.
- [95] J. Yedidia, W. Freeman, and Y. Weiss, “Generalized belief propagation,” *Advances Neural Information Processing Systems (NIPS)*, vol. 13, pp. 689–695, Dec. 2000.
- [96] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [97] N. Wiberg, “Codes and decoding on general graphs,” Ph.D. dissertation, Linköping University, Sweden, 1996.
- [98] S. M. Aji and R. J. McEliece, “The generalized distributive law,” *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 325–343, Mar. 2000.
- [99] G. D. Forney, “Codes on graphs: Normal realizations,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 520–548, Feb. 2001.
- [100] D. J. C. MacKay, “Good error correcting codes based on very sparse matrices,” *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 399–431, Mar. 1999.
- [101] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, “Construction of irregular ldpc codes with low error floors,” in *Proc. IEEE Int. Conf. Commun. (ICC’03)*, Anchorage, Alaska, USA, May 11–15, 2003.
- [102] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

- [103] S. H. Müller-Weinfurtner, “Coding approaches for multiple antenna transmission in fast fading and OFDM,” *IEEE Trans. Signal Process.*, vol. 50, no. 10, pp. 2442–2450, Oct. 2002.
- [104] T. Wo, J. C. Fricke, and P. A. Hoeher, “A graph-based iterative Gaussian detector for frequency-selective MIMO channels,” in *Proc. IEEE Information Theory Workshop (ITW’06)*, Chengdu, China, Oct. 2006.
- [105] T. Wo and P. A. Hoeher, “Low-complexity Gaussian detector for MIMO systems,” *Journal of Electrical and Computer Engineering*, vol. 2010, Article ID 706464, 13 pages, 2010.
- [106] J. C. Fricke, M. Sandell, J. Mietzner, and P. A. Hoeher, “Impact of the Gaussian approximation on the performance of the probabilistic data association MIMO decoder,” vol. 2005, no. 5, pp. 796–800, Dec. 2005.
- [107] X. Wang and H. V. Poor, “Iterative (turbo) soft interference cancellation and decoding for coded CDMA,” *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.
- [108] M. Tüchler, R. Kötter, and A. Singer, “Turbo equalization: Principle and new results,” *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, May 2002.
- [109] H. El Gamal and A. Hammons, “A new approach to layered space-time coding and signal processing,” *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2321–2334, Sep. 2001.
- [110] D. Seethaler, G. Matz, and F. Hlawatsch, “An efficient MMSE-based demodulation for MIMO bit-interleaved coded modulation,” in *Proc. IEEE GLOBECOM 2004*, Dallas, USA, Dec. 2004, pp. 2455–2459.
- [111] M. Butler and I. Collings, “A zero-forcing approximate log-likelihood receiver for MIMO bit-interleaved coded modulation,” *IEEE Commun. Lett.*, vol. 8, no. 2, pp. 105–107, Feb. 2004.

- [112] M. Witzke, S. Bärö, F. Schreckenbach, and J. Hagenauer, “Iterative detection of MIMO signals with linear detectors,” in *Proc. Signals, Systems and Computers*, Pacific Grove, USA, Nov. 2002, pp. 289–293.
- [113] H. V. Poor and S. Verdu, “Probability of error in MMSE multiuser detection,” *IEEE Trans. Inf. Theory*, vol. 43, no. 3, pp. 858–871, May 1997.

Own Publications

- [1] C. Knievel, Z. Shi, P. A. Hoeher, and G. Auer, “2D graph-based soft channel estimation for MIMO-OFDM,” in *Proc. IEEE International Conference on Communications*, Cape Town, South Africa, May 2010.
- [2] Z. Shi, T. Wo, and P. A. Hoeher, “Superposition mapping with adaptive bit loading for BICM-OFDM systems,” in *Proc. International Symposium on Turbo codes & Iterative Information Processing*, Brest, France, Sep. 2010.
- [3] Z. Shi, T. Wo, P. A. Hoeher, and G. Auer, “Graph-based soft iterative receiver for higher-order modulation,” in *Proc. 12th IEEE International Conference on Communication and Technology*, Nanjing, China, Nov. 2010.
- [4] Z. Shi and P. A. Hoeher, “Low-density hybrid check coded superposition mapping in BICM-OFDM,” in *Proc. International Symposium on Turbo codes & Iterative Information Processing*, Gothenburg, Sweden, Aug. 2012.
- [5] T. Wo, P. A. Hoeher, and Z. Shi “Graph-based soft channel estimation for fast fading channels,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 12, pp. 4243-4251, Dec. 2012.
- [6] Z. Shi, P. A. Hoeher, and A. A. Saed “Graph-based bit-wise soft channel estimation for superposition mapping,” *Journal of Intelligent Computing*, vol. 4, issue 1, pp. 17-22, Mar. 2013.

Appendix A

List of Abbreviations

APP	A posteriori probability
AWGN	Additive white Gaussian noise
BICM	Bit-interleaved coded modulation
BPSK	Binary phase shift keying
DEC	Decoder
DEM	Demodulator
EPA	Equal power allocation
EXIT	Extrinsic information transfer
GA	Gaussian approximation
GPA	Grouped power allocation
IDM	Interleave-division multiplexing
LDHC	Low-density hybrid-check
LDPC	Low-density parity-check
LLR	Log likelihood ratio
MAI	Multiple antenna interference
MLC	Multilevel coding
ML-IDMA	Multi-layer interleave-division multiple access
MMSE	Minimum mean square error
MN	Mapping node
MI	Mutual information

MIMO	Multiple-input multiple-output
OFDM	Orthogonal frequency-division multiplexing
PCN	Parity-check node
PDF	Probability density function
PSK	Phase shift keying
PSM	Phase-shifted superposition mapping
QAM	Quadrature amplitude modulation
SISO	Single-input single-output
SM	Superposition mapping
SM-EPA	Superposition mapping with equal power allocation
SM-GPA	Superposition mapping with grouped power allocation
SNR	Signal-to-noise ratio
TCM	Trellis-coded modulation
TS	Trellis shaping
UPA	Unequal power allocation
VN	Variable node
ZF	Zero forcing

Appendix B

List of Symbols

b	Info bit
d	Bipolar bit
c	Code bit
h	Channel coefficient
k	Time index
p	Parity-check bit
t	Transmit antenna index
r	Receive antenna index
u	Information bit
x	Transmitted symbol
y	Received symbol
z	Additive white Gaussian noise
α	Weighting factor
π	Interleaver
BW	Bandwidth Efficiency
C	Capacity
E_b	Energy per bit
E_s	Energy per symbol
G	Group size

I	Mutual information
L	Number of levels
M	Symbol alphabet
N	Number of bits per symbol
N_T	Number of transmit antennas
N_R	Number of receive antennas
N_0	Noise spectral density
R	Code rate