

## Research Article

# Iterative Processing for Superposition Mapping

**Tianbin Wo, Meelis Noemm, Dapeng Hao, and Peter Adam Hoehner**

*Information and Coding Theory Laboratory, University of Kiel, Kaiserstrasse 2, 24143 Kiel, Germany*

Correspondence should be addressed to Tianbin Wo, [wtb@tf.uni-kiel.de](mailto:wtb@tf.uni-kiel.de)

Received 12 March 2010; Accepted 27 June 2010

Academic Editor: Christian Schlegel

Copyright © 2010 Tianbin Wo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Superposition mapping (SM) is a modulation technique which loads bit tuples onto data symbols simply via linear superposition. Since the resulting data symbols are often Gaussian-like, SM has a good theoretical potential to approach the capacity of Gaussian channels. On the other hand, the symbol constellation is typically nonbijective and its characteristic is very different from that of conventional mapping schemes like QAM or PSK. As a result, its behavior is also quite different from conventional mapping schemes, particularly when applied in the framework of bit-interleaved coded modulation. In this paper, a comprehensive analysis is provided for SM, with particular focus on aspects related to iterative processing.

## 1. Introduction

According to Shannon's information theory, the capacity of a Gaussian channel corresponds to the maximum mutual information between channel inputs and outputs [1, 2]. Given a power constraint, this maximum can be achieved if and only if the channel inputs are Gaussian distributed. Since Shannon's seminal work in 1948, many approaches have been proposed to map binary bits onto Gaussian-like distributed symbols. Among these approaches, the most well-known ones are Huffman decoding [3] and signal shaping [4]. By using a Huffman decoder to map bits onto a signal constellation with a well-designed distribution, for example, the Maxwell-Boltzmann distribution [5], one can maximize the mutual information between channel input and output for a given symbol cardinality and average power. Unfortunately, this approach demands variable-rate transmission and is consequently undesirable for practical applications. In contrast, signal shaping techniques are fixed-rate transmission schemes and for this reason have attracted interest in the field of real-world implementations. There are two popular methods for signal shaping, namely trellis shaping [6] and shell mapping [7, 8]. Both trellis shaping and shell mapping share a common idea, that is to construct a high-dimensional uniform constellation which results in low-dimensional nonuniform constituent constellations. They are both able to deliver a shaping gain of about

1.0 dB [9] without any additional effort with respect to channel coding, but subject to the assumption of noniterative demapping and decoding. In case of iterative soft-in soft-out (SISO) demapping and decoding, the block-wise mapping manner inherent in popular shaping techniques presents a bottleneck to the design of receiver algorithms.

To overcome the drawbacks of Huffman decoding and signal shaping, several researchers proposed different transmission schemes employing linear superposition to generate Gaussian-like symbols [10–12]. Without loss of generality, we denote the key component of such techniques as *superposition mapping* (SM). The characteristic feature of SM is that the conversion from binary bits to data symbols is done by a certain form of superposition instead of bijective (one-to-one) mapping. Since the output of a superposition mapper is often Gaussian-like, the necessity of doing active signal shaping is eliminated. On the other hand, superimposed signal components interfere with each other, and the resulting relationship between bit tuples and data symbols is often nonbijective. To guarantee a perfect reconstruction at the receiver side, channel coding and interleaving are typically mandatory, and iterations between the decoder and demapper are essential. This incurs a structure well known as bit-interleaved coded modulation (BICM) [13]. For an easy reference, we may call such a transmission technique bit-interleaved coded modulation with superposition mapping (BICM-SM).

In this paper, we provide a comprehensive study on the performance of BICM-SM, with particular focus on aspects related to iterative demapping and decoding. Theoretical as well as numerical results are provided. Via mutual information analysis, the capacity-achieving potential of superposition mapping is demonstrated. Due to the superimposed signal structure and the nonuniform signal distribution, the requirements of a superposition mapper on the channel code are essentially different from that of a uniform bijective mapper like PSK/QAM. Through EXIT chart analysis, we will show that repetition coding is in fact superior for coded SM systems. In addition to these analytical investigations, two practical issues are also considered. A novel tree-pruning concept is proposed to reduce the demapping complexity, and a baseband clipping scheme together with an iterative soft compensation mechanism is presented to reduce the signal peak-to-average power ratio (PAPR), while keeping the system performance degradation at an acceptable level.

The remainder of this paper is organized as follows. Section 2 introduces the Gaussian channel model under investigation. Section 3 provides a brief description of superposition mapping and related information theoretical aspects. Section 4 treats the application of superposition mapping in the framework of bit-interleaved coded modulation. In Section 5, an extensive study is carried out on iterative demapping and decoding process of coded SM systems. Particular focus is put on the convergence behavior of the iterative receiver given different channel codes. In Section 6, several approaches for low-complexity demapping are discussed, and in Section 7 a novel method for PAPR control is proposed. Finally, Section 8 summarizes the paper and suggests some interesting topics for future work.

## 2. Gaussian Channel

The additive white Gaussian noise (AWGN) channel is perhaps the most important channel model with continuous outputs. Though simple, it models the fundamental effects of communication in a noisy environment. The discrete-time complex AWGN channel model can be written as

$$y[k] = x[k] + z[k], \quad z[k] \sim \mathcal{CN}(0, \sigma_z^2), \quad (1)$$

where  $k$  is the time index. The mutual information between the channel input and output is given by

$$I(x; y) = h(y) - h(y | x) = h(y) - h(z), \quad (2)$$

where  $h(\cdot)$  denotes differential entropy. Since the normal distribution maximizes the entropy for a given variance, the maximum of  $I(x; y)$  is achieved when  $y$  is Gaussian distributed. Strictly speaking,  $y$  can only be Gaussian if  $x$  is Gaussian. In practice, however, it often suffices if  $x$  is Gaussian-like, given a reasonable signal-to-noise ratio (SNR).

## 3. Superposition Mapping

Figure 1 shows the general structure of superposition mapping. After serial-to-parallel (S/P) conversion, each code bit

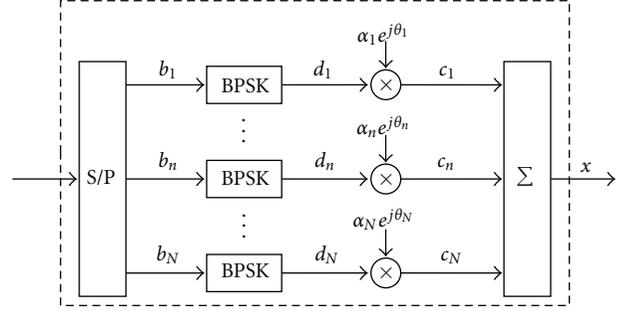


FIGURE 1: Superposition mapping (SM).

$b_n$  is first converted into a binary antipodal symbol  $d_n$ . (Due to superposition mapping, binary antipodal component symbols bring no limit with respect to the overall bandwidth efficiency.) Then, power allocation and phase allocation are performed. Afterwards, the complex component symbols  $c_n$  (called “chips”) are linearly superimposed to create a finite-alphabet data symbol. SM can be described by the following equation:

$$x = \sum_{n=1}^N c_n = \sum_{n=1}^N d_n \alpha_n e^{j\theta_n}, \quad d_n \in \{\pm 1\}, \quad (3)$$

where  $N$  is called the bit load of the superposition mapper. By tuning the amplitudes  $\alpha_n$  and the phases  $\theta_n$ , different superposition mapping schemes can be obtained. In case of equal power allocation, that is, when all amplitudes  $\alpha_n$  are the same, all bits  $b_n$  are equally protected. It was shown in [14] that, given a BICM system with standard coding approaches, direct superposition mapping (DSM) with equal power allocation suffers from limited supportable bandwidth efficiency, and DSM with unequal power allocation suffers from power efficiency degradation. In contrast, phase-shifted superposition mapping (PSM) provides virtually unlimited supportable bandwidth efficiency and desirable power efficiency. For this reason, this paper will exclusively focus on PSM, while most of the analysis can easily be extended to the case of DSM.

**3.1. Phase-Shifted Superposition Mapping.** PSM is characterized by the following power and phase allocation:

$$\begin{aligned} \alpha_n &= \alpha \quad \text{for } 1 \leq n \leq N, \\ \theta_n &= \frac{\pi(n-1)}{N} \quad \text{for } 1 \leq n \leq N. \end{aligned} \quad (4)$$

All chips are allocated with the same magnitude, and each chip is allocated with a unique phase uniformly drawn over the interval  $[0, \pi)$ . Substituting (4) into (3), we obtain

$$\begin{aligned} \text{Re}\{x\} &= \sum_{n=1}^N \text{Re}\{c_n\} = \sum_{n=1}^N \alpha d_n \cos\left(\frac{\pi(n-1)}{N}\right), \\ \text{Im}\{x\} &= \sum_{n=1}^N \text{Im}\{c_n\} = \sum_{n=1}^N \alpha d_n \sin\left(\frac{\pi(n-1)}{N}\right). \end{aligned} \quad (5)$$

Since  $|c_n|^2 \equiv \alpha^2$ , the total energy spent for each bit is constant. Hence, PSM is characterized by a high power efficiency provided by superposition mapping with equal power allocation. On the other hand, from (5) one can see that unequal power allocation is actually done in the real and imaginary dimension, respectively, due to the individual phase shift. This increases the cardinality of the output symbol and greatly enhances the supportable bandwidth efficiency. As a consequence, PSM does not suffer from limited supportable bandwidth efficiency.

Figure 2 illustrates the PSM constellation for different bit loads, and Figure 3 shows the probability distribution of one quadrature component of the PSM outputs. For  $N = 4$ , the PSM constellation looks like circular 16-QAM, while for larger  $N$ , the constellation points tend to be geometrically Gaussian-like distributed. (The central limit theorem cannot be applied here, since  $\{c_1, \dots, c_N\}$  are independent but not identically distributed due to individual phase shifts.) It can be easily proven that the real part and the imaginary part of  $x$  are statistically independent. Therefore, the constellation points of PSM are approximately circular Gaussian distributed, though in no case perfect.

**3.2. An Information Theoretical View.** Given the distribution of  $x$ , the mutual information (MI)  $I(x; y)$  can be numerically evaluated. Figure 4 provides MI curves for PSM as well as for square QAM. Comparing Figure 4(a) with Figure 4(b), it can be seen that PSM outperforms QAM in the left region, and it is almost capacity achieving, which is due to the Gaussian-like symbol distribution. The gap between the capacity curve and the slope for high-order QAM in Figure 4(b) corresponds to the ultimate shaping gain [9]. On the other hand, PSM is worse than QAM in the right region, which is due to a smaller minimum distance between distinct constellation points. Last but not least, in most cases, PSM and QAM provide roughly the same entropy per symbol, given an identical bit load  $N$ , which can be seen by comparing the smooth sections of the MI curves. An important message from the above observations is that, with PSM, signal shaping is no longer necessary to approach the capacity of the Gaussian channel. As a matter of fact, the mutual information given PSM input symbols can be arbitrarily close to the capacity at any SNR, as long as the bit load  $N$  is large enough.

**3.3. Rate Limit of PSM.** Following (1), the limit of coding rate for PSM transmission over the AWGN channel is given by

$$R \leq \frac{I(x; y)}{N} \leq \frac{H(x)}{N} \approx 1. \quad (6)$$

Hence, the bandwidth efficiency of PSM is virtually unlimited and the required spreading factor ( $1/R$ ) should not be very large. The bandwidth efficiency limit at about 2 bits/symbol per signal dimension reported in [15] is in fact due to the adopted Gaussian-approximation-based demapping algorithm rather than the mapping scheme itself.

## 4. Bit-Interleaved Coded Modulation with Superposition Mapping

Bit-interleaved coded modulation [13] is widely recognized as a promising technique to approach the channel capacity at high SNRs. Typically, higher-order QAM is used to achieve a high bandwidth efficiency, while an interleaver is placed between the encoder and the signal mapper to exploit the bit diversity of QAM. With the same structure but replacing QAM with superposition mapping (SM), we obtain the BICM-SM transmission scheme under investigation.

**4.1. General Structure.** Figure 5 shows the transmitter and receiver structure of BICM-SM, where SD stands for superposition demapping. Since a superposition mapper is often a nonbijective (many-to-one) mapper, iterations between the demapper and the decoder are not only necessary but mandatory. A superposition demapper usually can not work properly without utilizing the redundancy introduced by the channel code. This is quite different in comparison with the demapper of conventional mapping schemes, like a PSK/QAM demapper, which does not necessarily have to cooperate with the channel decoder.

**4.2. Soft-In Soft-Out Demapping.** The superposition demapper needs to provide soft decisions of each code bit given the channel outputs and the a priori information from the decoder. Given an AWGN channel, an APP soft-input soft-output (SISO) demapper for SM can be described as follows. Let

$$\mathbf{b}_{\sim n} \doteq [b_1, b_2, \dots, b_{n-1}, b_{n+1}, \dots, b_N] \quad (7)$$

collect the bits excluding  $b_n$  loaded on one symbol, we have

$$\begin{aligned} \text{LLR}(b_n) &\doteq \ln \frac{p(y | b_n = 0)}{p(y | b_n = 1)} \\ &= \ln \frac{\sum_{\mathbf{b}_{\sim n}} p(y | b_n = 0, \mathbf{b}_{\sim n}) P(\mathbf{b}_{\sim n})}{\sum_{\mathbf{b}_{\sim n}} p(y | b_n = 1, \mathbf{b}_{\sim n}) P(\mathbf{b}_{\sim n})}, \end{aligned} \quad (8)$$

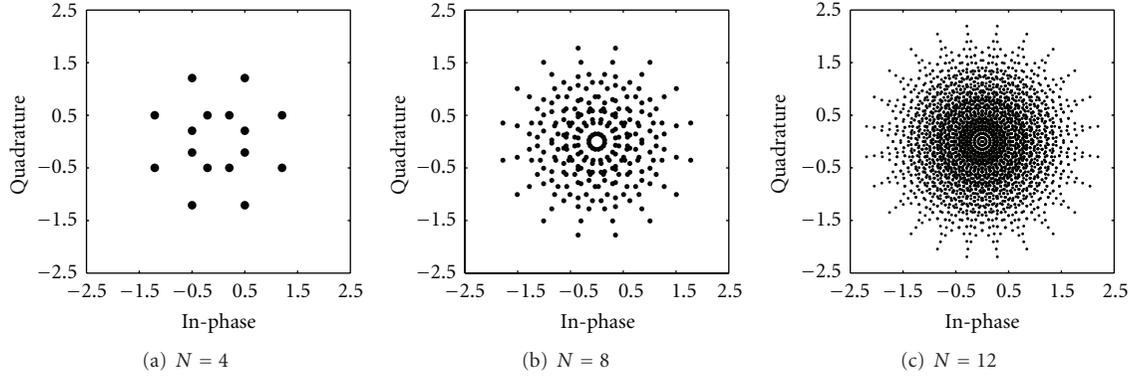
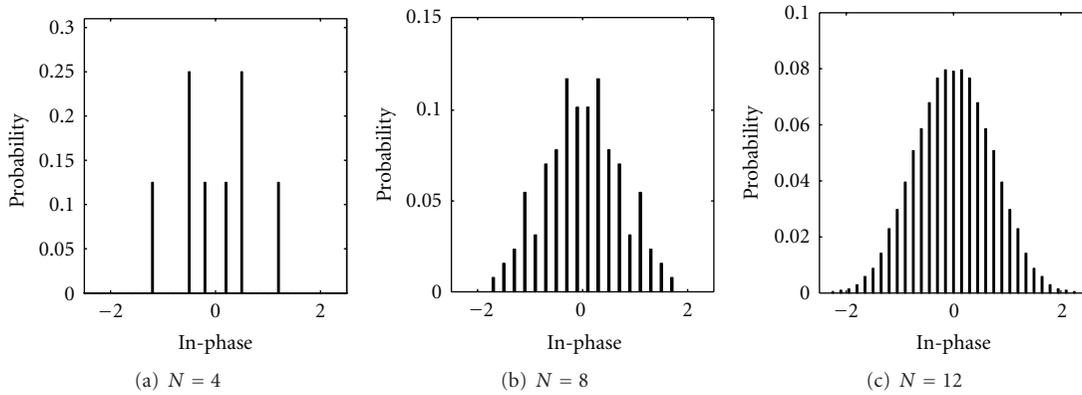
where

$$P(\mathbf{b}_{\sim n}) = \prod_{i=1, i \neq n}^N P(b_i), \quad (9)$$

assuming statistical independence of interleaved code bits. The APP demapping algorithm, in spite of its high complexity, is the first choice for theoretical analyses, as it provides the best performance. In the following measurements, APP demapping is applied, if not specifically addressed.

## 5. Iterative Processing for Superposition Mapping

Due to the geometrically nonuniform symbol distribution, we observe that PSM is better than square QAM in the sense of mutual information over the Gaussian channel. As long as the bit load  $N$  is large enough, PSM can efficiently

FIGURE 2: Constellation diagrams for PSM,  $\alpha = \sqrt{1/N}$ .FIGURE 3: Distribution of  $\text{Re}\{x\}$  for PSM,  $\alpha = \sqrt{1/N}$ .

fill up the 1.53 dB gap, well known as the ultimate shaping gain in the terminology of signal shaping, between the channel capacity and the mutual information with uniform signalling. Nevertheless, due to linear superposition with equal power allocation, the behavior of a PSM demapper, in the scenario of iterative demapping and decoding is very different from that of a demapper of conventional mapping schemes like PSK/QAM. This in fact implies some different concepts for the code design. In this section, we will have an in-depth study on the iterative demapping and decoding process of coded PSM systems via an EXIT chart analysis [16, 17].

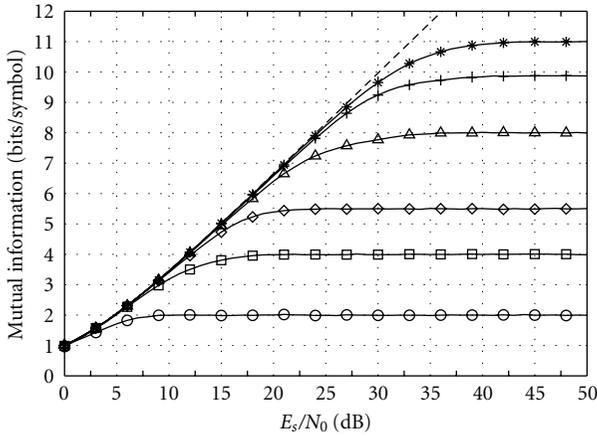
*5.1. First Impression on the Performance.* To start, let us first have a look at the performance of coded PSM. We consider two type of codes, low-density parity check (LDPC) codes, which are known to be capacity-achieving for binary-input Gaussian channels, and repetition codes, considered to be weak codes as they provide no coding gain at all on the AWGN channel. As shown in Figure 6, the true situation is in fact opposite to the common understanding, that is, repetition-coded PSM can even outperform LDPC-coded PSM, particularly at large bit loads  $N$ . Another noteworthy phenomenon is that the performance of LDPC-coded PSM degrades severely as the bit load  $N$  increases. All these

observations indicate that the classical purely parity-check-based channel codes do not really fit with a superposition mapper, which is to be studied in the next sections.

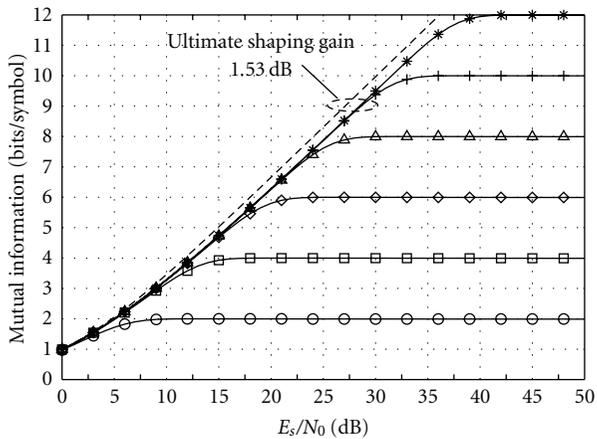
*5.2. MI Transfer Characteristics of APP Demapper.* To start an analysis on PSM, we will have a look at the mutual information (MI) transfer characteristics of the SISO APP demapper. Square QAM with Gray labeling will be used as reference for comparison.

Let us begin the discussion with the starting points in the left side of Figure 7. Compared to PSM, QAM with Gray labeling has considerably higher starting points, given the same bit load. The starting points show how reliable will be the output of the demapper given only the channel output and no a priori information from the decoder. For PSM with large  $N$ , the starting points are very low. Hence, the demapper output in the first iteration(s) is very weak. This comes from the fact that the constellation diagram, Figure 2, is densely populated in the central region, which leads to a reduced minimum distance and in some cases even to nonbijective mapping. This effect can be clearly observed in Figure 7(b) for both PSM and QAM, where the starting points improve as the SNR increases.

The slope of the curves presents the most obvious difference. For QAM with Gray labeling, the curves are



(a) PSM



(b) QAM

FIGURE 4: Mutual information of PSM/QAM over the AWGN channel.

more or less horizontal, with modest slopes for  $N \geq 8$ . On the other hand, for PSM the curves have steep slopes, excluding  $N = 2$ . For  $N > 6$ , the curves are even convex. The slope of the curves characterizes the importance or usefulness of iterations. Hence in case of PSM, as the feedback from the decoder becomes more reliable from iteration to iteration, the demapper output improves. On the contrary for QAM, usually only a few iterations are sufficient and the gain by feedback of extrinsic information is noticeably smaller.

At last we come to the ending points of the MI transfer plots. Figure 7(a) presents an interesting fact. In the case of PSM, given the same  $E_c/N_0$ , for all selections of  $N$ , the curves end in the same point. This relates to the fact that all code bits are converted to binary antipodal symbols. If the feedback information is very reliable, then we are left with a detection scenario for BPSK transmission over a Gaussian channel. For QAM, the ending points differ significantly and for  $N > 2$

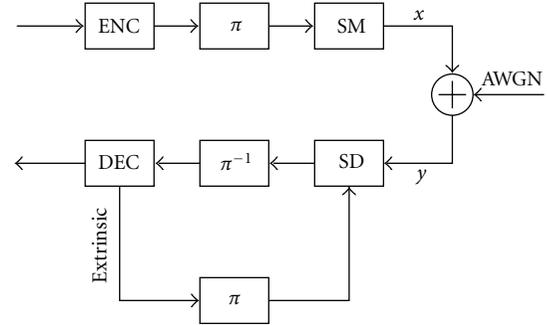


FIGURE 5: Bit-interleaved coded modulation with superposition mapping.

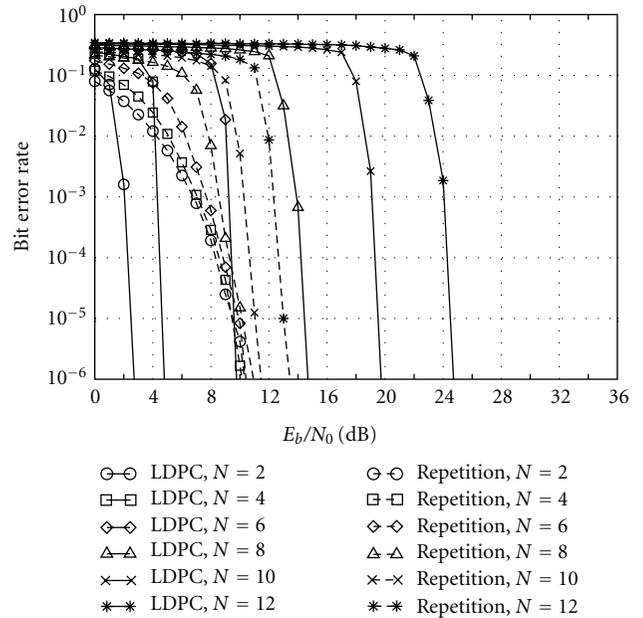


FIGURE 6: BER performance of PSM with rate 1/2 (3,6)-regular LDPC codes and rate 1/2 regular repetition codes.

the points are always lower than the corresponding ones for PSM. This can be explained by the fact that for QAM bits are unequally protected.

**5.3. Convergence Behavior of Coded PSM.** For coded modulation, an important issue is the convergence of the iterative demapping and decoding process. Whether an iterative receiver can converge or not is determined by the channel SNR and, more importantly, by the relation between the decoder characteristics and the demapper characteristics. As a commonly used tool, an EXIT chart can elegantly demonstrate the suitability of a channel code for a given signal mapper. By means of an EXIT chart analysis, we will see that there is indeed a strong reason for the good performance of repetition-coded PSM systems, and we will also see that the requirements of superposition mapping on channel codes are essentially different with respect to conventional mapping schemes.

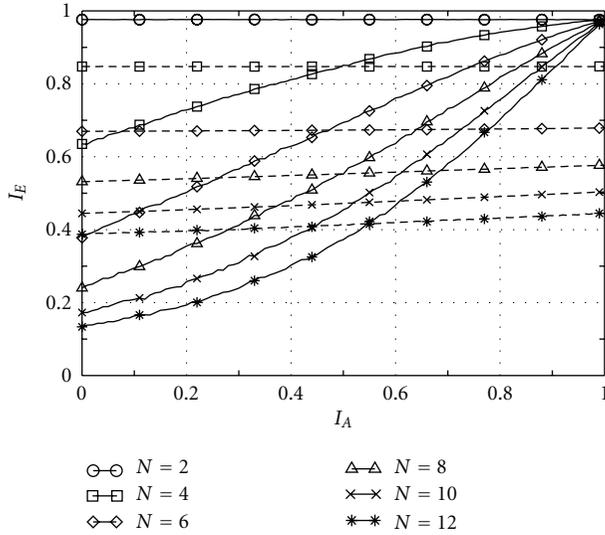
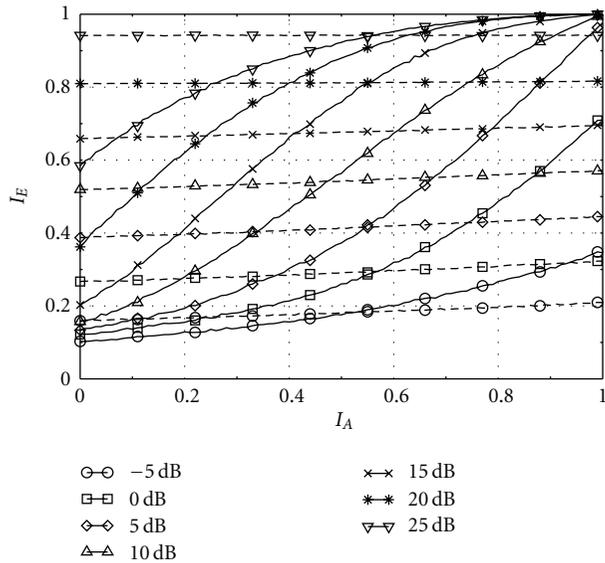
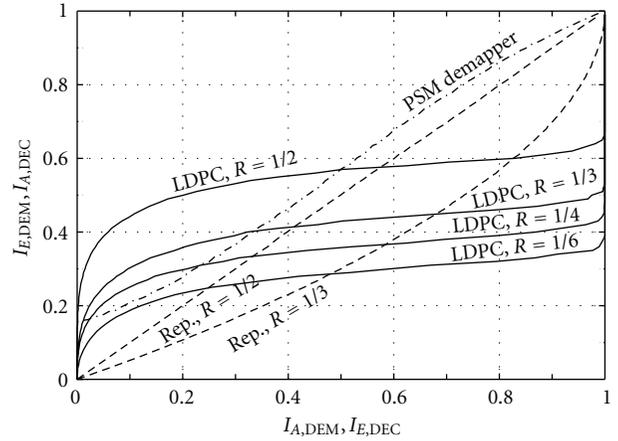
(a)  $E_c/N_0 = 5$  dB(b)  $N = 12$ 

FIGURE 7: MI transfer characteristics of PSM demapper (solid lines) and that of Gray labeled QAM demapper (dashed lines).

Let us assume that the SNR per code bit is  $E_c/N_0 = 10$  dB and the bit load is  $N = 12$ . Given this setup, the MI transfer curve of a PSM demapper is plotted as a dashed dot curve in Figure 8. The MI transfer curves for several decoders are also plotted, with the horizontal and vertical axes swapped. To guarantee convergence, the MI transfer curve of the decoder must be below the MI transfer curve of the demapper, so as to open the tunnel for iterative refining of the soft decisions. Otherwise, the iterative process will get stuck at a certain point and no additional gain is achievable by further iterations.

As one can see from Figure 8, given a regular LDPC code with column weight 3, one needs to drop the code rate to as low as 1/6 in order to open the tunnel. The main problem

FIGURE 8: EXIT chart for coded PSM,  $N = 12$ ,  $E_c/N_0 = 10$  dB.

is in the left region. As already explained in Section 5.2, the MI transfer curve of a PSM demapper with large  $N$  starts from a rather low point but ends at a rather high point. Meanwhile, the MI transfer curve of an LDPC code always possesses a wide quasi-flat section, mainly due to the nature of parity checks. This situation makes the tunnel difficult to open in the left region but often unnecessarily wide in the right region, which clearly explains the poor performance of LDPC-coded PSM at large bit loads  $N$ , as shown in Figure 6. As a matter of fact, the classical strong codes, such as Turbo codes and LDPC codes, purely built upon parity checks, do not fit with PSM very well. To fully exploit the capacity-achieving potential of superposition mapping, new codes and new code design concepts are necessary. Nevertheless by checking Figure 7(b) carefully, one recognizes that the MI transfer characteristics of LDPC codes perfectly match with that of QAM, which explains the excellent performance of LDPC-coded QAM systems.

In contrast to parity-check-based codes, a simple rate 1/2 repetition code can already effectively open the tunnel for convergence, which can also be validated by the good performance of repetition-coded PSM in Figure 6. Repetition codes come with zero coding gain, in the sense that lowering the ending point of the MI transfer curve is only possible via an equivalent amount of degradation in the  $E_c/N_0$ . Nevertheless, due to the special property of PSM, this issue does not present a big problem. For reasonable SNRs, the MI transfer curves of PSM always end at a very high point. Note that, for PSM with large bit loads, the primary task of channel coding is no longer to combat the additive noise but to guarantee the separation of superimposed binary chips. As long as all chips are perfectly separated, the BER performance of PSM will asymptotically approach that of BPSK, while the performance of uncoded BPSK is already very good at  $E_c/N_0 = 10$  dB. Above all, the difficult part is the first few iterations. A decoder has to be able to deliver reliable feedback given very weak inputs in order to let the iterative demapping and decoding go on properly. With respect to this concern, however, a repetition code is the best choice. One may call the checks of a repetition code as equality checks.

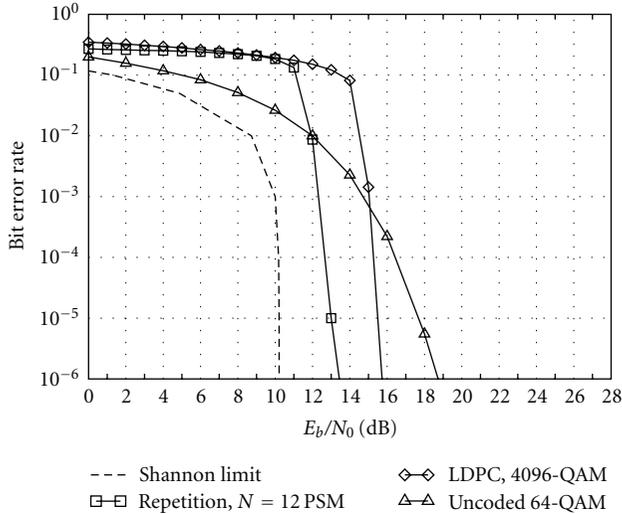


FIGURE 9: BER versus  $E_b/N_0$ , bandwidth efficiency 6 bits/symbol, rate 1/2 repetition code, and rate 1/2 (6000, 12000, 3, 6)-regular LDPC code.

There is effectively only one info bit involved in each equality check. As a result, the feedback information of an equality check can be as strong as needed at the very beginning of iterative processing, if the coding rate is low enough, illustrated by the MI transfer curve of a rate 1/3 repetition code in Figure 8. Note that the MI transfer curve of the PSM demapper is often concave, which creates a severe problem for parity-check-based codes, but this is not a problem for repetition codes, because the curves for repetition codes are also concave for  $R < 1/2$ . Therefore, although a pure repetition code cannot be optimal for PSM, it is definitely a good choice among others.

To locate the performance of repetition-coded PSM, we compare it to several systems with the same bandwidth efficiency in Figure 9. For an information rate of 6 bits per channel symbol, uncoded 64-QAM has a distance of about 9 dB to the Shannon limit, while a rate 1/2 LDPC-coded 4096-QAM system shortens this distance by 3 dB. In comparison, rate 1/2 repetition-coded PSM with  $N = 12$  is in fact just 3 dB away from the Shannon limit, in spite of zero coding gain. Although the codes for both PSM and QAM have not yet been optimized, these results give a rough impression about their relative performance.

## 6. Alternative Demappers

APP demapping is a good choice for system analysis but the computational complexity is very high. Hence, two alternative demappers are brought out and a brief analysis is given.

**6.1. Gaussian Approximation Approach.** The Gaussian approximation (GA) is well known from multiuser systems. This concept has been widely used in multiple-access/multiplexing systems [18, 19]. The main idea is that

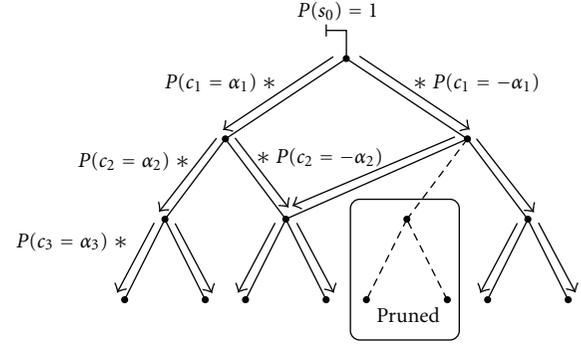


FIGURE 10: Illustration of the update of a priori values and the pruning concept.

the interfering superimposed symbols, together with the Gaussian noise

$$v_n = y - c_n = \sum_{i=1, i \neq n}^N c_i + z, \quad (10)$$

are approximated by a single Gaussian random variable. Hence, the LLR of  $c_n$  can be obtained via

$$\begin{aligned} \text{LLR}(c_n) &\doteq \ln \frac{p(y | c_n = +\alpha_n)}{p(y | c_n = -\alpha_n)} \\ &\approx -\frac{(y - \alpha_n - \mu_{v_n})^2}{\sigma_{v_n}^2} + \frac{(y + \alpha_n - \mu_{v_n})^2}{\sigma_{v_n}^2} \\ &= 4 \cdot \frac{\text{Re}\{\alpha_n^* \cdot (y - \mu_{v_n})\}}{\sigma_{v_n}^2}, \end{aligned} \quad (11)$$

where the mean and variance of the interfering signal are updated by feedback information from the decoder. According to the turbo principle, as the feedback information improves from iteration to iteration, the demapper estimates get more reliable. The complexity increases linearly with the bit load  $N$ , which makes Gaussian approximation very attractive for practical implementations. However, the approximation is not always good enough and can severely limit the performance. Details on this will be given in Section 6.5.

**6.2. Tree-Based Approach.** In [12], SM is modelled by a tree diagram. Since a tree diagram is a special case of a Markov process, the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [20] can be used for SISO demapping of SM to reduce the computational complexity. Let  $s_n$  denote the state at the  $n$ th level. By the property of superposition mapping, the relationship between the states of two neighboring levels is simply given by

$$s_n = s_{n-1} + c_n. \quad (12)$$

The a priori distribution of the states is evaluated recursively according to

$$P(s_n) = \sum_{c_n} P(s_{n-1} = s_n - c_n) P(c_n) \quad (13)$$

through the tree by propagating the values from root to leaves. The a priori information is given via the branches to the lower level and multiplied with the corresponding chip probability, as illustrated in Figure 10. Now, the extrinsic LLR of the  $n$ th chip can be obtained via

$$\begin{aligned} \text{LLR}(c_n) &\doteq \ln \frac{p(y | c_n = +\alpha_n)}{p(y | c_n = -\alpha_n)} \\ &= \ln \frac{\sum_{s_{n-1}} p(y | s_n = s_{n-1} + \alpha_n) P(s_{n-1})}{\sum_{s_{n-1}} p(y | s_n = s_{n-1} - \alpha_n) P(s_{n-1})}. \end{aligned} \quad (14)$$

The likelihood of the received value given the state at the  $n$ th level can be calculated by a backward recursion through the tree. Starting from the leaves,

$$p(y | s_n) = \frac{1}{\pi\sigma_z^2} \exp\left(-\frac{|y - s_n|^2}{\sigma_z^2}\right), \quad (15)$$

one calculates  $p(y | s_n)$  recursively by adding together the likelihood values from the branches that have been multiplied with the corresponding chip probabilities. This approach cleverly utilizes the probability propagation and hence considerably reduces the computational complexity. However, the complexity is still exponentially increasing. Inspired by the tree structure, we would like to introduce a novel method of how to reduce the size of the tree without big sacrifices concerning the performance.

**6.3. Pruned Tree-Based Approach.** The core idea is based on the fact that if at some level of the tree two nodes are close to each other, then during the subsequent tree expansion the corresponding subtrees will always keep the distance relation. (This means that the superposition values for nodes are similar.) In other words, the two subtrees are the same, just one is a shifted version of the other. In the special case that two nodes have exactly the same superposition value, then the subtrees are identical. Such subtrees are redundant and can be pruned. Figure 10 illustrates one possible merging scenario. Considering the demapping process, nothing is changed. All the bit paths still exist, the only difference compared to the full tree is that all pruned node values are shifted by a certain constant. In order to control the pruning rate (and hence the computational complexity), we introduce a merging threshold and denote it by  $\epsilon$ . Now starting from the root, for every level the squared difference values of all nodes are checked if they fulfil

$$\left|s_n^i - s_n^j\right|^2 < \epsilon, \quad (16)$$

where  $s_n^i$  and  $s_n^j$  denote two node values from that level. If two nodes are in the range of  $\epsilon$ , then  $s_n^j$  is merged to  $s_n^i$  and pruned. With a single merging operation actually  $2^{N-n}$  constellation point pairs are merged, that differ by  $s_n^i - s_n^j$ , because a node at level  $n$ , if expanded, would relate to  $2^{N-n}$  leaves. Hence, the tree is reduced by the given sub-tree from node  $s_n^j$  to the leaves, resulting in  $2^{N-n+1} - 1$  pruned nodes. Once the whole tree is processed, all nodes in the range of  $\epsilon$  are merged and the tree will be stored for permanent use.

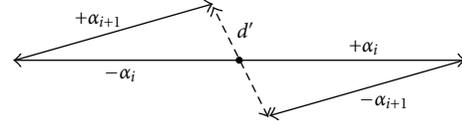


FIGURE 11: Distance between bit sequences “01” and “10”.

TABLE 1: Complexity reduction of the tree with given  $\epsilon$ .

$N$	$\epsilon$	Total nodes	Nodes left	$ \mathcal{X}' $
6	0.1786	127	100	42
8	0.0761	511	258	90
10	0.0391	2047	634	196
12	0.0227	8191	1678	468
14	0.0143	32767	3303	804
16	0.0096	131071	6269	1364

However, there are a few problems. First, with multiple mergers, a node may drift quite far from its original value, maximally by

$$\max(d) < \epsilon \cdot (N - n + 1), \quad (17)$$

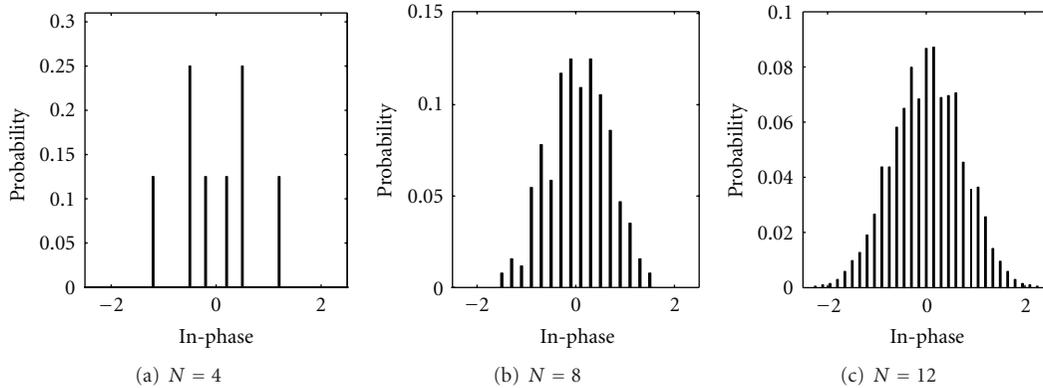
if the remained node always gets merged to another node. The second issue comes from detection. After pruning the tree, some constellation points do not exist in the tree anymore. Hence, given a cut symbol is transmitted, an increased SNR will not lead to a better demapping performance, as such signal point does not exist in the tree anymore. To overcome this problem, we should adjust the transmitter to fit to the receiver. Therefore, the same pruned tree should be used to select the transmit symbol, given the coded bits. One takes  $N$  coded bits and follows the tree from root to leaf, while choosing the branch according to the bit. This way the ambiguity between the mapper and demapper is removed, as both sides are working with the same constellation space.

The parameter  $\epsilon$  gives a certain freedom to balance between complexity and performance. Clearly if one would prune the tree too much, the performance would degrade significantly. From simulations, a certain point was found after that catastrophic merging takes place. This happens if  $\epsilon$  is large enough so that any code bit sequences, “01” and “10”, for any  $N$ , get merged. The distance  $d'$  between these two bit sequences is shown in Figure 11 by the dashed line.

Hence, to avoid such short span mergers,  $\epsilon$  should be chosen smaller than  $d'^2$ . Given the amplitude factor  $\alpha = 1/\sqrt{N}$ , the boundary values are listed in Table 1. The formula to calculate  $d'^2$  for a fixed  $N$  is

$$\begin{aligned} d'^2 &= (2(c_n - c_{n+1}))^2 \\ &= 4 \left( \alpha \cos\left(\frac{\pi(n-1)}{N}\right) - \alpha \cos\left(\frac{\pi n}{N}\right) \right. \\ &\quad \left. + \alpha \sin\left(\frac{\pi(n-1)}{N}\right) - \alpha \sin\left(\frac{\pi n}{N}\right) \right)^2. \end{aligned} \quad (18)$$

The node reduction is significant. For  $N = 16$  already approximately 95% of the tree is pruned and the rate

FIGURE 12: Distribution of  $\text{Re}\{x\}$  for pruned tree,  $\alpha = \sqrt{1/N}$ .

increases with increasing  $N$ . In the last column, we can see the symbol cardinality corresponding to the pruned tree,  $|\mathcal{X}'|$ . For  $N = 16$ , it is about 2% of  $2^{16}$ . Clearly the mapping is no longer bijective. The new constellation diagram and probability distribution are shown in Figures 13 and 12. As stated before, after pruning is finished, all nodes in the range of  $\epsilon$  are merged. Hence, the central region is no longer geometrically densely populated, as seen in Figure 2, but instead probabilistically nonuniform. Correspondingly, there are many bit sequences merged to constellation points in the central region and less near the border.

**6.4. Complexity Comparison.** As mentioned before, the Gaussian approximation demapper has a very low complexity that increases linearly with  $N$ . APP and full tree-based demapper both have exponentially increasing complexity, but the latter already shows a noticeable reduction. For APP, to calculate the LLRs for each chip, the demapping algorithm has to go through  $2^N$  constellation points, meaning for each symbol it visits  $N \cdot 2^N$  points. However, for tree-based demapping, due to the tree structure the complexity to calculate the LLR for each chip varies. For the top level one needs to consider just 2 nodes. Totally, considering the forward and backward recursion and the LLR calculation, a tree-based demapper visits roughly  $5 \cdot 2^N$  points. Hence, already the full tree reduces the complexity compared to APP demapping. As can be seen from Table 1, the pruned tree significantly reduces the whole tree size, leading to a proportionally smaller computational complexity.

**6.5. Performance.** The Gaussian approximation-based demapper has a very low complexity, but it is associated with a severe performance degradation. In Figure 14, the MI transfer characteristics for both pruned tree and GA demapper are shown. Clearly GA does not work so well. In [21] it has been pointed out that for a rate 1/2 repetition code, GA can only support bit loads up to  $N = 4$  and after that presents a very high error floor. The pruned tree approach (Using  $\epsilon$  values from Table 1) shows much smaller degradation in the MI transfer characteristics with respect to GA. The comparison between the BER performance

of pruned tree and GA demapper is given in Figure 15. For  $N > 4$ , the GA demapper makes a severe problem for convergence. Comparing Figure 15 with Figure 6, we see that tree pruning brings some performance degradation. This is easy to understand, since a certain amount of information is lost during pruning. Nevertheless, tree pruning does not make a big problem for convergence.

## 7. PAPR Control and Compensation

The output symbols of a superposition mapper inherently exhibit a large peak-to-average power ratio (PAPR). High peaks occur when the superimposed chips have similar polarities. The PAPR of the output symbols is defined as

$$\text{PAPR} = \frac{\max\{|x|^2\}}{E\{|x|^2\}}, \quad (19)$$

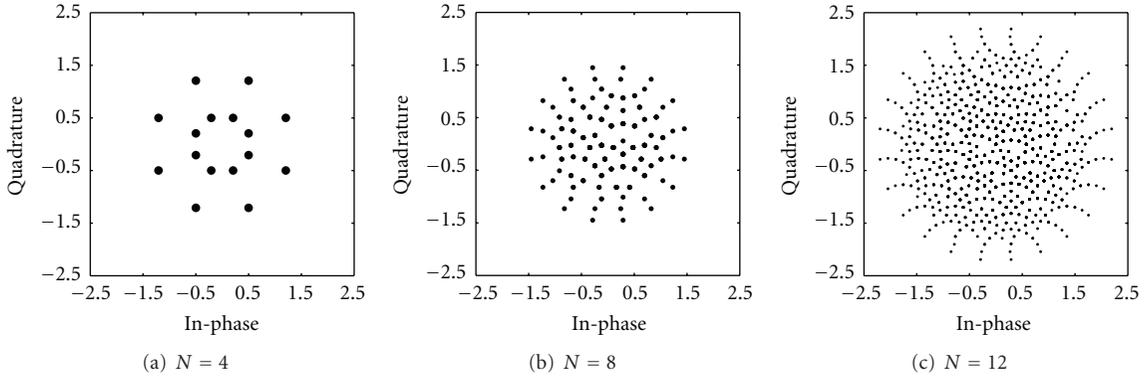
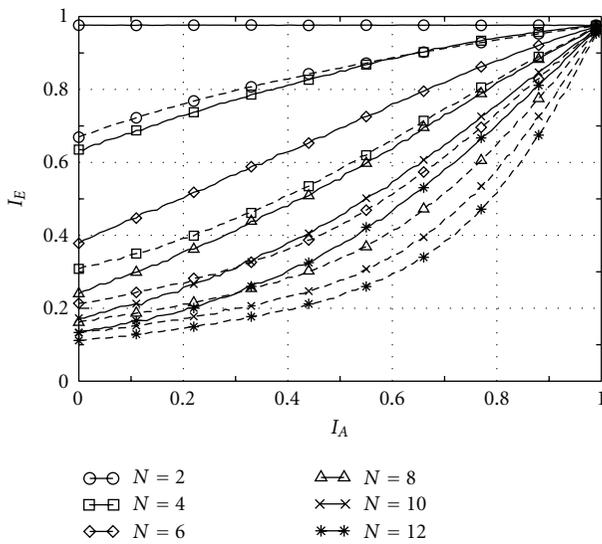
where  $E\{|x|^2\}$  denotes the average symbol power (for simplicity, we use  $P_x$  in the following). For PSM ( $\alpha_n = 1/\sqrt{N}$ ,  $\theta_n = \pi(n-1)/N$ ), the PAPR value can be calculated as

$$\begin{aligned} \text{PAPR} &= \frac{1}{N} \left| \sum_{n=1}^N e^{j\pi(n-1)/N} \right|^2 \\ &\approx (0.4N + 0.1) \quad \text{for } N \geq 4, \end{aligned} \quad (20)$$

where the first equation comes from the fact that the highest peak happens when the component bits are all zeros or all ones. From (20), it is clear that the PAPR grows linearly with the bit load  $N$ . Consequently, the power amplifier has to operate with a large back-off, which reduces the power efficiency.

Clipping at the transmitter side is an efficient and simple method to reduce the PAPR. As shown in Figure 16, if the output symbol has a larger magnitude than the given clipping threshold  $A$ , the clipper (CLP) will limit the magnitude of the symbol while keeping its phase as follows:

$$\bar{x} = \begin{cases} x, & \text{if } |x| \leq A, \\ A \cdot \frac{x}{|x|}, & \text{otherwise.} \end{cases} \quad (21)$$

FIGURE 13: Constellation diagrams for pruned tree,  $\alpha = \sqrt{1/N}$ .FIGURE 14: MI transfer characteristics for pruned tree demapper (solid lines) and for GA demapper (dashed lines),  $E_c/N_0 = 5$  dB.

The clipping process is usually characterized by the clipping ratio (CR)  $\gamma$ , which is defined as

$$\gamma = \frac{A}{\sqrt{P_x}}. \quad (22)$$

Certainly, the power of clipped symbols decreases with  $\gamma$ . As shown in [22], the average output power of the clipper is given by

$$P_{\bar{x}} = (1 - e^{-\gamma^2})P_x. \quad (23)$$

The power loss due to clipping can be compensated by subsequent normalization. To preserve the same average power compared to the unclipped symbol, all symbols must be amplified by a factor  $a \geq 1$ :

$$a = \sqrt{\frac{P_x}{P_{\bar{x}}}} = \sqrt{\frac{1}{1 - e^{-\gamma^2}}}. \quad (24)$$

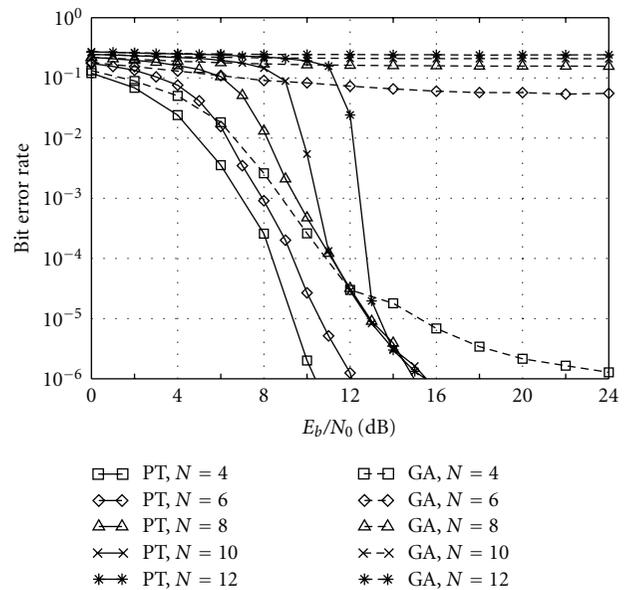


FIGURE 15: BER performance of PSM with pruned tree (PT) and GA demapper, rate 1/2 repetition code.

As a result, more power is allocated to the unclipped symbols after normalization. This process can be treated as a special form of unequal error protection, which makes the unclipped symbols more robust to the channel noise. The PAPR of the clipped symbol remains the same after normalization since the symbols within the burst are multiplied by the same factor  $a$ .

With a subsequent pulse shaping filter, there will be no bandwidth expansion for the signal in continuous-time domain. However, new peaks may grow since filtering usually causes overlapping between the consequent symbols.

The distortion due to clipping can be described by a clipping noise  $z_{cl}$ . The clipped symbol  $\bar{x}$  can be modeled as a summation of the unclipped symbol  $x$  and the clipping noise  $z_{cl}$  as follows:

$$\bar{x} = x + z_{cl}. \quad (25)$$

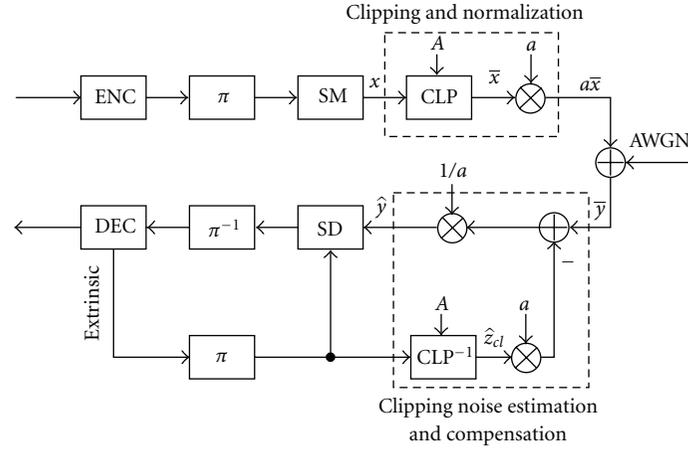


FIGURE 16: Coded PSM system with PAPR control and compensation.

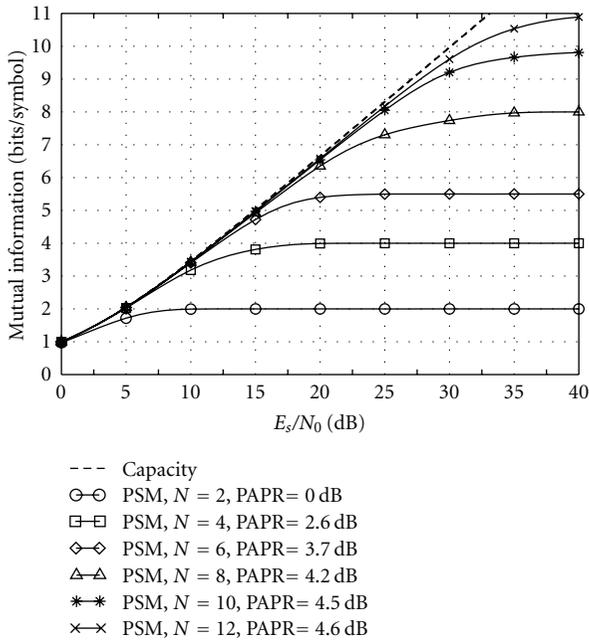


FIGURE 17: Mutual information of peak-limited PSM over the AWGN channel.

After normalization and for the AWGN channel, the received signal  $\bar{y}$  can be written as follows:

$$\begin{aligned}\bar{y} &= a \cdot \bar{x} + z \\ &= a \cdot (x + z_{cl}) + z.\end{aligned}\quad (26)$$

**7.1. Mutual Information of Clipped PSM.** The mutual information of clipped PSM is given in Figure 17, in which the PSM symbols are clipped to have the same PAPR as conventional QAM with the same bit load. Compared to the unclipped PSM in Figure 4(a), the mutual information of clipped PSM does not degrade significantly. Revisiting

Figure 4(b), we see that even with the same PAPR, PSM still outperforms QAM in the slope region.

**7.2. Clipping Noise Estimation and Compensation.** To compensate the distortions introduced by clipping, clipping noise should be estimated and removed from the received signal. Assuming that the clipping threshold  $A$  and the normalization factor  $a$  are known at the receiver side, the clipping noise can be estimated, for example, as follows [23]:

$$\hat{z}_{cl}^{\text{OPT}} = \sum_{x \in \mathcal{X}} P(x) \cdot z_{cl,x}, \quad (27)$$

where  $\mathcal{X}$  denotes the symbol alphabet and  $z_{cl,x}$  is the clipping noise corresponding to symbol  $x$ .

Equation (27) is optimal in the sense of minimizing the mean square error of the estimated clipping noise. Unfortunately, the computational complexity of this algorithm grows exponentially with bit load  $N$ . As an alternative algorithm, soft compensation algorithm is proposed by Tong et al. in [24, 25]. The clipping noise is treated as an equivalent Gaussian noise sample, and a look-up table method is used to speed up the detection. Another alternative is the soft reconstruction algorithm (SRA). The idea is to use decoder feedback to reconstruct the soft transmitted symbol then repeat the clipping process as in the transmitter and derive the clipping noise. In the following, we will briefly introduce the SRA algorithm proposed in [26] and extend it to the case of APP demapping.

Let  $L_{c_n}$  denote the extrinsic LLR of binary chips from the channel decoder, then its corresponding soft binary chip can be calculated as follows:

$$\mu_{c_n} = \tanh\left(\frac{L_{c_n}}{2}\right). \quad (28)$$

Using the superposition mapping rule, the soft symbol  $\mu_x$  can be obtained by superimposing the binary soft chips with known power and phase allocation as follows:

$$\mu_x = \sum_{n=1}^N \alpha_n e^{j\theta_n} \mu_{c_n}. \quad (29)$$

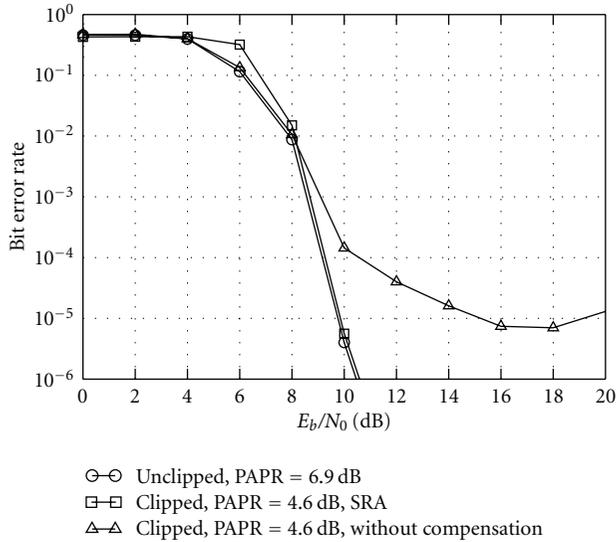


FIGURE 18: BER performance of clipped PSM with rate 1/4 repetition code and bit load  $N = 12$ .

With the knowledge of the reconstructed symbol  $\mu_x$  and the clipping threshold  $A$ , the estimated clipping noise can be obtained as follows:

$$\hat{z}_{cl}^{SRA} = \begin{cases} 0, & \text{if } |\mu_x| \leq A, \\ \mu_x \cdot \left( \frac{A}{|\mu_x|} - 1 \right), & \text{otherwise.} \end{cases} \quad (30)$$

Then the received signal is refined by subtracting the estimated clipping noise as follows:

$$\hat{y} = \frac{\bar{y} - a \cdot \hat{z}_{cl}^{SRA}}{a}. \quad (31)$$

The refined signal can be used in the APP demapper. Compared to the optimal solution in (27), the proposed SRA method has only a linear complexity.

Figure 18 shows the BER performance for the clipped PSM transmission over the AWGN channel. After clipping, the PAPR of the output symbol is reduced from 6.9 dB to 4.6 dB. It can be seen that an error floor exists even at the high SNR region if clipping noise is not compensated. With the SRA algorithm, the SNR loss due to clipping is reduced to 0.15 dB at  $10^{-5}$ . This result demonstrates that the SRA algorithm can efficiently compensate the performance loss due to clipping while introducing only marginal complexity overhead.

## 8. Conclusions and Future Work

In this paper, we have carried out an extensive study on phase-shifted superposition mapping (PSM), particularly on its behavior in the scenario of iterative demapping and decoding. It is shown that PSM is a quasi-bijective mapping scheme and provides no rate limit in a coded system. Due to a geometrically Gaussian-like symbol distribution, PSM

has a good potential to approach the capacity of Gaussian channels. Analogous to conventional mapping schemes like PSK/QAM, PSM can easily be applied in a bit-interleaved coded modulation (BICM) scenario, albeit with different requirements on the channel code. Via an EXIT chart analysis, it is found that for PSM with large bit loads a simple repetition code can already outperform a classical LDPC code. Though surprising, the reason for this phenomenon is indeed simple. Because of superimposing binary chips with identical power, the main task of the channel code is to guarantee a perfect separation of superimposed chips instead of combating the noise. A repetition code simply works better than a parity-check code with respect to this purpose. Numerical results show that rate 1/2 regular-repetition-coded PSM with bit load  $N = 12$  can outperform regular-LDPC-coded QAM with the same bandwidth efficiency and is indeed just 3 dB away from the Shannon limit at a BER of  $10^{-6}$ .

Besides theoretical concerns, several practical issues are also treated in this contribution. Using a tree diagram to represent the constellation evolution process of PSM and pruning those nodes that are close enough to each other, a dramatic complexity reduction can be achieved with an acceptable performance degradation. Furthermore, by baseband clipping at the transmitter side in conjunction with iterative soft compensation at the receiver side, the peak-to-average power ratio of PSM symbols can be controlled to a reasonable level without much sacrificing the BER performance.

There are at least two interesting topics for future work. A repetition code can give an excellent performance for a PSM system, particularly in the initial iterations. Nevertheless, in late-stage iterations parity-check codes can provide some gain over repetition codes. Therefore, finding good hybrid repetition and parity-check codes deserves to be an interesting topic for superposition mapping. Second, the good performance of the pruned-tree-based SISO demapping algorithm provides an important hint. In the PSM constellation, some points are close to each other and some not. This certainly brings some penalty in case of noisy channels. It is worthwhile to check if one can use a tree to refine the PSM constellation to achieve additional performance improvement.

## Acknowledgment

This work has been supported by the German Research Foundation (DFG) under Contracts nos. HO 2226/10-1 and HO 2226/9-2.

## References

- [1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.
- [2] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2006.
- [3] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for Gaussian channels," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 913–929, 1993.

- [4] R. F. H. Fischer, *Precoding and Signal Shaping for Digital Transmission*, John Wiley & Sons, New York, NY, USA, 2002.
- [5] E. Schrödinger, *Statistical Thermodynamics*, Cambridge University Press, Cambridge, UK, 1962.
- [6] G. D. Forney Jr., "Trellis shaping," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 281–300, 1992.
- [7] G. R. Lang and F. M. Longstaff, "A leech lattice modem," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 6, pp. 968–973, 1989.
- [8] P. Fortier, A. Ruiz, and J. M. Cioffi, "Multidimensional signal sets through the shell construction for parallel channels," *IEEE Transactions on Communications*, vol. 40, no. 3, pp. 500–512, 1992.
- [9] D. G. Forney Jr. and G. Ungerboeck, "Modulation and coding for linear Gaussian channels," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2384–2415, 1998.
- [10] L. Duan, B. Rimoldi, and R. Urbanke, "Approaching the AWGN channel capacity without active shaping," in *Proceedings of IEEE International Symposium on Information Theory (ISIT '97)*, p. 374, Ulm, Germany, June 1997.
- [11] H. Schoeneich and P. A. Hoeher, "Adaptive interleave-division multiple access—a potential air interface for 4G bearer services and wireless LANs," in *Proceedings of the International Conference on Wireless and Optical Communications and Networks (WOCN '04)*, Muscat, Oman, June 2004.
- [12] X. Ma and L. Ping, "Coded modulation using superimposed binary codes," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3331–3343, 2004.
- [13] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, 1998.
- [14] T. Wo and P. A. Hoeher, "Superposition mapping with application in bit-interleaved coded modulation," in *Proceedings of the International ITG Conference on Source and Channel Coding (SCC '10)*, Siegen, Germany, January 2010.
- [15] H. Schoeneich, *Adaptiver Interleave-Division Mehrfachzugriff (IDMA) mit Anwendung in der Mobilfunkkommunikation*, Ph.D. dissertation, University of Kiel, Kiel, Germany, 2007.
- [16] S. ten Brink, "Convergence of iterative decoding," *Electronics Letters*, vol. 35, no. 13, pp. 1117–1118, 1999.
- [17] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.
- [18] L. Ping, L. Liu, and W. K. Leung, "A simple approach to near-optimal multiuser detection: Interleave-division multiple access," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '03)*, pp. 391–396, New Orleans, La, USA, March 2003.
- [19] H. Schoeneich and P. A. Hoeher, "A hybrid multiple access scheme approaching single user performance," in *Proceedings of the Workshop on Signal Processing in Communications (SPC '03)*, pp. 163–168, Baiona, Spain, September 2003.
- [20] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.
- [21] M. Noemm, T. Wo, and P. A. Hoeher, "Multilayer APP detection for IDM," *Electronics Letters*, vol. 46, no. 1, pp. 96–97, 2010.
- [22] H. Ochiai and H. Imai, "Performance analysis of deliberately clipped OFDM signals," *IEEE Transactions on Communications*, vol. 50, no. 1, pp. 89–101, 2002.
- [23] S. M. Kay, *Fundamentals of Statistical Signal Processing*, vol. 1 of *Estimation Theory*, Prentice Hall, Upper Saddle River, NJ, USA, 1st edition, 1993.
- [24] J. Tong, L. Ping, and X. Ma, "Superposition coding with peak-power limitation," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, pp. 1718–1723, Istanbul, Turkey, June 2006.
- [25] J. Tong, L. Ping, and X. Ma, "Superposition coded modulation with peak-power limitation," *IEEE Transactions on Information Theory*, vol. 55, no. 6, pp. 2562–2576, 2009.
- [26] D. Hao and P. A. Hoeher, "Iterative estimation and cancellation of clipping noise for multi-layer IDMA systems," in *Proceedings of the International ITG Conference on Source and Channel Coding (SCC '08)*, Ulm, Germany, January 2008.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

