Enumerative sphere shaping techniques for short blocklength wireless communications

Enumerative Sphere Shaping Techniques for Short Blocklength Wireless Communications

Yunus Can Gültekin

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-5178-1

Cover design by Yunus Can Gültekin.
Printed by Gildeprint.

The research presented in this dissertation was conducted in the Information and Communication Theory Lab (ICTLab) of the Signal Processing Systems (SPS) group at the department of Electrical Engineering, Eindhoven University of Technology (TU/e).

This work has been supported by TU/e Impuls program, a strategic cooperation between NXP Semiconductors and TU/e.

Enumerative Sphere Shaping Techniques for Short Blocklength Wireless Communications

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op woensdag 16 december 2020 om 13:30 uur

door

Yunus Can Gültekin

geboren te İzmir, Turkije

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr.ir. A.M.J. Koonen
1e promotor:	prof.dr.ir. Frans M.J. Willems
copromotoren:	dr.ir. Wim J. van Houtum
	Assoc. Prof. Alex Alvarado
leden:	Prof. Gerhard Kramer (Technical University of Munich)
	Prof. Erik Agrell (Chalmers University of Technology)
	prof.dr.ir. Sonia H. de Groot
adviseur:	dr.ir. Alessio Filippi (NXP Semiconductors)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Summary

Nowadays, wireless communication standards are promising data rates around 10 Gbps to their users. Compared to the standards that came out at the beginning of the millennia, this translates to more than two orders of magnitude increase in data rates over two decades. This increase does not only stem from the use of multi-antenna techniques or higher frequencies and the consequent utilization of larger bandwidths. There is also a rapid improvement in the efficient usage of the available spectrum. Advanced forward error correction (FEC) codes that virtually close the coding gap became widely available with low-density parity-check and polar codes leading the way. Furthermore, 1024-point constellations—which were once deemed unrealistic—are now considered as ordinary parts of wireless systems.

Recently, the popularization of high-order modulation formats brought a new dimension to the research on bandwidth-efficient communications: constellation shaping. Constellation shaping identifies techniques to optimize the properties of modulation formats to match the characteristics of the communication channel. By realizing shaping, it is possible to save more than 30% transmit power. Moreover, this room for improvement is particularly significant for constellations larger than 16-point. Consequently, Böcherer *et al.* introduced probabilistic amplitude shaping (PAS) in 2015 as a power-efficient transmission strategy in which both the coding and the shaping gaps are closed.

PAS improves the performance of digital communication systems by including amplitude shaping as an outer code—the inner code being a systematic FEC code—and closing the so-called shaping gap for the additive white Gaussian noise (AWGN) channel. During our research, we have investigated enumerative sphere shaping (ESS) as the amplitude shaping technique in the PAS framework for short blocklengths. ESS specifies a pair of algorithms that create a fixed-to-fixed length mapping from messages to channel inputs with minimum energy. For this purpose, we construct an enumerative amplitude trellis containing the sequences sorted lexicographically. The enumerative trellis needs to be stored in the memory to realize ESS. Moreover, significant computational power is necessary to realize the algorithms.

In this thesis, we first examine sphere shaping and PAS from an information-theoretic perspective. Next, we evaluate the performance of PAS with ESS over the AWGN and fading channels. We then study the practical implementation of ESS in the PAS framework, compare it to alternative sphere shaping algorithms such as two algorithms by (Laroia et al., 1994), and provide techniques to decrease its storage and computational complexity. Finally, we study a partial shaping scheme based on ESS in which we focus on quantized channel input distributions.

First, we provide an information-theoretic framework based on weak typicality to study the achievable rates of PAS. Within this framework, we develop random sign-coding arguments, and we show that PAS achieves the capacity of discrete memoryless channels with symmetric capacity-achieving distributions. We consider both symbol-metric and bit-metric decoding (BMD). Then we investigate the optimality of sphere codes, and we show that they minimize the rate loss at any blocklength. We demonstrate that for short blocklengths for the AWGN channel, sphere shaping whose objective is to construct energy-efficient signal sets outperforms constant composition distribution matching whose motivation is to match the capacity-achieving distribution.

Second, we study the optimum shaping and FEC coding rates for PAS that maximize the performance gain. Inspired by (Wachsmann *et al.*, 1999), we use the gap-to-capacity of shaped BMD as the performance metric. We show that for a given target rate and constellation, there is an optimum shaping and coding redundancy combination that minimizes the gap-to-capacity. Moreover, we repeat this study for fading channels, and we show that as the communication channel becomes more dynamic, i.e., changes first to Rician, then to Rayleigh, increased coding redundancy is required to optimize the performance. We justify these observations by simulating ESS-based PAS. Finally, we introduce an input-selection procedure that enables the use of nonsystematic convolutional codes from the IEEE 802.11 standard in PAS, and we assess the performance of PAS with these codes.

Third, we investigate the practical implementation of ESS. To decrease the storage and computational complexity, we propose a bounded precision (BP) version of ESS. We prove the invertibility of the shaping function for BP ESS, and we show that the resulting rate loss can be upper-bounded as a function of the precision. A consequence of the BP implementation is that it enables sliding-window shaping (SWS). SWS requires only local and 16-bit arithmetic operations, in contrast to the original technique where the entire input sequence must be kept in the processor. Finally, to further decrease the required storage, we propose to store only a single column of the trellis and compute the rest on-the-fly. Our results apply to Laroia's Algorithm 1, while our BP technique also applies to Laroia's Algorithm 2, i.e., shell mapping.

Finally, we study partial shaping, and we show that keeping specific amplitude bits uniform and independent of the others, i.e., using quantized distributions, does not lead to a significant decrease in achievable rates. Then we propose a corresponding amplitude shaping architecture, partial ESS, which reduces the required storage and computational complexity of shaping in return to a negligible loss in performance.

Ayşe (Perihan) Kılıç'ın anısına

Table of Contents

List of Abbreviations

The following abbreviations are used throughout this thesis:

5G	Fifth Generation	JSCC	Joint Source-channel Coding
AIR	Achievable Information Rate	KL	Kullback–Leibler
ASK	Amplitude-shift Keying	LA1	Laroia's Algorithm 1
AWGN	Additive White Gaussian Noise	LDPC	Low-density Parity-check
BICM	Bit-interleaved CM	LM Rate	Lower Bound on the Mismatch Capacity
BMD	Bit-metric Decoding	LSB	Least Significant Bit
BP	Bounded Precision	LUT	Lookup Table
BRGC	Binary Reflected Gray Code	MB	Maxwell-Boltzmann
CC	Convolutional Code	MI	Mutual Information
CCDM	Constant Composition DM	MLC	Multilevel Coding
CM	Coded Modulation	MPDM	Multiset-partition Distribution Matching
CSI	Channel State Information	MSB	Most Significant Bit
D&C	Divide-and-conquer	OFDM	Orthogonal Frequency-division Multiplexing
dB	Decibel	OtF	On-the-fly
DM	Distribution Matching	P-ESS	Partial Enumerative Sphere Shaping
DMC	Discrete Memoryless Channel	PAS	Probabilistic Amplitude Shaping
ESS	Enumerative Sphere Shaping	PHY	Physical Layer
FEC	Forward Error Correction	PS	Probabilistic Shaping
FER	Frame Error Rate	QAM	Quadrature Amplitude Modulation
FSM	Finite State Machine	RQ	Research Question
FP	Full Precision	SM	Shell Mapping
GB	Gigabyte	SMD	Symbol-metric Decoding
GMI	Generalized MI	SNR	Signal-to-noise Ratio
GS	Geometric Shaping	SWS	Sliding Window Shaping
IP	Internet Protocol	TCM	Trellis Coded Modulation

Part I

Introduction

CHAPTER 1

Introduction

"With every mistake, we must surely be learning."
- George Harrison

1

1.1 Motivation

In the context of engineering, most articles and theses written on communications in the past decade start with an emphasis on one of the following: "the demand for increased data rates" or "the ever-increasing number of devices connected to the internet". Consequently, communications literature may seem to be suffering from a constant lack of creativity to an outside observer, at least in the introductory sections. However, although quite appealing, it is not our objective to revolt against cliches in this thesis. All in all, there is a perfectly justifiable reason behind this routine: The data rates and the number of connected devices are indeed increasing at a pace that is hard to keep up with. According to the Cisco Visual Networking Index [1], the average Wi-Fi connection speed will increase from 24.4 megabits per second (Mbps) in 2017 to 54.2 Mbps in 2022 globally. The same study forecasts that the global IP traffic will rise from 122 exabytes[1] per month (EBpm) to 396 EBpm, with more than 25 billion connected devices. To provide a more striking and relatable example, especially to the same outside observer we mentioned above, consider a movie with 4K resolution which typically has a size around 100 gigabytes (GB). Downloading this movie with the 28.8 kbps dial-up internet connections of the 1990s would take a few weeks shy of a year. However, using Wi-Fi 6 with speeds up to 10 Gbps, the same movie can be downloaded in a couple of minutes.[2] These changes are so drastic that it is understandable to mention them whenever possible. In the end, cliches are cliches for a reason, aren't they?

Behind these advances in communication engineering, there lies a century's worth of research and development on coded modulation (CM), accelerated by the seminal works of Claude E. Shannon [2]. From a purely Shannon-theoretic perspective, there are two fundamental ways of increasing the data rate of a communication system in bits per unit time. On the one hand, for a fixed spectral efficiency, one can increase the bandwidth, and hence, the data rate. As an example, a Wi-Fi-based wireless link and a 5G new radio-based wireless link usually operate at similar spectral efficiencies. However, the former frequently utilizes 40 MHz bands while the latter can employ up to 400 MHz [3], leading to an order of magnitude difference in data rate. With this motivation, wireless communications researchers have been exploring the possibility of communication using terahertz carrier frequencies, where there is a vast amount of bandwidth available [4]. On the other hand, for a fixed bandwidth, one can increase the spectral efficiency, and hence, the data rate. This elegant approach aims to utilize the available bandwidth more efficiently, rather than seeking for a trivial solution such as using more bandwidth. In this line of research, the goal is basically to operate as close as possible to *channel capacity* [2].

The first ingredient necessary to communicate at rates close to capacity is channel coding. A careful examination of Shannon's second theorem (the noisy-channel coding theorem) shows that rather than prescribing ways to identify capacity-achieving codes, he only proved the existence of codes that enable reliable information transmission. The following decades

[1] One exabyte is 10^{18} bytes, where a byte is 8 bits.

[2] Wi-Fi 6 is the marketing name of the IEEE 802.11ax standard.

saw an outstanding effort to develop such forward error correction (FEC) codes and optimal decoding algorithms. As an example, convolutional codes and the Viterbi algorithm were developed around the 1960s [5], and today, they are included in every device which can connect to a Wi-Fi access point. Capacity-approaching turbo codes were introduced in the 1990s [6] and standardized for cellular and satellite communications. Soon after, Gallager's low-density parity-check (LDPC) codes were rediscovered [7, 8], and became an essential part of every new generation communication protocol. Finally, in 2009, polar codes were introduced by Arıkan, proven to be capacity-achieving, and quickly put to use in the 5G standard [9].

The second ingredient required to operate close to capacity is constellation shaping, which broadly identifies techniques used to optimize the properties of the channel inputs. A second inspection of the channel coding theorem shows that the channel inputs must be Gaussian-distributed to achieve the capacity of the additive white Gaussian noise (AWGN) channel, which is used to model the actual physical channel in many practical scenarios. Equivalently, the capacity of the AWGN channel can be achieved if the channel inputs are confined in a sphere when represented as points in the Euclidean space. In fact, this equivalence is the reason why people fell into two camps in their perception of shaping, and it is the focus of the main philosophical discussion in this thesis. On the one hand, the goal of shaping is formulated as to "match" the capacity-achieving distribution which is called the direct method by Calderbank and Ozarow in [10]. On the other, supporters of the indirect method (including the author of this thesis) advocated the construction of the most energy-efficient channel input set. Starting from the early 1980s, both direct and indirect shaping approaches have been investigated, and shaping even found application in commercial systems such as the V.34 voiceband modem recommendation [11]. However, it was not until recently that an efficient and flexible way to combine constellation shaping with channel coding is introduced.[3]

Sketched in 2014 [12], probabilistic amplitude shaping (PAS) is devised as a capacity-achieving CM strategy that incorporates shaping with coding in 2015 [13]. As shown in Fig. 1.1, the key idea behind PAS is to realize shaping prior to coding, which is sometimes called the "reverse concatenation" in the context of constrained coding [14, 15]. In this structure, shaping is used to select the amplitudes of the channel inputs and to optimize the signaling for the channel in the presence of input constraints. Then systematic channel coding is employed to select the signs, and to enable FEC. This strategy also enables transmission rate adaptation in the shaping stage, by tuning the parameters of the shaping block, which would otherwise require the implementation of many different channel codes and modulation formats, e.g., as in [16, Table 21-24]. PAS quickly attracted attention, especially in the optical communication society, and it is demonstrated both numerically and experimentally that PAS provides rate adaptivity, reach increase, and improved power-efficiency for optical links [17–19]. However, optical communication was not the only area in which PAS is recognized as a key technique, and PAS is considered for inclusion in the future cellular and digital subscriber line standards as well [20, 21]. This thesis is a result of our four-years-long study

[3] Unlike the case for channel coding, we leave the historical background of shaping for Chapter 2.

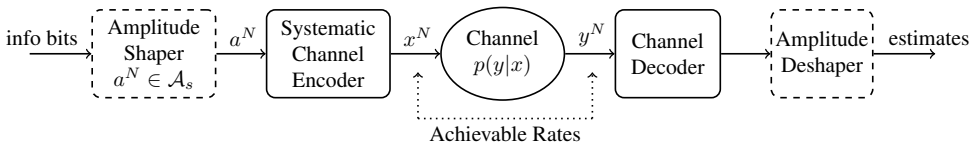

Figure 1.1: Reverse concatenation architecture. Amplitude shaping and deshaping blocks (dashed boxes) are examined in this thesis.

centered around PAS, focusing mainly on its amplitude shaping stage.

1.2 Thesis Scope

In Chapter 2 which constitutes Part I of this thesis with the current chapter, we provide some background information on communication channels, coded modulation strategies, and achievable rates. Then in Part II, we play the part of an information-theorist and analyze the fundamental limits of PAS and the amplitude shaping stage for both asymptotically large and finite blocklength regimes. The research questions (RQs) that we try to answer in this part and the outline of the corresponding chapters are as follows.

RQ-1 What are the achievable information rates (AIRs) of PAS for symbol-metric decoding (SMD) and bit-metric decoding (BMD)? Is it possible to achieve the capacity of memoryless channels with PAS? What are the optimum shaping and coding rates in PAS that maximize AIR gains?

In Chapter 3, we revisit the problem of computing AIRs of PAS, and show that for memoryless channels with symmetric capacity-achieving distributions, PAS achieves the capacity. Unlike the AIR computations in [22–24] which follow Gallager's error exponent approach [25, Ch. 5], we base our derivations on weak typicality [26, Secs. 3.1, 7.6, 15.2] in a modified manner. Our main contribution in this chapter is to provide the *random sign-coding* framework that unifies achievability results for all PAS settings. Moreover, in our achievability proofs, the codes are generated as constructively as possible unlike the random codes in most proofs of Shannon's channel coding theorem. Finally, we use a modified version of the gap-to-capacity curves proposed by Wachsmann *et al.* in [27] to find the optimum shaping and coding rates for PAS. We demonstrate that as the channel becomes more and more frequency selective, the coding rate should be decreased to obtain the optimum performance, while the shaping rate increases. Considering Fig. 1.1, we take $N \to \infty$ in this chapter.

RQ-2 What is the "best" amplitude shaping strategy for finite values of the blocklength N? What are the metrics to be used to assess the "goodness" of different amplitude shaping approaches?

In Chapter 4, we first introduce performance metrics and the notion of "optimality" for amplitude shaping codes $\mathcal{A}_s$. We investigate two different approaches. The first is constant composition distribution matching (CCDM) [28] which represents the direct method in the Calderbank/Ozarow terminology [10]. The second is sphere shaping [29, 30] which represents the indirect side. We show that both these approaches are asymptotically optimum for large N. Then we demonstrate the superiority of sphere shaping over CCDM for finite N in terms of rate loss and shaping gain.

The main takeaways from Part II are that (1) PAS achieves capacity, and (2) sphere shaping is the best amplitude shaping technique for finite (especially for small) N.

In Part III of this thesis, we look at the problem at hand through the eyes of a communication engineer, and we investigate sphere shaping algorithms and their practical implementation within the PAS framework. The RQs that we try to answer in this part and the outline of the corresponding chapters are as follows.

RQ-3 How can sphere shaping be realized algorithmically? Which algorithm provides high performance with low complexity? What is the end-to-end decoding performance of PAS using sphere shaping over the AWGN and frequency selective channels?

In Chapter 5, we introduce enumerative sphere shaping (ESS) as an effective way to realize sphere shaping. We compare ESS with two different sphere shaping algorithms proposed in [30] by Laroia, Farvardin, and Tretter, and we show that ESS provides virtually the same performance with smaller computational and storage requirements. Then we provide end-to-end decoding results based on Monte Carlo simulation of PAS with ESS and LDPC codes, and we demonstrate that more than 1 dB improvement in power-efficiency can be obtained with ESS for both the AWGN and frequency selective channels, over a large range of transmission rates and shaping blocklengths.

RQ-4 Can PAS be incorporated into existing communication systems that are based on the IEEE 802.11 standard? Can PAS be combined with the nonsystematic convolutional codes used in 802.11 [16] which are a mandatory part of the standard?

In Chapter 6, we provide a guideline on how to use the nonsystematic convolutional codes from the IEEE 802.11 in the PAS framework. An intermediate layer called the *input-select* layer is proposed to be placed in between the shaping and coding layers of PAS to ensure that the temporal structure of amplitude

sequences is preserved through nonsystematic encoding. This way, PAS can be fully integrated into 802.11-based systems, in both convolutional- and LDPC-coded modes.

RQ-5 Can we further improve the energy-efficiency of ESS for practical scenarios? How can ESS be implemented with low storage complexity, minimal computational requirements, and limited latency?

In Chapter 7, we first devise an algorithm to compute the exact amplitude distribution of the operational shaping set of ESS. We use this algorithm to demonstrate that the energy efficiency of ESS is virtually the same as the algorithms from [30] for moderate to long blocklengths. For short blocklengths where ESS is slightly inefficient, we propose a heuristic routine to optimize ESS. Then we introduce bounded precision sphere shaping implementation that decreases the required storage, arithmetic precision, and computational power for ESS and the two other sphere shaping algorithms from [30]. Next, we propose a sliding window shaping method for ESS which works with limited (and fixed) arithmetic precision and operates with smaller latency than the classical implementation. Finally, on-the-fly computation techniques are employed to realize ESS with further decreased storage requirements.

RQ-6 How much do we need to shape the channel input to reap most of the possible shaping gain? Is it possible to obtain a reduction in required storage and computational complexity by realizing a "rough" shaping strategy? How can this rough shaping be realized based on ESS?

In Chapter 8, we evaluate the loss in AIR and shaping gain resulting from transmitting channel inputs with quantized Gaussian distributions. We demonstrate that shaping one or two most significant amplitude bits of the binary labels of the channel inputs (while keeping the rest uniform) is enough to obtain most of the possible gain. Then we introduce partial ESS (P-ESS) in which ESS is used to shape some amplitude bits while the remaining amplitude bits are reserved for data bits. Simulation results are then provided to show that P-ESS provides virtually the same performance as ESS over the AWGN channel.

The main takeaways from Part III are that (1) ESS is an effective sphere shaping algorithm, and (2) implementation of ESS can be tailored to specific constraints imposed by the available hardware resources such as limited memory and computational power, and finite arithmetic precision.

Finally, Chapter 9 concludes with a summary and a discussion of our results.

1

CHAPTER 2

Preliminaries

2.1 Notation Convention

We use calligraphic letters $\mathcal{X}$ to denote sets of real numbers $\{\mathcal{X}(1), \mathcal{X}(2), \cdots, \mathcal{X}(|\mathcal{X}|)\}$. The n-fold Cartesian product of $\mathcal{X}$ with itself is denoted by $\mathcal{X}^n$ while $\mathcal{X} \times \mathcal{Y}$ is the Cartesian product of $\mathcal{X}$ and $\mathcal{Y}$. We define $\mathcal{X}\mathcal{Y}$ as $\{xy : x \in \mathcal{X}, y \in \mathcal{Y}\}$.

Capital letters X are used to denote random variables while lower case letters x are used to denote their realizations. We use X^N and x^N to denote random vectors $(X_1, X_2, \cdots, X_N)$ and their realizations $(x_1, x_2, \cdots, x_N)$, respectively. Concatenation of two vectors x^N and y^N is indicated by (x^N, y^N). Element-wise multiplication of x^N and y^N is denoted by $x^N \otimes y^N$. We use bold letters $\mathbf{T}$ to specify matrices where $\mathbf{t}_i$ denotes the i^{th} column of $\mathbf{T}$, i.e., $\mathbf{T} = [\mathbf{t}_0 \mathbf{t}_1 \cdots \mathbf{t}_N]$.

For $x \in \mathcal{X}$ and a discrete set $\mathcal{X}$, the probability of occurrence is expressed as $\Pr\{X = x\}$. The probability (mass or density) function of a (discrete or continuous) random variable X is denoted by $p(x)$. The joint probability function of X and Y is denoted by $p(x, y)$. The conditional probability of Y given X is denoted by $p(y|x)$. The entropy of a discrete random variable X is denoted by $H(X)$ (in bits). The (differential) entropy of a continuous random variable X is denoted by $h(X)$ (in bits). The expected value of X is shown as $E[X]$.

We use $\mathbb{1}[\cdot]$ to indicate the indicator function which is 1 when its argument is true and 0 otherwise. The operator $[\cdot]^+$ is defined as $\max\{0, \cdot\}$. The superscript "$\bullet$" indicates signaling with sphere shaping, while the superscript "$\blacksquare$" indicates uniform signaling.

2.2 Channel Models

2.2.1 The AWGN Channel

The time-discrete AWGN channel shown in Fig. 2.1 (left) is modeled as $Y_n = X_n + Z_n$ where $X_n \in \mathcal{X}$ and $Y_n \in \mathcal{Y}$ are the channel input and output at time $n = 1, 2, \cdots, N$, respectively [26, Ch. 9]. Here N is the blocklength, and both $\mathcal{X}$ and $\mathcal{Y}$ are sets of real numbers. The noise variables Z_n's are independently and identically distributed (i.i.d.) according to a zero-mean Gaussian distribution with variance σ^2, and they are assumed to be independent of the input X. We assume that there is an average power constraint such that any channel input sequence $(x_1, x_2, \cdots, x_N)$ satisfies

$$\frac{1}{N} \sum_{n=1}^{N} x_n^2 \leq P. \tag{2.1}$$

For the AWGN channel, the signal-to-noise ratio (SNR) is defined as $\text{SNR} = \mathbb{E}\left[X^2\right] / \sigma^2$.

2.2.2 Fading Channels

The time-discrete communication channel in the existence of fading is modeled as $Y_n = H_n X_n + Z_n$ as shown in Fig. 2.1 (right). Here, H_n is the fading coefficient at time n. In

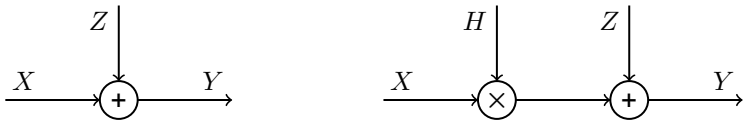

Figure 2.1: Channel Models: (Left) the AWGN channel and (right) fading channels.

the literature, the fading coefficients H_n are frequently assumed to be i.i.d. according to a Nakagami distribution with fading parameter $1/2 \leq \eta < \infty$ [31, Sec. 3.2.2], and they are assumed to be independent of X. The channel in Fig. 2.1 (right) reduces to a Rayleigh fading channel for $\eta = 1$, to a Rician fading channel with parameter K for $\eta = (K+1)^2/(2K+1)$, and to the AWGN channel for $\eta = \infty$. The Rayleigh and Rician fading channel models are used in our paper [32].

2.3 Channel Capacity

For a given communication channel, the *channel capacity* C is an upper bound on the amount of information that can be transmitted reliably per unit of channel use. This reliability is in the sense that the probability of making an error can be made arbitrarily small, as the number of channel uses grows to infinity [2]. Shannon's second theorem (the noisy-channel coding theorem) states that at any rate below C, it is possible to transmit information reliably, while at any rate above C, the probability of making an error is bounded away from zero [2].[1]

2.3.1 Discrete Memoryless Channels

Consider a memoryless channel for which the channel law is given by

$$p(y^N|x^N) = \prod_{n=1}^{N} p(y_n|x_n) \tag{2.2}$$

with discrete input $X \in \mathcal{X}$ and discrete output $Y \in \mathcal{Y}$. The capacity of this discrete memoryless channel (DMC) is defined as

$$C = \max_{p(x)} I(X;Y) \tag{2.3}$$

in bits per real symbol (bit/1-D). Here $I(X;Y)$ is the mutual information (MI) of the channel input X and output Y. The maximization in (2.3) is over all possible input distributions $p(x)$.

[1] On the other hand, Shannon's first theorem (noiseless coding theorem) states that N i.i.d random variables with entropy $H(X)$ cannot be compressed into less than $NH(X)$ bits reliably, while the converse statement also holds [2]. This reliability is in the sense that the risk of information loss can be made arbitrarily small as $N \to \infty$ [33, Sec. 4.4].

Shannon proved the achievability of C for DMCs in [2]—in fact, he only provided an outline of the proof—using the *asymptotic equipartition property* and the concept of *typicality* [26, Ch. 3]. We used similar arguments in our paper [34].

2.3.2 The AWGN Channel with Power Constraint

The capacity of the AWGN channel in Fig. 2.1 (left) is [26, Sec. 9.1]

$$C = \max_{p(x):\mathbb{E}[X^2]\leq P} I(X;Y) = \frac{1}{2}\log_2\left(1+\frac{P}{\sigma^2}\right) \tag{2.4}$$

in bit/1-D, and it is shown versus SNR $= P/\sigma^2$ in Fig. 2.2. The maximum in (2.4) is obtained when X_n's are i.i.d. according to the zero-mean Gaussian distribution with variance P [2]. The corresponding random coding argument shows that input sequences, drawn from a Gaussian distribution, are likely to lie in an N-sphere of squared radius $N(P+\varepsilon)$ for any $\varepsilon > 0$, when $N \to \infty$. Therefore, it is reasonable to choose the input sequences inside a sphere, or equivalently, to use an N-sphere as the signal space boundary, to achieve capacity. Alternatively, the sphere hardening result that is discussed, e.g., by Wozencraft and Jacobs in [35, Sec. 5.5, Fig. 5.20], shows that practically all codewords are near the surface of the sphere as $N \to \infty$. Consequently, one could argue that codewords chosen from the surface of a sphere would lead to good signal sets. We compared both approaches in our papers [32,36]. The first approach we refer to as sphere shaping, and it is our main focus in this thesis. The second approach we refer to as constant composition shaping.

2.3.3 Capacity at Finite Blocklengths

Channel capacity is an upper bound on the maximum AIR for a given SNR that holds for asymptotically large signaling blocklengths N. On the other hand, to limit complexity and latency, practical communication systems operate at finite N, which is often small, where transmitting at a rate close to capacity reliably is not a realistic objective. In [37, eq. (1)], Polyanskiy *et al.* provided the normal approximation $R_{\max}$ to the maximal AIR for the AWGN channel in the finite blocklength regime. More specifically,

$$R_{\max}(\varepsilon, N) \approx C - \sqrt{\frac{V}{N}}Q^{-1}(\varepsilon) \tag{2.5}$$

where ε is the error probability, Q^{-1} is the inverse Q-function, and V is the AWGN channel dispersion defined as [37, eq. (293)]

$$V = \frac{(\log_2 e)^2}{2}\left(1-\frac{1}{(1+\mathrm{SNR})^2}\right). \tag{2.6}$$

We used (2.5) as a benchmark in our paper [32]. In Fig. 2.2, we also show $R_{\max}(10^{-3}, N)$ for $N = 100, 250, 1000$ versus SNR. We see from the inset figure that signaling at rate 3 bit/1-D with $N = 100$ introduces a gap to capacity of approximately 2 dB.

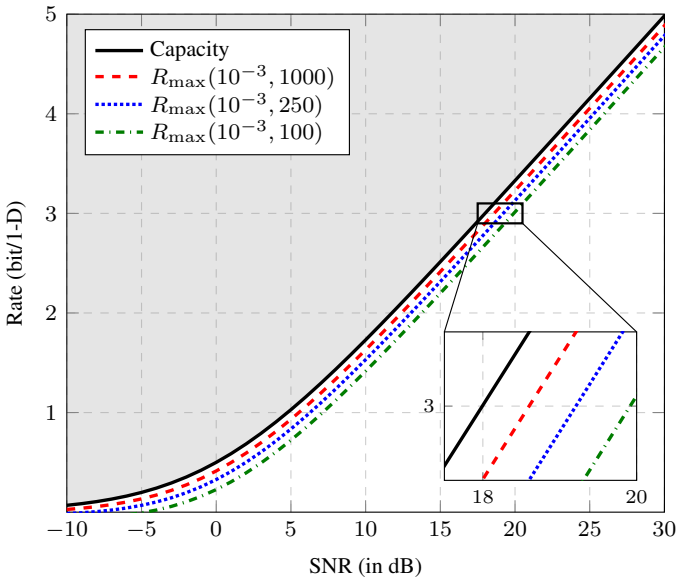

Figure 2.2: (Solid black) The capacity of the AWGN channel, and (dashed red, dotted blue, dash-dotted green) Polyanskiy's normal approximation to the maximum achievable rate for the AWGN channel in the finite blocklength regime for $N = 1000, 250, 100$, respectively.

2.4 Coded Modulation

After Shannon set the fundamental limits of communication, the research focused on designing communication systems that operate close to the channel capacity. The joint or independent design of channel encoders/decoders and modulators/demodulators is called *coded modulation* (CM). In the following, we introduce concepts related to CM, and we briefly discuss the principles of bit-interleaved CM (BICM).

2.4.1 Signal Sets and Binary Labeling

An elementary constraint in practical communication system design is that the channel input alphabet $\mathcal{X}$ must be discrete. One of the most frequently used signal sets is the 2^m-amplitude-shift keying (ASK) alphabet which is in the form

$$\mathcal{X} = \{\pm 1, \pm 3, \cdots, \pm(2^m - 1)\} \tag{2.7}$$

where $2^m = M$ is the constellation size, and the integer m is the number of bits required to represent each input $x \in \mathcal{X}$. The alphabet in (2.7) can be factorized as $\mathcal{X} = \mathcal{S}\mathcal{A}$ where $\mathcal{S} = \{-1, 1\}$ and $\mathcal{A} = \{1, 3, \cdots, 2^m - 1\}$ are the sign and amplitude alphabets, respectively.

Table 2.1: The BRGC for 8-ASK

A	7	5	3	1	1	3	5	7
S	-1	-1	-1	-1	1	1	1	1
X	-7	-5	-3	-1	1	3	5	7
C_1	0	0	0	0	1	1	1	1
C_2	0	0	1	1	1	1	0	0
C_3	0	1	1	0	0	1	1	0

A binary labeling strategy assigns a unique m-tuple $C_1C_2\cdots C_m$ to every possible channel input symbol $x \in \mathcal{X}$. One of the most frequently used labeling strategies is the binary reflected Gray code (BRGC) [38, Definition 2.10] in which labels of adjacent symbols differ by only one bit. When a BRGC is used, the binary label $C_1C_2\cdots C_m$ of a 2^m-ASK constellation point can be decomposed into the sign bit C_1 and the amplitude bits $C_2C_3\cdots C_m$. In our papers and this thesis, we only focus on ASK alphabets labeled with the BRGC, and we use X and $C_1C_2\cdots C_m$ interchangeably. The BRGC of order $m = 3$ is tabulated in Table 2.1 along with the amplitudes and the signs of each channel input. Color codes relate the amplitude bits to the amplitudes with red, and the sign bit to the sign with blue.

2.4.2 Bit-interleaved Coded Modulation

CM combines multi-level modulation with FEC, and it is indispensable for digital communication strategies targeting high transmission rates. To realize CM, different techniques have been proposed in the literature, such as multilevel coding (MLC) [27, 39], trellis CM (TCM) [40], and BICM [38, 41–44]. Among the many proposed CM architectures, the de-facto standard is to combine a high-order modulation format with a binary FEC code using a binary labeling strategy (typically a BRGC), frequently in the absence of an interleaver, and to use bit-metric decoding (BMD) at the receiver, which corresponds to the BICM paradigm.[2] Throughout this thesis, we will restrict our attention to BICM systems.

In Fig. 2.3, the block diagram of a BICM transceiver is provided. At the transmitter, a k-bit information sequence $u^k = (u_1, u_2, \cdots, u_k)$ is encoded by a rate $R_\mathrm{c} = k/(mN)$ binary FEC code. Afterward, the coded sequence c^{mN} is divided into m-bit vectors, each of which is mapped to a channel input symbol x via the symbol mapper. Finally, the sequence x^N is transmitted over the channel. The transmission rate of this construction is $R_\mathrm{t} = k/N$ bit/1-D.

At the receiver, BMD is employed. First, log-likelihood ratios (LLR) are computed for each bit independently. The LLR of the bit-level j of channel input x is

$$L_j = \log \frac{\sum_{x\in\mathcal{X}_{j,0}} p(x)p(y|x)}{\sum_{x\in\mathcal{X}_{j,1}} p(x)p(y|x)}, \tag{2.8}$$

[2] The term "BMD" refers to the demapping/decoding strategies where the bit-levels C_i's are treated independently at the receiver. BMD will be discussed in more detail in Section 2.7.

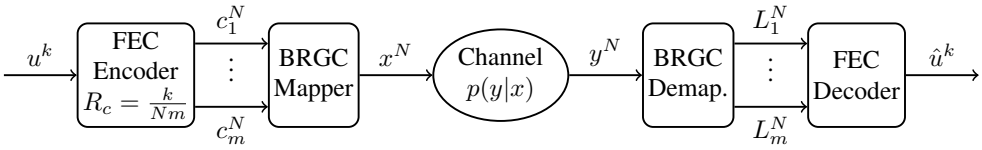

Figure 2.3: BICM system under consideration with transmission rate $R_t = k/N$ bit/1-D.

for $j = 1, 2, \cdots, m$, where y is the corresponding channel output. Here $\mathcal{X}_{j,u}$ denotes the subset of $\mathcal{X}$ which have $C_j = u$ in their binary labels for $u \in \{0, 1\}$. Then based on the LLRs, an FEC decoder recovers the information bits and outputs the estimates $\hat{u}^k$. In this thesis, we use the system in Fig. 2.3 as our baseline for performance comparison.

2.5 Shaping Gap for the AWGN Channel

In most communication systems and standards, each possible channel input symbol is transmitted with equal probability. In Fig. 2.4, $I(X;Y)$ is shown for the AWGN channel where X is distributed uniformly over 2^m-ASK alphabets. It is visible that as SNR $\to \infty$, the MI is bounded away from the capacity, and it converges to m.

When the input X has a continuous uniform distribution, we obtain the maximum AIR for uniform signaling. This so-called *uniform capacity* is computed for an input which is uniformly distributed over the interval $(\pm\sqrt{3P}, \pm\sqrt{3P})$ [45, Sec. 4.2.6], and it is shown in Fig. 2.4. We observe that there is a gap to capacity that results from using uniform inputs instead of Gaussian ones. This gap, i.e., the increase in required SNR to achieve a given rate, is called the *ultimate shaping gap*, and it is asymptotically equal to 1.53 dB for large SNRs. Equivalently, the ultimate shaping gap can be expressed as a decrease in the AIR for asymptotically large SNRs, and it is equal to 0.255 bit/1-D. In the following, we will derive this ultimate shaping gap using information-theoretic and geometric arguments.

2.5.1 Information-theoretic Perspective

Continuous Gaussian Distribution

The capacity-achieving input distribution for the AWGN channel is a zero-mean Gaussian distribution. Without loss of generality, we assume that $E[X^2] = 1$, and hence, SNR $= 1/\sigma^2$. When the capacity-achieving distribution is used, the channel output Y is the summation of two independent zero-mean Gaussian random variables, and thus, it is a zero-mean Gaussian random variable with variance $E[Y^2] = 1 + \sigma^2$. Consequently, $I(X;Y)$ can be written for a

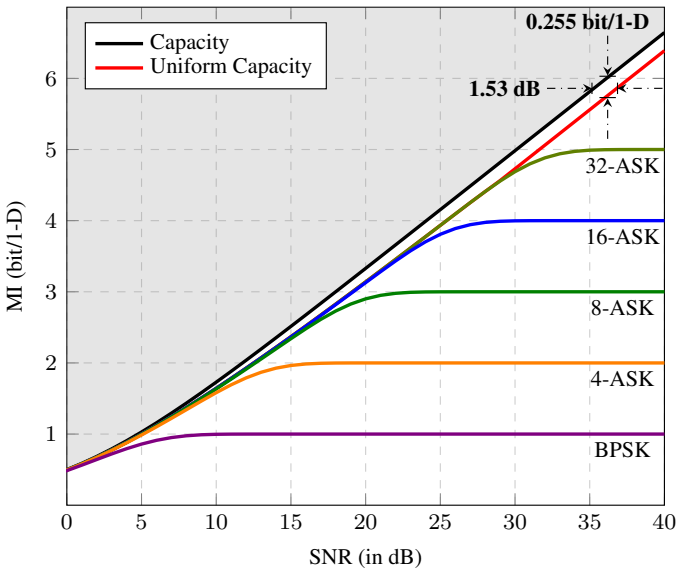

Figure 2.4: The capacity of the AWGN channel (black), the capacity of the AWGN channel for uniform X (red), and $I(X;Y)$ for uniform X where $\mathcal{X}$ is given by (2.7) for $1 \leq m \leq 5$.

Gaussian X as

$$\begin{aligned} C = I_{\mathrm{g}}(X;Y) &= h(Y_{\mathrm{g}}) - h(Y_{\mathrm{g}}|X_{\mathrm{g}}) \\ &= h(Y_{\mathrm{g}}) - h(N) \qquad (2.9) \\ &= \frac{1}{2}\log_2 2\pi e(1+\sigma^2) - \frac{1}{2}\log_2 2\pi e\sigma^2 \\ &= \frac{1}{2}\log_2 \frac{1+\sigma^2}{\sigma^2} \\ &= \frac{1}{2}\log_2(1+\mathrm{SNR}) \qquad (2.10) \end{aligned}$$

where (2.9) follows from $Y = X + N$, and from the independence of X and N. The expression in (2.10) is the famous AWGN channel capacity presented already in (2.4).

Continuous Uniform Distribution

When the channel input X has the uniform distribution

$$p(x) = \begin{cases} \frac{1}{\sqrt{12}} & \text{if } |x| \leq \sqrt{3}, \\ 0 & \text{otherwise,} \end{cases} \qquad (2.11)$$

with variance $E[X^2] = 1$, its differential entropy is $h(X) = \log_2 \sqrt{12}$. Since X is independent of N, the distribution of the channel output $Y = X + N$ can be found by convolving the distributions of X and N. For asymptotically large SNRs, i.e., as $\sigma^2 \to 0$, $p(n)$ converges to the Dirac delta $\delta(n)$, and therefore, $p(y)$ converges to $p(x)$. Here we assumed that the convolution of $f(t)$ and $\delta(t)$ is equal to $f(t)$.

The Ultimate Shaping Gap

Finally, the difference between the MIs of the Gaussian- and uniform-input cases is

$$\begin{aligned} I_{\mathrm{g}}(X;Y) - I_{\mathrm{u}}(X;Y) &= h(Y_{\mathrm{g}}) - h(Y_{\mathrm{u}}) \\ &= \frac{1}{2}\log_2 2\pi e(1+\sigma^2) - h(X_{\mathrm{u}} + N) \\ &\overset{\sigma^2 \to 0}{\to} \frac{1}{2}\log_2 2\pi e - \lim_{\sigma^2 \to 0} h(X_{\mathrm{u}} + N) \\ &= \frac{1}{2}\log_2 2\pi e - h(X_{\mathrm{u}}) \\ &= \frac{1}{2}\log_2 2\pi e - \log_2 \sqrt{12} \\ &= \frac{1}{2}\log \frac{\pi e}{6} = 0.2546 \text{ bits}, \end{aligned} \tag{2.12}$$

which is the ultimate shaping gap as shown in Fig. 2.4. When $\sigma^2 \to 0$, the AWGN channel capacity is given by $C = (\log_2 \mathrm{SNR})/2$. If $2C_1 = \log_2 \mathrm{SNR}_1$, $2C_2 = \log_2 \mathrm{SNR}_2$, and $C_2 - C_1 = 0.2546$ from (2.12), then

$$10 \log_{10} \frac{\mathrm{SNR}_2}{\mathrm{SNR}_1} = 10 \log_{10} 2^{2 \cdot 0.2546} = 1.5329 \text{ dB}, \tag{2.13}$$

which tells that the (vertical) ultimate shaping gap of 0.2546 bits in rate is equivalent to a (horizontal) gap of 1.5329 dB in SNR as shown in Fig. 2.4.

2.5.2 Geometric Perspective

When the channel input X has a uniform distribution, channel input sequences form an N-cube in the Euclidean space. On the other hand, the most-energy efficient geometry for a fixed volume is the N-sphere. We note here that the *volume* of the signal space in the continuous domain is analogous to the *cardinality* of the signal set in the discrete domain, and they both determine the rate.

N-spherical Signal Spaces

The volume of the N-sphere of radius R is $V^{\bullet} = B_N R^N$ where for even N,

$$B_N = \frac{\pi^{N/2}}{(N/2)!}. \tag{2.14}$$

The volume of an N-D shell of outer radius ρ with infinitesimal thickness $\Delta\rho$ is

$$B_N\rho^N - B_N(\rho-\Delta\rho)^N \approx B_N N\rho^{N-1}\Delta\rho. \tag{2.15}$$

Then assuming that the points inside the N-sphere of radius R are uniformly distributed, i.e., $p(x^N) = 1/V^\bullet$, the average energy is

$$\begin{aligned} E^\bullet &= \frac{1}{V^\bullet}\int_0^R B_N N\rho^{N-1}\rho^2 \mathrm{d}\rho \\ &= \frac{1}{V^\bullet}\int_0^R B_N N\rho^{N+1}\mathrm{d}\rho \\ &= \frac{1}{V^\bullet}B_N\frac{N}{N+2}R^{N+2} \\ &= \frac{N}{N+2}R^2. \end{aligned} \tag{2.16}$$

N-cubical Signal Spaces

The volume of the N-cube of side length d is $V^\blacksquare = d^N$. Assuming that the points inside the N-cube of side length d are uniformly distributed, i.e., $p(x^N) = 1/d^N$, the average energy is $E^\blacksquare = Nd^2/12$.

The Ultimate Shaping Gap

When the N-cube and the N-sphere have the same volume, i.e., the same rate,

$$B_N R^N = d^N \Longrightarrow \frac{d^2}{R^2} = (B_N)^{2/N}. \tag{2.17}$$

Then the ratio of the average energy of the N-cube of side length d to that of the N-sphere of radius R is

$$\frac{E^\blacksquare}{E^\bullet} = \frac{Nd^2}{12}\frac{N+2}{NR^2} \overset{(2.17)}{=} \frac{N+2}{12}(B_N)^{2/N}. \tag{2.18}$$

As $N \to \infty$,

$$B_N = \frac{\pi^{N/2}}{(N/2)!} \approx \frac{\pi^{N/2}}{\sqrt{\pi N}\exp(-N/2)(N/2)^{N/2}} \tag{2.19}$$

which follows from Stirling's approximation. Therefore, as $N \to \infty$,

$$(B_N)^{2/N} \approx \frac{\pi}{\exp(-1)N/2} = \frac{2\pi e}{N}. \tag{2.20}$$

Finally,

$$\lim_{N\to\infty} 10\log_{10}\frac{E^{\blacksquare}}{E^{\bullet}} = \lim_{N\to\infty} 10\log_{10}\left(\frac{N+2}{12}(B_N)^{2/N}\right) \tag{2.21}$$

$$= 10\log_{10}\frac{\pi e}{6} \tag{2.22}$$

$$= 1.5329 \text{ dB}. \tag{2.23}$$

Interestingly, the ultimate shaping gap of 1.53 dB can be derived both from the ratio of the second moment of an N-cube to that of an N-sphere and from entropy calculations with 1-D distributions. Though we have to note that this fact was found remarkable by Forney *et al.* already 40 years ago [46, Sec. IV-B].

2.6 Constellation Shaping

2.6.1 Shaping with Discrete Signal Sets

Since the capacity-achieving distribution for the AWGN channel is symmetric around the origin, i.e., it can be factorized as $p(x) = p(s)p(a)$ where $p(s)$ is uniform, we restrict our attention to the distribution of amplitudes, and we assume that the signs are uniform.

There is no analytical expression for the distribution that maximizes the AIR for an ASK constellation and a σ^2. For such constellations, Maxwell-Boltzmann (MB) distributions of the form

$$p(a) = K(\lambda)\, e^{-\lambda a^2},\ a \in \mathcal{A}, \tag{2.24}$$

are pragmatically chosen for amplitude shaping in [13, 47], since they maximize the entropy for a fixed average energy, or equivalently, minimize the average energy for a given entropy [26, Ch. 12]. Furthermore, to achieve a given target rate, the gap between the required SNRs for the capacity-achieving distribution and the optimum MB distribution is insignificant for ASK constellations [23, Table 5.1]. In (2.24), the parameter λ governs the variance of the distribution, and the parameter

$$K(\lambda) = \frac{1}{\sum_{a\in\mathcal{A}} e^{-\lambda a^2}} \tag{2.25}$$

normalizes it.

In Fig. 2.5, the maximum $I(X;Y)$ is shown for 2^m-ASK alphabets for the AWGN channel. The maximization is done over all possible channel input distributions $p(x) = p(s)p(a)$ where S is uniform, and A is MB-distributed. We see that by transmitting channel inputs with MB-distributed amplitudes, it is possible to close the shaping gap for ASK constellations over a wide range of transmission rates virtually completely. For instance, consider the inset figure which shows the region around the target rate 3 bit/1-D for 16 ASK. Here, a shaping gain of 1.08 dB can be obtained by transmitting 16-ASK symbols with MB-distributed amplitudes.

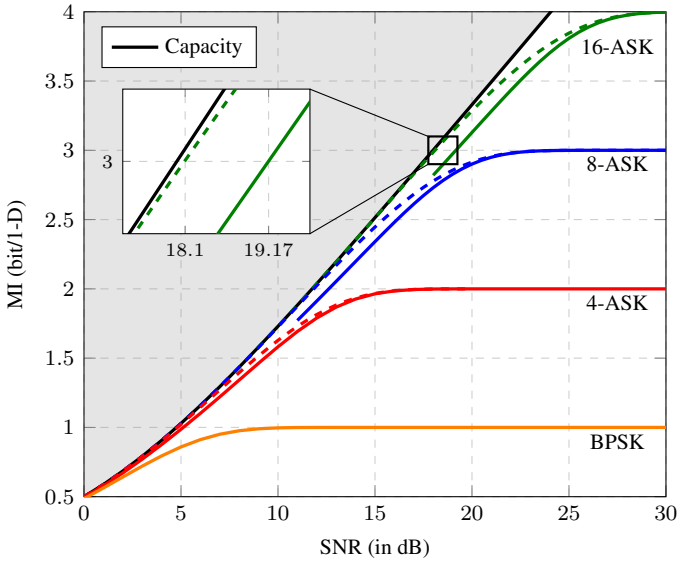

Figure 2.5: (Solid black) The capacity of the AWGN channel, (solid others) $I(X;Y)$ for uniform X, and (dashed) maximum $I(X;Y)$ for MB-distributed X where $\mathcal{X}$ is given by (2.7) for $1 \leq m \leq 4$. For BPSK, $|\mathcal{A}| = 1$, and thus, it is not possible to realize shaping. For 4-ASK, $|\mathcal{A}| = 2$, and thus, the capacity-achieving distribution can be written as an MB distribution. Therefore, the dashed red curve shows the maximum AIR for 4-ASK.

The remaining 0.1 dB gap to capacity is largely due to the discrete nature of $\mathcal{A}$, and negligibly due to the suboptimality of the MB distribution. Furthermore, for all rates below 3 bit/1-D, the shaping gap is virtually closed. For the rates above, it is reasonable to switch to 32-ASK.

In a similar (and pragmatic) manner, sphere shaping is considered also for multidimensional ASK constellations to obtain high energy efficiency, and to obtain (discrete) Gaussian-like-distributed inputs in [29, 30]. In our paper [36], we showed that if the region of an ASK lattice that is bounded by an N-sphere is considered, an MB distribution is induced on the 1-D constituent constellation asymptotically for large N. We emphasize that although MB-distributed and sphere-shaped ASK constellations maximize the energy efficiency, they do not maximize the AIR.

2.6.2 A Brief History of Constellation Shaping Techniques

There exist numerous techniques in the literature, most of them proposed in the late 1980s and early 1990s, that attempt to close the shaping gap. Motivated by the fact that the capacity-achieving distribution for the AWGN channel is Gaussian, these techniques fundamentally

aim at one of the following. The first goal is to construct a signal constellation with a Gaussian-like geometry, which is called geometric shaping (GS) [48–55]. The other approach is to induce a Gaussian-like distribution over the equidistant signal structure, which is called probabilistic shaping (PS) [10, 29, 30, 47, 56]. PS techniques can be further classified into two subgroups using the terminology introduced by Calderbank and Ozarow in [10]. The *direct* approach is to start with a target distribution (which is typically close to the capacity-achieving distribution in an information-theoretic sense) over a low-dimensional signal constellation, and have an algorithm try to obtain it [10, 47]. Following recent literature [57], the direct approach can also be called distribution matching (DM). The *indirect* approach is to start with a target rate and bound the multi-dimensional signal structure by a sphere, which we call sphere shaping [29, 30]. Here, a Gaussian distribution is induced indirectly (when $N \to \infty$) as a by-product. Finally, there exist some *hybrid shaping* approaches in which GS and PS are combined [58–60].

In the context of BICM, signal shaping techniques again attracted a considerable amount of attention in the 2000s. GS was investigated for BICM in [61–63], and PS was studied in [64–67]. An iterative demapping and decoding architecture with PS was proposed in [68]. The achievability of the so-called generalized MI (GMI) was shown for independent but shaped bit-levels in [69]. In [70], it was demonstrated that the GMI is a nonconvex function of the input bit distribution, i.e., the problem of computing the input distribution that maximizes GMI is nonconvex. An efficient numerical algorithm to compute optimal input distributions for BICM was introduced by [71]. The effect of mismatched shaping, i.e., not using the true symbol probabilities or reference constellation at the receiver, was examined in [72]. The achievable rates, error exponents, and error probability of BICM with PS were analyzed in [73]. Signal shaping was investigated for BICM at low SNR in [74]. PS in BICM was considered for Rayleigh fading channels in [75, 76].

For a detailed discussion on GS, we refer the reader to [45, Sec. 4.5]. In [13, Sec. II], a concise review of PS is provided. In this thesis, we restrict our attention to PS.

2.6.3 Probabilistic Amplitude Shaping

Recently in [13], probabilistic amplitude shaping (PAS) has been proposed to provide low-complexity integration of shaping into existing BICM systems. PAS uses a reverse concatenation strategy where the shaping operation precedes FEC coding as shown in Fig. 2.6. This construction has been first examined for constrained coding problems in [14]. A corresponding soft-decision decoding approach for this structure was studied in [15]. PAS can be considered as an instance of the Bliss architecture [14] in which a shaping code is used in the outer layer, and then parity symbols are added in the inner layer. The main advantage of this structure is that amplitude shaping can be added to existing BICM systems as an outer code. In addition to closing the shaping gap, PAS moves the rate adaptation functionality to the shaping layer. This means that instead of using many FEC codes of different rates to obtain a granular set of transmission rates as in [77, Table 5b], the rate can be adjusted by

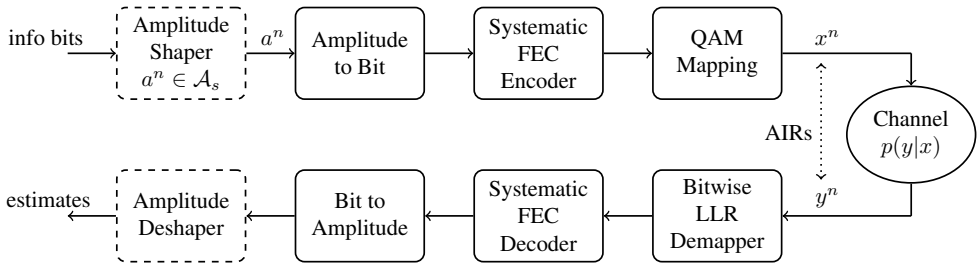

Figure 2.6: Reverse concatenation architecture. Amplitude shaping and deshaping blocks (dashed boxes) are examined in this thesis.

the amplitude shaper with a fixed FEC code. Owing to these advantages, PAS has attracted a lot of attention. PAS has been combined with LDPC codes [13], polar codes [78], and convolutional codes [32]. Its performance has been evaluated over the AWGN channel [13], optical channels [18, 79, 80], wireless channels [32], and parallel channels with channel state information (CSI) available at the transmitter [81].

In summary, PAS combines shaped amplitudes with signs generated by an FEC code in the form of parity based on the binary labels of these amplitudes. Due to the uniform check bit assumption explained in [13, Sec. IV-A.2], the signs have an (approximately) uniform distribution which is required to obtain the capacity-achieving distribution. In the following, the basic and modified PAS structures will be explained. In our papers [32, 36, 82, 83], we used these PAS structures.

2.6.3.1 Basic PAS Structure

Figure 2.7 shows the basic PAS architecture where first, an amplitude shaping block maps a k-bit information sequence u^k to an N-amplitude sequence $a^N = (a_1, a_2, \cdots, a_N)$ in an invertible manner, where $a_j \in \mathcal{A}$ for $j = 1, 2, \cdots, N$. After this mapping block, the amplitudes are transformed into bits using the $m-1$ amplitude bits of the corresponding BRGC. We note that due to the shaped nature of a^N, the bits at the output of the amplitude-to-bit conversion in Fig. 2.7 are nonuniform. These $N(m-1)$ nonuniform bits $c_2^N, c_3^N, \cdots, c_m^N$ are then used as the input of a systematic, rate $R_c = (m-1)/m$ FEC code which is specified by an $N(m-1)$-by-Nm parity-check matrix $\mathbf{P}$. The N-bit parity output of this code is employed as the sign bit-level, i.e., the first bit of the binary labels, to determine the sign sequence $s^N = (s_1, s_2, \cdots, s_N)$. Finally, $x^N = s^N \otimes a^N \in \mathcal{S}^N \mathcal{A}^N$ is transmitted over the channel. The transmission rate of this scheme is $R_t = k/N$ bit/1-D.

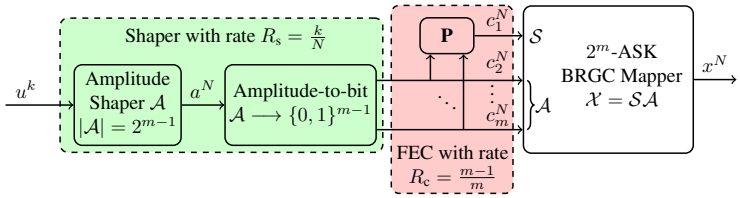

Figure 2.7: Basic PAS structure. All information is carried in the amplitudes of the channel inputs. Transmission rate is $R_t = k/N$ bit/1-D.

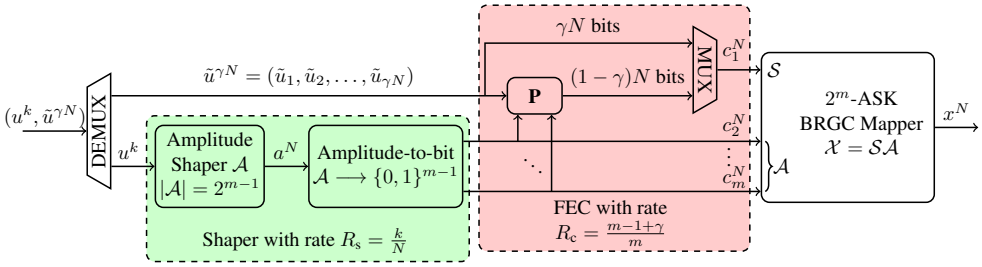

Figure 2.8: Modified PAS structure. Information is carried in the amplitudes of the channel inputs and in some signs. Transmission rate is $R_t = k/N + \gamma$ bit/1-D.

2.6.3.2 Modified PAS Structure

To use a higher FEC code rate $R_c > (m-1)/m$, a modified PAS architecture is proposed in [13] as shown in Fig. 2.8. The code rate in this scheme is $R_c = (m-1+\gamma)/m$ where $\gamma = R_c m - (m-1)$ specifies the rate of extra data (in bit/1-D) that will be transmitted. In this modified structure, in addition to the $N(m-1)$ bit output of the shaper, extra γN information bits $\tilde{u}^{\gamma N}$ are fed to the FEC code which is now specified by an $N(m-1+\gamma)$-by-Nm parity-check matrix $\mathbf{P}$. The $(1-\gamma)N$ bit parity output of the FEC code is then multiplexed with the information bits $\tilde{u}^{\gamma N}$ to form an N-bit sequence s^N that will select the signs. The transmission rate of this scheme is $R_t = k/N + \gamma$ bit/1-D.

Example 2.1 (Shaping, FEC code and transmission rates in PAS). Consider the PAS architecture using 8-ASK, a rate $R_c = 5/6$ FEC code, and a target transmission rate $R_t = 2.25$ bit/1-D. The rate of the additional information is $\gamma = R_c m - (m-1) = 0.5$ bit/1-D. Therefore, the rate of the amplitude shaper should be $k/N = R_t - \gamma = 1.75$ bit/1-D.

2.6.3.3 PAS Receiver

At the receiver, LLRs are computed using (2.8). We emphasize that unlike uniform signaling where $p(x)$ is uniform and can be removed from (2.8) without affecting the performance,

$p(x)$ is nonuniform for PAS and must be used when computing (2.8). Then based on the LLRs, a binary FEC decoder recovers the bits that were encoded by the FEC code. For the basic PAS architecture shown in Fig. 2.7, the output of the decoder consists of the estimates of the amplitude bits. Then these are mapped back to the information bit estimates using the inverse functions of the blocks in the shaper (green boxes in Fig. 2.7 and 2.8), i.e., the corresponding bit-to-amplitude mapper followed by the corresponding amplitude deshaper. In addition to this, for the generalized PAS architecture shown in Fig. 2.8, the decoder also outputs the estimates of the γN extra data bits which were used as some of the signs.

2.7 Achievable Information Rates

For a memoryless channel which is characterized by an input alphabet $\mathcal{X}$, input distribution $p(x)$, and channel law $p(y|x)$, the maximum AIR is the MI $I(X;Y)$ of the channel input X and output Y. Consequently, the capacity of this channel is defined as $I(X;Y)$ maximized over all possible input distributions $p(x)$, typically under an average power constraint, e.g., in [26, Sec. 9.1]. The MI can be achieved, e.g., with MLC and multi-stage decoding [27, 39].

In BICM systems, channel inputs are uniquely labeled with $\log_2|\mathcal{X}| = m$-bit binary strings. At the transmitter, the output of a binary FEC code is mapped to channel inputs using this labeling strategy. At the receiver, BMD is employed, i.e., binary labels C_i's are assumed to be independent, and consequently, the symbol-wise decoding metric is written as the product of bit-metrics

$$\mathfrak{q}(x,y) = \prod_{i=1}^{m} \mathfrak{q}_i\left(c_i\left(x\right), y\right) \tag{2.26}$$

where $c_i(x)$ is the value at the i^{th} position of the binary label of x. Since the metric in (2.26) is in general not proportional to $p(y|x)$, i.e., there is a mismatch between the actual channel law and the one assumed at the receiver, this setup is called *mismatched decoding*.

Different AIRs have been derived for this so-called mismatched decoding setup. One of these is the GMI [84, 85]

$$\text{GMI}\left(p(x)\right) = \max_{s\geq 0} E\left[\log \frac{\left[\mathfrak{q}(X,Y)\right]^s}{\sum_{x\in\mathcal{X}} p(x)\left[\mathfrak{q}(x,Y)\right]^s}\right], \tag{2.27}$$

which reduces to [38, Th. 4.11, Corollary 4.12], [44]

$$\text{GMI}\left(p(c_1)p(c_2)\cdots p(c_m)\right) = \sum_{i=1}^{m} I(C_i;Y) \tag{2.28}$$

when the bit-levels are independent at the transmitter, i.e., $p(x) = p(c_1, c_2, \cdots, c_m) = p(c_1)p(c_2)\cdots p(c_m)$, and

$$\mathfrak{q}_i(c_i, y) = p(y|c_i). \tag{2.29}$$

The rate (2.28) is achievable for both uniform and shaped bit-levels [42, 69]. The problem of computing the bit-level distributions that maximize the GMI in (2.28) is shown to be nonconvex in [70]. The parameter that maximizes (2.27) to obtain (2.28) is $s = 1$.

Another AIR for mismatched decoding is the LM (lower bound on the mismatch capacity) rate [73, 85]

$$\mathrm{LM}\left(p(x)\right) = \max_{s\geq 0, r(\cdot)} E\left[\log \frac{\left[\mathrm{q}(X,Y)\right]^s r\left(X\right)}{\sum_{x\in\mathcal{X}} p(x)\left[\mathrm{q}(x,Y)\right]^s r\left(x\right)}\right] \tag{2.30}$$

where $r(\cdot)$ is a real-valued cost function defined on $\mathcal{X}$. The expectations in (2.27) and (2.30) are taken with respect to $p(x, y)$.

When there is dependence among bit-levels, i.e., when $p(x) \neq p(c_1)p(c_2)\cdots p(c_m)$, the rate [12, 86]

$$R_{\mathrm{BMD}}\left(p(x)\right) = H\left(C_1 C_2 \cdots C_m\right) - \sum_{i=1}^{m} H(C_i|Y) \tag{2.31}$$

is achievable by BMD for any joint input distribution $p(x) = p(c_1, c_2, \cdots, c_m)$. In [12, 86], the achievability of (2.31) is derived using random coding arguments based on strong typicality [87, Ch. 1]. Later in [88, Lemma 1], it is shown that (2.31) is an instance of the so-called LM rate (2.30) for $s = 1$, the symbol decoding metric (2.26), bit decoding metrics (2.29), and the cost function

$$r(c_1, c_2, \cdots, c_m) = \frac{\prod_{i=1}^{m} p(c_i)}{p(c_1, c_2, \cdots, c_m)}. \tag{2.32}$$

We note here that R_{BMD} in (2.31) can be negative as discussed in [88, Sec. II-B]. In such cases, R_{BMD} should not be considered as an achievable rate. To avoid this, R_{BMD} is defined as the maximum of (2.31) and zero in [88, eq. (1)].

2

Part II

An Information-theoretic Study of Amplitude Shaping

2

CHAPTER 3

Achievable Rates of PAS: Random Sign-coding Arguments

"Almost all events are almost equally surprising."
- T. Cover and J. Thomas

Parts of this chapter are published in:

Y. C. Gültekin, A. Alvarado, and F. M. J. Willems, "Achievable information rates of probabilistic amplitude shaping: An alternative approach via random sign-coding arguments," *in Proc. Int. Zurich Seminar on Inf. and Commun. (IZS)*, Zurich, Switzerland, Feb. 2020. (Abstract & poster presentation)

Y. C. Gültekin, A. Alvarado, and F. M. J. Willems, "Achievable information rates for probabilistic amplitude shaping: An alternative approach via random sign-coding arguments," *Entropy*, vol. 22, no. 7: 762, July 2020.

Notational Caveat: Unlike the rest of this dissertation, in this chapter, underlined capital and lower case letters $\underline{X}$ and $\underline{x}$ are used to denote random vectors $(X_1, X_2, \cdots, X_n)$ and their realizations $(x_1, x_2, \cdots, x_N)$, respectively. We note that $n = 1, 2, \cdots, N$ is the time index. Boldface capital and lower case letters $\boldsymbol{B}$ and $\boldsymbol{b}$ are used to denote collections of random variables $(B_1, B_2, \cdots, B_m)$ and their realizations $(b_1, b_2, \cdots, b_m)$, respectively. Underlined boldface capital and lower case letters $\underline{\boldsymbol{B}}$ and $\underline{\boldsymbol{b}}$ are used to denote collections of random vectors and their realizations, respectively,

$$\underline{\boldsymbol{B}} = \begin{pmatrix} \underline{B}_1 \\ \underline{B}_2 \\ \vdots \\ \underline{B}_M \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1N} \\ B_{21} & B_{22} & \cdots & B_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{M1} & B_{M2} & \cdots & B_{MN} \end{pmatrix} \tag{3.1}$$

$$\underline{\boldsymbol{b}} = \begin{pmatrix} \underline{b}_1 \\ \underline{b}_2 \\ \vdots \\ \underline{b}_M \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ b_{21} & b_{22} & \cdots & b_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ b_{M1} & b_{M2} & \cdots & b_{MN} \end{pmatrix}. \tag{3.2}$$

3.1 Introduction

In this chapter, we will address the research question **RQ-1** which concerns the achievable rates of the PAS framework. AIRs of PAS have been investigated in [22–24, 89] based on Gallager's error exponent approach [25, Ch. 5].

- In [22], a random code ensemble is considered from which the channel inputs ($\underline{x}$) are drawn. This is a special case of uniform random coding where for each message, there are multiple codewords. Among these, the codeword which is in the (strongly) typical set[1] with respect to a target distribution $p(x)$ is transmitted [22, Sec. 3]. Then the AIR in [22, eq. (34)] is derived for a general memoryless decoding metric $\mathrm{q}(x, y)$. It is shown that by properly selecting $\mathrm{q}(x, y)$, $I(X;Y)$ and R_{BMD} can be recovered from the derived AIR, and consequently, they can be achieved with PAS.
- In [23, Ch. 10], a random code ensemble is considered from which only the *signs* ($\underline{s}$) of the channel inputs are drawn, while their amplitudes ($\underline{a}$) are generated constructively. We call this the *random sign-coding* setup. The error exponent [23, eq. (10.42)] is then derived again for a general memoryless decoding metric.
- In [24, 89], error exponents of PAS have been examined based on the joint source-channel coding (JSCC) setup, and random sign-coding is considered, but only with SMD and only for the specific case where $\gamma = 0$.

[1] We refer the reader to [87, Ch. 1] for a detailed explanation of strong typicality.

Here, we derive AIRs of PAS in a random sign-coding framework based on weak typicality.[2] We first consider basic sign coding in which amplitudes of the channel inputs are generated constructively while the signs are drawn from a randomly generated code. Basic sign coding corresponds to PAS with $\gamma = 0$, i.e., Fig. 2.7. Then we consider modified sign coding in which only some of the signs are drawn from the random code while the remaining are chosen directly by information bits. Modified sign coding corresponds to PAS with $0 < \gamma < 1$, i.e., Fig. 2.8. We compute AIRs for both SMD and BMD.

Our first objective is to provide alternative proofs of achievability in which the codes are generated as constructively as possible. In our random sign-coding experiment, both the amplitude sequences ($\underline{a}$) and the sign sequence parts ($\underline{s}_{\mathrm{i}}$) which are information bits are constructively produced, and only the remaining signs ($\underline{s}_{\mathrm{p}}$) are randomly generated as illustrated in Fig. 3.1 (which is a simplified version of Fig. 2.8). In most proofs of Shannon's channel coding theorem, channel input sequences ($\underline{x}$) are drawn at random, and the existence of a good code is demonstrated. Therefore, these proofs are not constructive and cannot be used to identify good codes as discussed, e.g., in [91, Sec. I] and references therein. On the other hand, in our proofs using random sign-coding arguments, it is self-evident how—at least a part of—the code should be constructed. Our second objective is to provide a unified framework in which all possible PAS scenarios are considered, i.e., SMD or BMD at the receiver with $0 \leq \gamma < 1$, and corresponding AIRs are determined using a single technique, i.e., the random sign-coding argument.

Note that our approach differs from the random sign-coding setup considered in [23] and [24] where *all* the signs ($\underline{s}_{\mathrm{i}}$ and $\underline{s}_{\mathrm{p}}$) are generated randomly which is called *partially systematic encoding* in [23, Ch. 10]. We will show later that only $\underline{s}_{\mathrm{p}}$ needs to be chosen randomly. Furthermore, we define a special type of typicality ($\mathcal{B}$-typicality, see Definition 3.1 below) that allows us to avoid the mismatched JSCC approach of [24, 89].

3.2 Weak Typicality

Let $\varepsilon > 0$ and n be a positive integer. Consider X with probability distribution $p(x)$. Then the (weak) typical set $\mathcal{A}_\varepsilon^n(X)$ of length-n sequences with respect to $p(x)$ is defined as

$$\mathcal{A}_\varepsilon^n(X) \triangleq \left\{ \underline{x} \in \mathcal{X}^n : \left| -\frac{1}{n} \log p(\underline{x}) - H(X) \right| \leq \varepsilon \right\} \tag{3.3}$$

where

$$p(\underline{x}) \triangleq \prod_{i=1}^{n} p(x_i). \tag{3.4}$$

The cardinality of the typical set $\mathcal{A}_\varepsilon^n(X)$ satisfies [26, Th. 3.1.2]

$$(1-\varepsilon)2^{n(H(X)-\varepsilon)} \overset{\text{(a)}}{\leq} |\mathcal{A}_\varepsilon^n(X)| \overset{\text{(b)}}{\leq} 2^{n(H(X)+\varepsilon)} \tag{3.5}$$

[2] We refer the reader to [26, Sec. 3.1, 7.6 and 15.2] and [90, Ch. 20] for a detailed discussion on weak typicality.

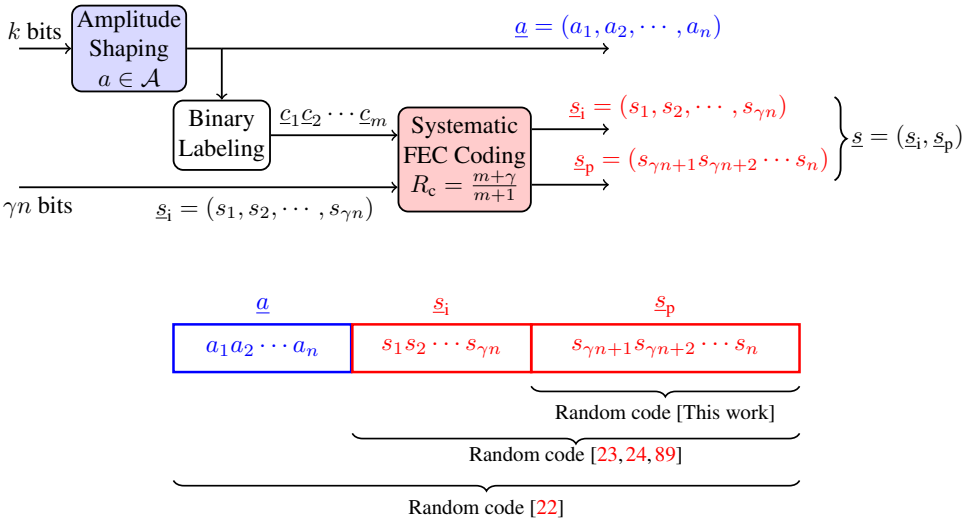

Figure 3.1: (Top) A simplified block diagram of PAS. (Bottom) The scope of the random coding experiments considered in this work and in [22–24, 89].

where (a) holds for n sufficiently large, (b) holds for all n. For $\underline{x} \in \mathcal{A}_\varepsilon^n(X)$, the probability of occurrence can be bounded as [26, eq. (3.6)]

$$2^{-n(H(X)+\varepsilon)} \leq p(\underline{x}) \leq 2^{-n(H(X)-\varepsilon)}. \tag{3.6}$$

The idea of typical sets can be generalized for pairs of n-sequences. Now consider the pair of random variables (X, Y) with probability distribution $p(x, y)$. Then the typical set $\mathcal{A}_\varepsilon^n(XY)$ of pairs of length-n sequences with respect to $p(x, y)$ is defined as

$$\begin{aligned}\mathcal{A}_\varepsilon^n(XY) \triangleq \Bigg\{ (\underline{x}, \underline{y}) \in \mathcal{X}^n \times \mathcal{Y}^n : &\left| -\frac{1}{n} \log p(\underline{x}) - H(X) \right| \leq \varepsilon, \\ &\left| -\frac{1}{n} \log p(\underline{y}) - H(Y) \right| \leq \varepsilon, \\ &\left| -\frac{1}{n} \log p(\underline{x}, \underline{y}) - H(X,Y) \right| \leq \varepsilon \Bigg\}\end{aligned} \tag{3.7}$$

where

$$p(\underline{x}, \underline{y}) \triangleq \prod_{i=1}^{n} p(x_i, y_i), \tag{3.8}$$

and where $p(x)$ and $p(y)$ are the marginal distributions that correspond to $p(x, y)$. The cardinality of the typical set $\mathcal{A}_\varepsilon^n(XY)$ satisfies [26, Th. 7.6.1]

$$|\mathcal{A}_\varepsilon^n(XY)| \leq 2^{n(H(X,Y)+\varepsilon)} \tag{3.9}$$

for all n. For $(\underline{x}, \underline{y}) \in \mathcal{A}_\varepsilon^n(XY)$, the probability of occurrence can be bounded in a similar manner to (3.6) as

$$2^{-n(H(X,Y)+\varepsilon)} \leq p(\underline{x}, \underline{y}) \leq 2^{-n(H(X,Y)-\varepsilon)}. \tag{3.10}$$

Along the same lines, joint typicality can be extended for collections of n-sequences $(\underline{X}_1, \underline{X}_2, \cdots, \underline{X}_m)$, and the corresponding typical set $\mathcal{A}_\varepsilon^n(X_1 X_2 \cdots X_m)$ can be defined similar to how (3.3) was extended to (3.7). Then the probability of occurrence can be bounded for $(\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_m) \in \mathcal{A}_\varepsilon^n(X_1 X_2 \cdots X_m)$ in a similar manner to (3.10) as

$$2^{-n(H(\boldsymbol{X})+\varepsilon)} \leq p(\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_m) \leq 2^{-n(H(\boldsymbol{X})-\varepsilon)} \tag{3.11}$$

where $\boldsymbol{X} = (X_1, X_2, \ldots, X_m)$.

Finally, we fix $\underline{x}$. The conditional (weak) typical set $\mathcal{A}_\varepsilon^n(Y|\underline{x})$ of length-n sequences is

$$\mathcal{A}_\varepsilon^n(Y|\underline{x}) = \left\{\underline{y} : (\underline{x}, \underline{y}) \in \mathcal{A}_\varepsilon^n(XY)\right\}. \tag{3.12}$$

In other words, $\mathcal{A}_\varepsilon^n(Y|\underline{x})$ is the set of all $\underline{y}$ sequences that are jointly typical with $\underline{x}$. For $\underline{x} \in \mathcal{A}_\varepsilon^n(X)$ and for sufficiently large n, the cardinality of the conditional typical set $\mathcal{A}_\varepsilon^n(Y|\underline{x})$ satisfies [26, Th. 15.2.2]

$$|\mathcal{A}_\varepsilon^n(Y|\underline{x})| \leq 2^{n(H(Y|X)+2\varepsilon)}. \tag{3.13}$$

3.3 $\mathcal{B}$-typicality

Definition 3.1 ($\mathcal{B}$**-typicality).** Let the input distribution $p(u)$ together with the transition law $p(v|u)$ determine the joint probability distribution $p(u, v) = p(u)p(v|u)$. Now we define

$$\mathcal{B}_{V,\varepsilon}^n(U) \triangleq \{\underline{u} : \underline{u} \in \mathcal{A}_\varepsilon^n(U) \text{ and } \Pr\{(\underline{u}, \underline{V}) \in \mathcal{A}_\varepsilon^n(UV) \mid \underline{U} = \underline{u})\} \geq 1 - \varepsilon\} \tag{3.14}$$

where $\underline{V}$ is the output sequence of a "channel" $p(v|u)$ when sequence $\underline{u}$ is input.

The set $\mathcal{B}_{V,\varepsilon}^n(U)$ in (3.14) guarantees that a sequence $\underline{u}$ in this $\mathcal{B}$-typical set, will with high probability lead to a sequence $\underline{v}$ that is jointly typical with $\underline{u}$. We note that U and/or V can be composite. The set $\mathcal{B}_{V,\varepsilon}^n(U)$ has three properties as stated in Lemma 3.1.

Lemma 3.1 ($\mathcal{B}$**-typicality properties).** The set $\mathcal{B}_{V,\varepsilon}^n(U)$ in Definition 3.1 has the following properties.

P_1 : For $\underline{u} \in \mathcal{B}_{V,\varepsilon}^n(U)$,

$$2^{-n(H(U)+\varepsilon)} \leq p(\underline{u}) \leq 2^{-n(H(U)-\varepsilon)}. \tag{3.15}$$

P_2 : For n large enough,

$$\sum_{\underline{u} \notin \mathcal{B}_{V,\varepsilon}^n(U)} p(\underline{u}) \leq \varepsilon.$$

P_3 : $|\mathcal{B}_{V,\varepsilon}^n(U)| \leq 2^{n(H(U)+\varepsilon)}$ holds for all n, while $|\mathcal{B}_{V,\varepsilon}^n(U)| \geq (1-\varepsilon)2^{n(H(U)-\varepsilon)}$ holds for n large enough.

3.3.1 Proof of $\mathcal{B}$-typicality Property P_1

We see from [26, eq. (3.6)] that for $\underline{u} \in \mathcal{A}_\varepsilon^n(U)$,

$$2^{-n(H(U)+\varepsilon)} \le p(\underline{u}) \le 2^{-n(H(U)-\varepsilon)}. \tag{3.16}$$

Due to Definition 3.1, each $\underline{u} \in \mathcal{B}_{V,\varepsilon}^n(U)$ is also in $\mathcal{A}_\varepsilon^n(U)$, more specifically, $\mathcal{B}_{V,\varepsilon}^n(U) \subseteq \mathcal{A}_\varepsilon^n(U)$. Consequently, (3.16) also holds for $\underline{u} \in \mathcal{B}_{V,\varepsilon}^n(U)$, which completes the proof of P_1.

3.3.2 Proof of $\mathcal{B}$-typicality Property P_2

Let $(\underline{U}, \underline{V})$ be independent and identically distributed with respect to $p(u, v)$. Then

$$\begin{aligned}
\Pr\{(\underline{U},\underline{V}) \in \mathcal{A}_\varepsilon^n(UV)\} &= \sum_{\underline{u}} p(\underline{u}) \sum_{\underline{v}:(\underline{u},\underline{v})\in\mathcal{A}_\varepsilon^n(UV)} p(\underline{v}|\underline{u}) \\
&= \sum_{\underline{u}\in\mathcal{B}_{V,\varepsilon}^n(U)} p(\underline{u}) \sum_{\underline{v}:(\underline{u},\underline{v})\in\mathcal{A}_\varepsilon^n(UV)} p(\underline{v}|\underline{u}) \\
&\quad + \sum_{\underline{u}\notin\mathcal{B}_{V,\varepsilon}^n(U)} p(\underline{u}) \sum_{\underline{v}:(\underline{u},\underline{v})\in\mathcal{A}_\varepsilon^n(UV)} p(\underline{v}|\underline{u}) \\
&\le \sum_{\underline{u}\in\mathcal{B}_{V,\varepsilon}^n(U)} p(\underline{u}) + \sum_{\underline{u}\notin\mathcal{B}_{V,\varepsilon}^n(U)} p(\underline{u})(1-\varepsilon) \qquad (3.17) \\
&= 1-\varepsilon+\varepsilon \sum_{\underline{u}\in\mathcal{B}_{V,\varepsilon}^n(U)} p(\underline{u}) \\
&= 1-\varepsilon+\varepsilon \Pr\{\underline{U} \in \mathcal{B}_{V,\varepsilon}^n(U)\}. \qquad (3.18)
\end{aligned}$$

Here (3.17) follows from Definition 3.1 which states that $\Pr\left\{(\underline{u},\underline{V}) \in \mathcal{A}_\varepsilon^n(UV) \middle| \underline{U} = \underline{u}\right\} < 1-\varepsilon$ for $\underline{u} \in \mathcal{A}_\varepsilon^n(U)$, if $\underline{u} \notin \mathcal{B}_{V,\varepsilon}^n(U)$. Then from (3.18), we obtain

$$\begin{aligned}
\Pr\{\underline{U} \in \mathcal{B}_{V,\varepsilon}^n(U)\} &\ge \frac{\Pr\{(U,V) \in \mathcal{A}_\varepsilon^n(UV)\} - 1 + \varepsilon}{\varepsilon} \\
&= 1 - \frac{\Pr\{(U,V) \notin \mathcal{A}_\varepsilon^n(UV)\}}{\varepsilon} \\
&\ge 1-\varepsilon \qquad (3.19)
\end{aligned}$$

for large enough n. Here (3.19) follows from [26, Th. 7.6.1] which states that $\Pr\{(\underline{U},\underline{V}) \in \mathcal{A}_\varepsilon^n(UV)\} \to 1$ as $n \to \infty$. This implies that $\Pr\{(\underline{U},\underline{V}) \notin \mathcal{A}_\varepsilon^n(UV)\} \le \varepsilon^2$ for positive ε and large enough n, which completes the proof of P_2.

3.3.3 Proof of $\mathcal{B}$-typicality Property P_3

We see from [26, Th. 3.1.2] that

$$|\mathcal{A}_\varepsilon^n(U)| \le 2^{n(H(U)+\varepsilon)}. \tag{3.20}$$

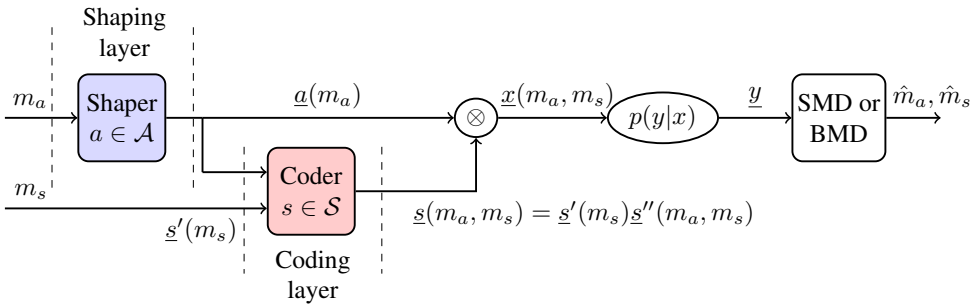

Figure 3.2: Sign-coding structure: Sign-coding is combined with amplitude shaping.

Since $\mathcal{B}^n_{V,\varepsilon}(U) \subseteq \mathcal{A}^n_\varepsilon(U)$ again by Definition 3.1, (3.20) also holds for $|\mathcal{B}^n_{V,\varepsilon}(U)|$. This proves the upper bound in P_3. We obtain the lower bound from (3.19) for n sufficiently large

$$\begin{aligned} 1-\varepsilon &\le \Pr\{\underline{U} \in \mathcal{B}^n_{V,\varepsilon}(U)\} \\ &\le \sum_{\underline{u} \in \mathcal{B}^n_{V,\varepsilon}(U)} 2^{-n(H(U)-\varepsilon)} \qquad (3.21) \\ &= |\mathcal{B}^n_{V,\varepsilon}(U)| 2^{-n(H(U)-\varepsilon)} \qquad (3.22) \end{aligned}$$

where (3.21) follows from (3.16).

3.4 Random Sign-coding Experiment

3.4.1 Sign-coding Setup

We cast the PAS structure shown in Fig. 3.1 (top) as a *sign-coding* structure as in Fig. 3.2. The sign-coding setup consists of two layers: a shaping layer and a coding layer.

Definition 3.2 (Sign-coding). For every message index pair (m_a, m_s), with uniform $m_a \in \{1, 2, \cdots, M_a\}$ and uniform $m_s \in \{1, 2, \cdots, M_s\}$, a sign-coding structure as shown in Fig. 3.2 consists of the following.

- A **shaping layer** that produces for every message index m_a, a length-n shaped amplitude sequence $\underline{a}(m_a)$ where the mapping is one-to-one. The set of amplitude sequences is assumed to be shaped but uncoded.
- An additional n_1-bit (uniform) information string in the form of a sign sequence part $\underline{s}'(m_s) = (s_1(m_s), s_2(m_s), \cdots, s_{n_1}(m_s))$ for every message index m_s.
- A **coding layer** that extends the first sign sequence part $\underline{s}'(m_s)$ for all m_a and m_s by adding $\underline{s}''(m_a, m_s) = (s_{n_1+1}(m_a, m_s), s_{n_1+2}(m_a, m_s), \cdots, s_n(m_a, m_s))$, a second

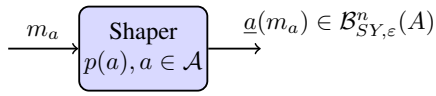

Figure 3.3: Shaping layer of the random sign-coding setup with SMD.

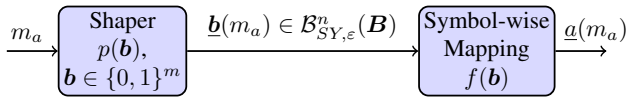

Figure 3.4: Shaping layer of the random sign-coding setup with BMD for 2^{m+1}-ASK.

length-n_2 (uniform) sign sequence part. This is obtained by using an encoder that produces redundant signs in the set $\mathcal{S}$ from $\underline{a}(m_a)$ and $\underline{s}'(m_s)$. Here $n_1 + n_2 = n$.

Finally, the transmitted sequence is $\underline{x}(m_a, m_s) = \underline{a}(m_a) \otimes \underline{s}(m_a, m_s)$, where $\underline{s}(m_a, m_s) = (\underline{s}'(m_s), \underline{s}''(m_a, m_s))$. The sign-coding setup with $n_1 = 0$ ($\gamma = 0$) is called *basic sign-coding* (basic PAS, Fig. 2.7), while the setup with $n_1 > 0$ ($\gamma > 0$) is called *modified sign-coding* (modified PAS, Fig. 2.8).

3.4.2 Shaping Layer

When SMD is employed at the receiver, the shaping layer is as shown in Fig. 3.3. Here, let A be distributed with $p(a)$ over $\mathcal{A}$. Then, the shaper produces for every message index m_a a length-n amplitude sequence $\underline{a}(m_a) \in \mathcal{B}^n_{SY,\varepsilon}(A)$. We note that for this sign-coding setup, the rate is

$$R = \frac{1}{n}\log_2|M_a M_s| = \gamma + \frac{1}{n}\log_2|\mathcal{B}^n_{SY,\varepsilon}(A)| \geq H(A) + \gamma - 2\varepsilon \tag{3.23}$$

where the inequality in (3.23) follows for n large enough from P_3.

On the other hand, when BMD is used at the receiver, the shaping layer is as shown in Fig. 3.4. Here, let $\boldsymbol{B} = (B_1, B_2, \cdots, B_m)$ be distributed with $p(\boldsymbol{b}) = p(b_1, b_2, \cdots, b_m)$ over $\{0,1\}^m$. The shaper produces for every message index m_a an n-sequence of m-tuples $\underline{\boldsymbol{b}}(m_a) = (\underline{b}_1(m_a), \underline{b}_2(m_a), \cdots, \underline{b}_m(m_a)) \in \mathcal{B}^n_{SY,\varepsilon}(B_1 B_2 \cdots B_m)$. Then, each m-tuple is mapped to an amplitude sequence $\underline{a}(m_a)$ by a symbol-wise mapping function $f(\cdot)$. We note that for this sign-coding setup, the rate is:

$$R = \frac{1}{n}\log_2|M_a M_s| = \gamma + \frac{1}{n}\log_2|\mathcal{B}^n_{SY,\varepsilon}(\boldsymbol{B})| \geq H(\boldsymbol{B}) + \gamma - 2\varepsilon \tag{3.24}$$

where the inequality in (3.24) follows for n large enough from P_3.

To realize $f(\cdot)$, we label the channel inputs with $(m+1)$-bit strings. The amplitude is addressed by m amplitude bits $\boldsymbol{B} = (B_1, B_2, \cdots, B_m)$, while the sign is addressed by a

sign bit S. The symbol-wise mapping function $f(\cdot)$ in Fig. 3.4 uses the addressing $\boldsymbol{B} \Longleftrightarrow A$. We note that unlike the case in Sec. 2.4.1, we use $(S, B_1, B_2, \cdots, B_m)$ to denote a channel input instead of $(C_1, C_2, \cdots, C_{m+1})$ to emphasize the distinction between the sign and the amplitudes. The amplitudes and signs of $x \in \mathcal{X}$ are tabulated for 8-ASK in Table 3.1 along with an example of the mapping function $f(b_1, b_2)$, namely the BRGC discussed in Sec. 2.4.1.

Table 3.1: Input alphabet and mapping function for 8-ASK.

A	7	5	3	1	1	3	5	7
S	-1	-1	-1	-1	1	1	1	1
X	-7	-5	-3	-1	1	3	5	7
B_1	0	0	1	1	1	1	0	0
B_2	0	1	1	0	0	1	1	0

3.4.3 Decoding Rules

At the receiver, SMD finds the unique message index pair $(\hat{m}_a, \hat{m}_s)$ such that the corresponding amplitude-sign sequence is jointly typical with the received output sequence $\underline{y}$, i.e., $(\underline{a}(\hat{m}_a), \underline{s}(\hat{m}_a, \hat{m}_s), \underline{y}) \in \mathcal{A}_\varepsilon^n(ASY)$.

On the other hand, BMD finds the unique message index pair $(\hat{m}_a, \hat{m}_s)$ such that the corresponding bit and sign sequences are (individually) jointly typical with the received sequence $\underline{y}$, i.e., $(\underline{s}(\hat{m}_a, \hat{m}_s), \underline{y}) \in \mathcal{A}_\varepsilon^n(SY)$, and $(\underline{b}_j(\hat{m}_a), \underline{y}) \in \mathcal{A}_\varepsilon^n(B_jY)$ for $j = 1, 2, \cdots, m$. We note that the decoder can use the bit-metrics $p(b_{ji} = 1|y_i) = 1 - p(b_{ji} = 0|y_i)$ for $j = 1, 2, \cdots, m$ and $i = 1, 2, \cdots, n$ to find $p(\underline{b}_j|\underline{y})$. Here b_{ji} is the j^{th} bit of the i^{th} symbol. Together with $p(\underline{y})$ and $p(\underline{b}_j)$, the decoder can check whether $(\underline{b}_j, \underline{y}) \in \mathcal{A}_\varepsilon^n(B_jY)$. We note that B_j's are in general not uniform. A similar statement holds for the uniform sign S.

3.5 Achievable Information Rates for Sign-coding

Here we investigate AIRs of the sign-coding architecture in Fig. 3.2. We consider both SMD and BMD at the receiver. In what follows, four AIRs are presented. The proofs are based on $\mathcal{B}$-typicality, a variation of weak typicality, and random sign-coding arguments. As indicated in Definition 3.2, signs S are assumed to be uniform in the proofs. We have not applied weak typicality for continuous random variables, discussed in [26, Sec. 8.2], [92, Sec. 10.4] and [90, Sec. 20.13], since our channels are discrete-input. However, it is possible that a hybrid version of weak typicality could be developed that matches with discrete-input continuous-output channels.

Definition 3.3 (Achievable information rate). A rate R is said to be achievable if for every $\delta > 0$ and n large enough, there exists a sign-coding encoder and a decoder such that

$(1/n)\log_2(M_a M_s) \geq R - \delta$ and error probability $P_e \leq \delta$.

To derive AIRs, we will follow the classical approach, e.g., as in [26, Sec. 7.7], and upper-bound the average probability of error $\overline{P}_e$, averaged over all sign-codewords in the sign-codebook and averaged over all sign-codebooks. This way, we will demonstrate the existence of at least one good sign code. Again as in [26, Sec. 7.7] and as explained in Sec. 3.4.3, we decode by joint typicality: the decoder looks for a unique message index pair $(\hat{m}_a, \hat{m}_s)$ for which the corresponding amplitude-sign sequence $(\underline{a}, \underline{s})$ is jointly typical with the received sequence $\underline{y}$.

By the properties of weak typicality and $\mathcal{B}$-typicality, the transmitted amplitude-sign sequence and the received sequence are jointly typical with a high probability for n large enough. We call the event that the transmitted amplitude-sign sequence is not jointly typical with the received sequence the *first error event* with average probability $\overline{P}_e(1)$. Furthermore, the probability that any other (not transmitted) amplitude-sign sequence is jointly typical with the received sequence vanishes for asymptotically large n. We call the event that there is another amplitude-sign sequence that is jointly typical with the received sequence the *second error event* with average probability $\overline{P}_e(2)$. Observing that these events are **not** disjoint, we can write [26, eq. (7.75)]

$$\overline{P}_e \leq \overline{P}_e(1) + \overline{P}_e(2). \tag{3.25}$$

3.5.1 Sign-coding with Symbol-metric Decoding

Theorem 3.1 (Basic sign-coding with SMD). For a discrete memoryless channel with amplitude shaping and basic sign-coding, the rate

$$R_{\text{SMD}}^{\gamma=0} = \max_{p(a):H(A)\leq I(SA;Y)} H(A) \tag{3.26}$$

is achievable using SMD.

Theorem 3.1 implies that for a memoryless channel, the rate $R = H(A)$ is achievable with basic sign-coding, as long as $H(A) \leq I(SA;Y) = I(X;Y)$ is satisfied. For the AWGN channel, this means that a range of rate-SNR pairs is achievable. One of these points, $H(A) = I(SA;Y)$, is on the capacity-SNR curve. The existence of an amplitude distribution $p(a)$ for which $H(A) = I(SA;Y)$ can be seen by first observing that

$$H(A) = I(SA;Y) = H(SA) - H(SA|Y) = H(A) + 1 - H(X|Y) \tag{3.27}$$

since the capacity-achieving distributions are symmetric, i.e., $H(SA) = H(A) + 1$. Consequently, $H(A) = I(SA;Y)$ and $H(X|Y) = 1$ are equivalent conditions. Then the existence of a point on the capacity-SNR curve for which $H(X|Y) = 1$ can be seen by observing that as SNR $\to \infty$, the equivocation $H(X|Y) \downarrow 0$. On the other hand as SNR $\downarrow 0$, the equivocation $H(X|Y) \to H(X) = H(A) + 1 \geq 1$. Then by the continuity of $H(X|Y)$ in the SNR, there is an SNR for which $H(X|Y) = 1$. The existence of this point on the capacity-SNR

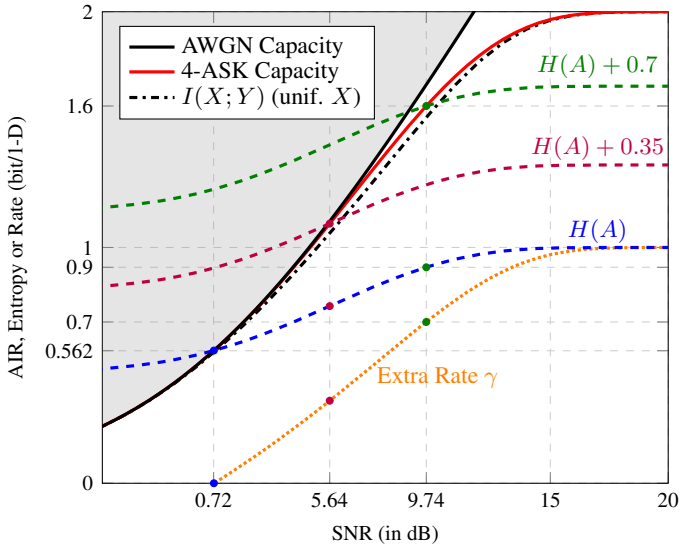

Figure 3.5: Sign-coding with SMD for 4-ASK. All $C_{\text{4-ASK}} \geq 0.562$ bit/1-D can be achieved with sign-coding.

curve will also be observed from Fig. 3.5 discussed in Example 3.1. We note that here, "capacity" indicates the largest achievable rate using $\mathcal{X}$ as the channel input alphabet under the average power constraint $E[X^2] \leq P$.

Theorem 3.2 (Modified sign-coding with SMD). For a discrete memoryless channel with amplitude shaping and modified sign-coding, the rate

$$R_{\text{SMD}}^{\gamma>0} = \max_{p(a),\gamma:H(A)+\gamma\leq I(SA;Y)} H(A)+\gamma \tag{3.28}$$

is achievable using SMD for $\gamma < 1$.

Theorem 3.2 implies that for a memoryless channel, the rate $H(A)+\gamma$ is achievable with modified sign-coding, as long as $R = H(A)+\gamma \leq I(SA;Y) = I(X;Y)$ is satisfied. For the AWGN channel, this means that all points on the capacity-SNR curve for which $H(X|Y) \leq 1-\gamma$ are achievable. This follows from

$$H(A)+\gamma \leq I(SA;Y) = H(SA) - H(SA|Y) = H(A)+1-H(X|Y), \tag{3.29}$$

i.e., the constraint in the maximization in (3.28).

Example 3.1. We consider the AWGN channel[3] with average power constraint $E[X^2] \leq P$. Figure 3.5 shows the capacity of 4-ASK

$$C_{\text{4-ASK}} = \max_{\substack{p(x):\mathcal{X}=\{-3,-1,+1,+3\},\\ E[X^2]\leq P}} I(X;Y) \tag{3.30}$$

together with the amplitude entropy $H(A)$ of the distribution that achieves this capacity. Here SNR $= E[X^2]/\sigma^2$, and σ^2 is the noise variance. Basic sign-coding achieves capacity only for SNR $= 0.72$ dB, i.e., at the point where $H(A) = I(X;Y)$ which is $C_{\text{4-ASK}} = 0.562$ bit/1-D. We see from Fig. 3.5 that the shaping gap is negligible around this point, i.e., the capacity $C_{\text{4-ASK}}$ of 4-ASK and the MI $I(X;Y)$ for uniform $p(x)$ are virtually the same. On the other hand, this gap is significant for larger rates, e.g., it is around 0.42 dB at 1.6 bit/1-D. To achieve rates larger than 0.562 bit/1-D on the capacity-SNR curve, modified sign-coding ($\gamma > 0$) is required. At a given SNR, $C_{\text{4-ASK}}$ can be written as $C_{\text{4-ASK}} = H(A) + \gamma$, i.e., when $H(A)$ curve is shifted above by γ, the crossing point is again at $C_{\text{4-ASK}}$ for that SNR. We have also plotted the additional rate $\gamma = C_{\text{4-ASK}} - H(A)$ in Fig. 3.5. As an example, at SNR $= 9.74$ dB, $C_{\text{ASK}} = H(A) + \gamma = 1.6$ can be achieved with modified sign-coding where $H(A) = 0.9$ and $\gamma = 0.7$. We observe that sign-coding achieves the capacity of 4-ASK for SNR ≥ 0.72 dB.

3.5.1.1 Proof of Theorem 3.1

For the error of the first kind, we can write:

$$\begin{aligned}
\overline{P}_e(1) &= \sum_{m_a=1}^{M_a} \frac{1}{M_a} \sum_{\underline{s}\in\mathcal{S}^n} p(\underline{s}) \sum_{\underline{y}\in\mathcal{Y}^n} p(\underline{y}|\underline{a}(m_a),\underline{s}) \mathbb{1}[(\underline{a}(m_a),\underline{s},\underline{y}) \notin \mathcal{A}_\varepsilon^n(ASY)] \\
&= \sum_{m_a} \frac{1}{M_a} \sum_{\underline{s}} \sum_{\underline{y}} p(\underline{s},\underline{y}|\underline{a}(m_a)) \mathbb{1}[(\underline{a}(m_a),\underline{s},\underline{y}) \notin \mathcal{A}_\varepsilon^n] \qquad (3.31) \\
&= \sum_{m_a} \frac{1}{M_a} \Pr\left\{(\underline{a}(m_a),\underline{S},\underline{Y}) \notin \mathcal{A}_\varepsilon^n \middle| \underline{A} = \underline{a}(m_a)\right\} \\
&\leq \sum_{m_a} \frac{\varepsilon}{M_a} \qquad (3.32) \\
&= \varepsilon \qquad (3.33)
\end{aligned}$$

where we simplified the notation by replacing $m_a = 1, 2, \cdots, M_a$ by m_a, $\underline{s} \in \mathcal{S}^n$ by $\underline{s}$, and $\underline{y} \in \mathcal{Y}^n$ by $\underline{y}$ in (3.31). Furthermore, we dropped the index of the typical set $\mathcal{A}_\varepsilon^n(ASY)$ and used $\mathcal{A}_\varepsilon^n$ instead. We will follow these notations for summations and typical sets for the

[3] In this case, we assume that the channel output Y is a quantized version of the continuous channel output $X+Z$. Furthermore, we assume that this quantization is with a resolution high enough that the discrete-output channel is an accurate model for the underlying continuous-output channel. Therefore, the achievability results we obtained for discrete memoryless channels carry over to the discrete-input AWGN channel.

rest of the chapter, assuming for the latter that the index of the typical set will be clear from the context. To obtain (3.31), we used $p(\underline{s})p(\underline{y}|\underline{a}(m_a),\underline{s}) = p(\underline{s},\underline{y}|\underline{a}(m_a))$. Then, (3.32) is a direct consequence of Definition 3.1 since $\underline{a}(m_a) \in \mathcal{B}^n_{SY,\varepsilon}(A)$ for $m_a = 1, 2, \cdots, M_a$.

For the error of the second kind, we can write:

$$
\begin{aligned}
&\overline{P}_e(2) \\
&\leq \sum_{m_a} \frac{1}{M_a} \sum_{\underline{s}} p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a),\underline{s}) \sum_{k_a=1,k_a\neq m_a}^{M_a} \sum_{\tilde{\underline{s}}\in\mathcal{S}^n} p(\tilde{\underline{s}}) \mathbb{1}[(\underline{a}(k_a),\tilde{\underline{s}},\underline{y}) \in \mathcal{A}^n_\varepsilon] \\
&= M_a \sum_{m_a} \sum_{\underline{s}} \frac{p(\underline{s})}{M_a} \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a),\underline{s}) \sum_{k_a\neq m_a} \sum_{\tilde{\underline{s}}} \frac{p(\tilde{\underline{s}})}{M_a} \mathbb{1}[(\underline{a}(k_a),\tilde{\underline{s}},\underline{y}) \in \mathcal{A}^n_\varepsilon] \qquad (3.34) \\
&\leq M_a 2^{6n\varepsilon} \sum_{m_a} \sum_{\underline{s}} p(\underline{a}(m_a))p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a),\underline{s}) \\
&\qquad\qquad \cdot \sum_{k_a\neq m_a} \sum_{\tilde{\underline{s}}} p(\underline{a}(k_a))p(\tilde{\underline{s}}) \mathbb{1}[(\underline{a}(k_a),\tilde{\underline{s}},\underline{y}) \in \mathcal{A}^n_\varepsilon] \qquad (3.35) \\
&\leq M_a 2^{6n\varepsilon} \sum_{\underline{a}\in\mathcal{A}^n} \sum_{\underline{s}} p(\underline{a})p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{a},\underline{s}) \sum_{\tilde{\underline{a}}\in\mathcal{A}^n} \sum_{\tilde{\underline{s}}} p(\tilde{\underline{a}})p(\tilde{\underline{s}}) \mathbb{1}[(\tilde{\underline{a}},\tilde{\underline{s}},\underline{y}) \in \mathcal{A}^n_\varepsilon] \qquad (3.36) \\
&= M_a 2^{6n\varepsilon} \sum_{(\underline{y},\underline{\tilde{x}})\in\mathcal{A}^n_\varepsilon} p(\underline{\tilde{x}})p(\underline{y}) \qquad (3.37) \\
&\leq 2^{n(H(A)+\varepsilon)} 2^{6n\varepsilon} |\mathcal{A}^n_\varepsilon(XY)| 2^{-n(H(X)-\varepsilon)} 2^{-n(H(Y)-\varepsilon)} \qquad (3.38) \\
&\leq 2^{n(H(A)+7\varepsilon)} 2^{n(H(X,Y)+\varepsilon)} 2^{-n(H(X)-\varepsilon)} 2^{-n(H(Y)-\varepsilon)} \qquad (3.39) \\
&= 2^{n(H(A)-I(SA;Y)+10\varepsilon)} \qquad (3.40)
\end{aligned}
$$

where we simplified the notation by replacing $k_a = 1, 2, \cdots, M_a : k_a \neq m_a$ by $k_a \neq m_a$, and $\tilde{\underline{s}} \in \mathcal{S}^n$ by $\tilde{\underline{s}}$ in (3.34). We will follow these notations for the rest of the chapter. Then

(3.35) follows for n sufficiently large and for $\underline{a} \in \mathcal{B}^n_{SY,\varepsilon}(A)$ from

$$
\begin{aligned}
\frac{1}{M_a} = \frac{1}{|\mathcal{B}^n_{SY,\varepsilon}(A)|} &\leq \frac{2^{-n(H(A)-\varepsilon))}}{1-\varepsilon} \qquad (3.41) \\
&= \frac{2^{2n\varepsilon}}{1-\varepsilon} 2^{-n(H(A)+\varepsilon)} \\
&\leq \frac{2^{2n\varepsilon}}{1-\varepsilon} p(\underline{a}) \qquad (3.42) \\
&\leq 2^{3n\varepsilon} p(\underline{a}) \qquad (3.43)
\end{aligned}
$$

where (3.41) follows from the $\mathcal{B}$-typicality property P_3, (3.42) follows from the $\mathcal{B}$-typicality property P_1, and (3.43) holds for all large enough n,

(3.36) follows from summing over $\underline{a} \in \mathcal{A}^n$ instead of over $\underline{a}(m_a) \in \mathcal{B}_\varepsilon^n$, and over $\tilde{\underline{a}} \in \mathcal{A}^n$ instead of $\underline{a}(k_a) \in \mathcal{B}_\varepsilon^n$ for $k_a \neq m_a$,

(3.37) is obtained by working out the summations over $\underline{a}$ and $\underline{s}$ and by replacing $(\tilde{\underline{s}}, \tilde{\underline{a}})$ with $\tilde{\underline{x}}$,

(3.38) follows from $M_a = |\mathcal{B}_\varepsilon^n(A)| \leq 2^{n(H(A)+\varepsilon)}$, i.e., the $\mathcal{B}$-typicality property P_3, and from (3.6), and

(3.39) follows from (3.9).

The conclusion from (3.40) is that for $H(A) < I(X;Y) - 10\varepsilon$, the error probability of the second kind

$$\overline{P}_e(2) \leq \varepsilon \tag{3.44}$$

for n large enough. Using (3.33) and (3.44) in (3.25), we find that the error probability averaged over all sign-codewords in the sign-codebook, and averaged over all sign-codebooks $\overline{P}_e \leq 2\varepsilon$ for n large enough. This implies the existence of a basic sign-code with total error probability $P_e = \Pr\{\hat{M}_a \neq M_a\} \leq 2\varepsilon$. This holds for all $\varepsilon > 0$, and therefore, the rate

$$R = H(A) \leq I(X;Y) \tag{3.45}$$

is achievable with basic sign-coding, which concludes the proof of Theorem 3.1.

3.5.1.2 Proof of Theorem 3.2

For the error of the first kind, we can write:

$$\begin{aligned}
&\overline{P}_e(1) \\
&= \sum_{m_a} \frac{1}{M_a} \sum_{m_s=1}^{M_s} \frac{1}{2^{n_1}} \sum_{\underline{s}'' \in \mathcal{S}^{n_2}} p(\underline{s}'') \\
&\qquad \cdot \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'') \mathbb{1}[(\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'', \underline{y}) \notin \mathcal{A}_\varepsilon^n] \\
&= \sum_{m_a} \frac{1}{M_a} \sum_{m_s} \sum_{\underline{s}''} 2^{-n} \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'') \mathbb{1}[(\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'', \underline{y}) \notin \mathcal{A}_\varepsilon^n] \qquad (3.46) \\
&= \sum_{m_a} \frac{1}{M_a} \sum_{m_s} \sum_{\underline{s}''} \sum_{\underline{y}} p(\underline{s}'(m_s)\underline{s}'', \underline{y}|\underline{a}(m_a)) \mathbb{1}[(\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'', \underline{y}) \notin \mathcal{A}_\varepsilon^n] \qquad (3.47) \\
&= \sum_{m_a} \frac{1}{M_a} \Pr\left\{(\underline{a}(m_a), \underline{S}, \underline{Y}) \notin \mathcal{A}_\varepsilon^n \middle| \underline{A} = \underline{a}(m_a)\right\} \\
&\leq \sum_{m_a} \frac{\varepsilon}{M_a} \qquad (3.48) \\
&= \varepsilon \qquad (3.49)
\end{aligned}$$

where we simplified the notation by replacing $\underline{s}'' \in \mathcal{S}^{n_2}$ by $\underline{s}''$ and $m_s = 1, 2, \cdots, M_s$ by m_s in (3.46). We will follow these notations for the rest of the chapter. To obtain (3.46), we used the fact that $\underline{S}''$ is uniform, more precisely $p(\underline{s}'') = 2^{-n_2}$. To obtain (3.47), we used the fact that $\underline{S}'$ is also uniform, and then, $2^{-n} p(\underline{y}|\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'') = p(\underline{s}'(m_s)\underline{s}'', \underline{y}|\underline{a}(m_a))$. Then, (3.48) is a direct consequence of Definition 3.1 since $\underline{a}(m_a) \in \mathcal{B}^n_{SY,\varepsilon}(A)$ for $m_a = 1, 2, \cdots, M_a$.

For the error of the second kind, we obtain:

$$
\begin{aligned}
\overline{P}_e(2) &\le \sum_{m_a} \frac{1}{M_a} \sum_{m_s} \frac{1}{2^{n_1}} \sum_{\underline{s}''} p(\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\qquad \cdot \sum_{(k_a,k_s)\neq(m_a,m_s)} \sum_{\tilde{\underline{s}}''} p(\tilde{\underline{s}}'') \mathbb{1}[(\underline{a}(k_a), \underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}^n_\varepsilon] \\
&= M_a 2^{n_1} \sum_{m_a, m_s, \underline{s}''} \frac{2^{-n}}{M_a} \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\qquad \cdot \sum_{(k_a,k_s)\neq(m_a,m_s)} \sum_{\tilde{\underline{s}}''} \frac{2^{-n}}{M_a} \mathbb{1}[(\underline{a}(k_a), \underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}^n_\varepsilon] \quad (3.50) \\
&= M_a 2^{n_1} \sum_{m_a, m_s, \underline{s}''} \frac{2^{-n}}{M_a} \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\qquad \cdot \sum_{k_a\neq m_a, k_s, \tilde{\underline{s}}''} \frac{2^{-n}}{M_a} \mathbb{1}[(\underline{a}(k_a), \underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}^n_\varepsilon] \\
&\quad + 2^{n_1} \sum_{m_a, m_s, \underline{s}''} \frac{2^{-n}}{M_a} \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\qquad \cdot \sum_{k_s\neq m_s, \tilde{\underline{s}}''} 2^{-n} \mathbb{1}[(\underline{a}(m_a), \underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}^n_\varepsilon] \quad (3.51) \\
&\le M_a 2^{n_1} 2^{6n\varepsilon} \sum_{m_a, m_s, \underline{s}''} p(\underline{a}(m_a)) p(\underline{s}'(m_s)\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\qquad \cdot \sum_{k_a\neq m_a, k_s, \tilde{\underline{s}}''} p(\underline{a}(k_a)) p(\underline{s}'(k_s)\tilde{\underline{s}}'') \mathbb{1}[(\underline{a}(k_a), \underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}^n_\varepsilon] \\
&\quad + 2^{n_1} 2^{3n\varepsilon} \sum_{m_a, m_s, \underline{s}''} p(\underline{a}(m_a)) p(\underline{s}'(m_s)\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{a}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\qquad \cdot \sum_{k_s\neq m_s, \tilde{\underline{s}}''} p(\underline{s}'(k_s)\tilde{\underline{s}}'') \mathbb{1}[(\underline{a}(m_a), \underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}^n_\varepsilon]. \quad (3.52)
\end{aligned}
$$

Here, we replaced nested summations over m_a, m_s, and $\underline{s}'$ by a single summation over $(m_a, m_s, \underline{s}')$ for the sake of better readability. We will use this notation for the rest of the chapter. Then

(3.50) follows from $n = n_1 + n_2$ and from the fact that $\underline{S}''$ is uniform, more precisely, $p(\underline{s}'') = 2^{-n_2}$,

(3.51) is obtained by splitting $(k_a, k_s) \neq (m_a, m_s)$ into $\{k_a \neq m_a, k_s\}$ and $\{k_a = m_a, k_s \neq m_s\}$, and

(3.52) follows for n sufficiently large and for $\underline{a} \in \mathcal{B}^n_{SY,\varepsilon}(A)$ from

$$\frac{1}{M_a} \overset{(3.43)}{\leq} 2^{3n\varepsilon} p(\underline{a}),$$

and from $p(\underline{s}'\underline{s}'') = 2^{-n}$.

From (3.52), we obtain:

$$\begin{aligned}
&\overline{P}_e(2) \\
&\leq M_a 2^{n_1} 2^{6n\varepsilon} \sum_{\underline{a},\underline{s}'\underline{s}''} p(\underline{a}) p(\underline{s}'\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{a},\underline{s}'\underline{s}'') \sum_{\tilde{\underline{a}},\tilde{\underline{s}}'\tilde{\underline{s}}''} p(\tilde{\underline{a}}) p(\tilde{\underline{s}}'\tilde{\underline{s}}'') \mathbb{1}[(\tilde{\underline{a}},\tilde{\underline{s}}'\tilde{\underline{s}}'',\underline{y}) \in \mathcal{A}^n_\varepsilon] \\
&\quad + 2^{n_1} 2^{3n\varepsilon} \sum_{\underline{a},\underline{s}'\underline{s}''} p(\underline{a}) p(\underline{s}'\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{a},\underline{s}'\underline{s}'') \sum_{\tilde{\underline{s}}'\tilde{\underline{s}}''} p(\tilde{\underline{s}}'\tilde{\underline{s}}'') \mathbb{1}[(\underline{a},\tilde{\underline{s}}'\tilde{\underline{s}}'',\underline{y}) \in \mathcal{A}^n_\varepsilon] \qquad (3.53) \\
&= M_a 2^{n_1} 2^{6n\varepsilon} \sum_{\underline{a},\underline{s}} p(\underline{a}) p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{a},\underline{s}) \sum_{\tilde{\underline{a}},\tilde{\underline{s}}} p(\tilde{\underline{a}}) p(\tilde{\underline{s}}) \mathbb{1}[(\tilde{\underline{a}},\tilde{\underline{s}},\underline{y}) \in \mathcal{A}^n_\varepsilon] \\
&\quad + 2^{n_1} 2^{3n\varepsilon} \sum_{\underline{a},\underline{s}} p(\underline{a}) p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{a},\underline{s}) \sum_{\tilde{\underline{s}}} p(\tilde{\underline{s}}) \mathbb{1}[(\underline{a},\tilde{\underline{s}},\underline{y}) \in \mathcal{A}^n_\varepsilon], \qquad (3.54)
\end{aligned}$$

where

(3.53) follows from summing over $\underline{a} \in \mathcal{A}^n$ instead of over $\underline{a}(m_a) \in \mathcal{B}^n_\varepsilon$, and over $\tilde{\underline{a}} \in \mathcal{A}^n$ instead of $\underline{a}(k_a) \in \mathcal{B}^n_\varepsilon$ for $k_a \neq m_a$. Moreover, it follows from summing over $\underline{s}' \in \mathcal{S}^{n_1}$ instead of $\underline{s}'(k_s)$ for $k_s = 1, 2, \cdots, M_s$ and $k_s \neq m_s$, and

(3.54) follows from substituting $\underline{s}$ for $\underline{s}'\underline{s}''$, and $\tilde{\underline{s}}$ for $\tilde{\underline{s}}'\tilde{\underline{s}}''$.

Finally, from (3.54), we obtain:

$$\begin{aligned}
\overline{P}_e(2) &= M_a 2^{n_1} 2^{6n\varepsilon} \sum_{\underline{y}} p(\underline{y}) \sum_{\tilde{\underline{x}}} p(\tilde{\underline{x}}) \mathbb{1}[(\tilde{\underline{x}},\underline{y}) \in \mathcal{A}^n_\varepsilon] \\
&\quad + 2^{n_1} 2^{3n\varepsilon} \sum_{\underline{a},\underline{y}} p(\underline{a},\underline{y}) \sum_{\tilde{\underline{s}}} p(\tilde{\underline{s}}) \mathbb{1}[(\underline{a},\tilde{\underline{s}},\underline{y}) \in \mathcal{A}^n_\varepsilon] \qquad (3.55) \\
&\leq 2^{n(H(A)+\varepsilon)} 2^{n\gamma} 2^{6n\varepsilon} |\mathcal{A}^n_\varepsilon(XY)| 2^{-n(H(X)-\varepsilon)} 2^{-n(H(Y)-\varepsilon)} \\
&\quad + 2^{n\gamma} 2^{3n\varepsilon} |\mathcal{A}^n_\varepsilon(SAY)| 2^{-n(H(A,Y)-\varepsilon)} 2^{-n(H(S)-\varepsilon)} \qquad (3.56) \\
&\leq 2^{n(H(A)+7\varepsilon)} 2^{n\gamma} 2^{n(H(X,Y)+\varepsilon)} 2^{-n(H(X)-\varepsilon)} 2^{-n(H(Y)-\varepsilon)} \\
&\quad + 2^{n\gamma} 2^{3n\varepsilon} 2^{n(H(S,A,Y)+\varepsilon)} 2^{-n(H(A,Y)-\varepsilon)} 2^{-n(H(S)-\varepsilon)} \qquad (3.57) \\
&= 2^{n(H(A)+\gamma+10\varepsilon-I(X;Y))} + 2^{n(\gamma+6\varepsilon-I(S;A,Y))}. \qquad (3.58)
\end{aligned}$$

Here, we substituted $n_1 = n\gamma$ in (3.56). Then

- (3.55) is obtained by working out the summations over $\underline{a}, \underline{s}$ in the first part and $\underline{s}$ in the second part. Moreover, we replaced $(\tilde{\underline{s}}, \tilde{\underline{a}})$ with $\tilde{\underline{x}}$,
- (3.56) is obtained using for the first part that $M_a = |\mathcal{B}_\varepsilon^n(A)| \leq 2^{n(H(A)+\varepsilon)}$, i.e., the $\mathcal{B}$-typicality property P_3 and (3.6). For the second part, we used (3.6) for $p(\underline{s})$ and (3.10) for $p(\underline{a}, \underline{y})$, and
- (3.57) follows from (3.9) and its extension to jointly typical triplets, more precisely, it follows from $|\mathcal{A}_\varepsilon^n(SAY)| \leq 2^{n(H(S,A,Y)+\varepsilon)}$.

The conclusion from (3.58) is that for $H(A) + \gamma < I(X;Y) - 10\varepsilon$ and for $\gamma < I(S;A,Y) - 6\varepsilon$, the error probability of the second kind

$$\overline{P}_e(2) \leq \varepsilon \tag{3.59}$$

for n large enough. The first constraint, i.e., $H(A) + \gamma < I(X;Y) - 10\varepsilon$, already implies the second constraint, i.e., $\gamma < I(S;A,Y) - 6\varepsilon$, since

$$\begin{aligned}
\gamma &< I(X;Y) - H(A) - 10\varepsilon \\
&\leq I(S,A;Y) - I(A;Y) - 10\varepsilon && (3.60)\\
&= I(S;Y|A) - 10\varepsilon && (3.61)\\
&\leq I(S;Y|A) + I(S;A) - 10\varepsilon \\
&= I(S;A,Y) - 10\varepsilon && (3.62)
\end{aligned}$$

where we substituted (S, A) for X in (3.60). Here, (3.60) follows from [26, Th. 2.4.1], and both (3.61) and (3.62) follow from the chain rule for MI [26, Th. 2.5.2].

Using (3.49) and (3.59) in (3.25), we find that the error probability averaged over all sign-codewords in the modified sign-codebook, and averaged over all modified sign-codebooks $\overline{P}_e \leq 2\varepsilon$ for n large enough. This implies the existence of a modified sign-code with total error probability $P_e = \Pr\{(\hat{M}_a, \hat{M}_s) \neq (M_a, M_s)\} \leq 2\varepsilon$. This holds for all $\varepsilon > 0$, and thus, the rate

$$R = H(A) + \gamma \leq I(X;Y) \tag{3.63}$$

is achievable with modified sign-coding, which concludes the proof of Theorem 3.2.

3.5.2 Sign-coding with Bit-metric Decoding

The following theorems give AIRs for sign-coding with BMD.

Theorem 3.3 (Basic sign-coding with BMD). For a discrete memoryless channel with amplitude shaping using 2^{m+1}-ASK and basic sign-coding, the rate

$$R_{\text{BMD}}^{\gamma=0} = \max_{p(\boldsymbol{b}):H(\boldsymbol{B})\leq R_{\text{BMD}}(p(x))} H(\boldsymbol{B}) \tag{3.64}$$

is achievable using BMD. Here $\boldsymbol{B} = (B_1, B_2, \ldots, B_m)$, $p(\boldsymbol{b}) = p(b_1, b_2, \ldots, b_m)$, $p(x) = p(s, b_1, b_2, \ldots, b_m)$, and $R_{\text{BMD}}(p(x))$ is as defined in (2.31).

Theorem 3.4 (Modified sign-coding with BMD). For a discrete memoryless channel with amplitude shaping using 2^{m+1}-ASK and modified sign-coding, the rate

$$R_{\text{BMD}}^{\gamma>0} = \max_{p(\boldsymbol{b}),\gamma: H(\boldsymbol{B})+\gamma \le R_{\text{BMD}}(p(x))} H(\boldsymbol{B}) + \gamma \tag{3.65}$$

is achievable using BMD for $\gamma < 1$.

Theorems 3.3 and 3.4 imply that for a memoryless channel, the rate $R = H(\boldsymbol{B}) + \gamma = H(A) + \gamma$ is achievable with sign-coding and BMD, as long as $R \le R_{\text{BMD}}$ is satisfied.

Remark 3.1 (Random sign-coding with binary linear codes). An amplitude can be represented by m bits. We can uniformly generate a code matrix with mn rows of length n. This matrix can be used to produce sign sequences. This results in the pairwise independence of any two different sign sequences, as is explained in the proof of [25, Th. 6.2.1]. Inspection of the proof of our Theorem 3.1 shows that only the pairwise independence of sign sequences is needed. Therefore, achievability can also be obtained with a binary linear code. Note that our linear code can also be seen as a systematic code that generates parity. The code rate of the corresponding systematic code is $m/(m+1)$. For BMD, similar reasoning shows that linear codes lead to achievability, and also for modified sign-coding achievability follows for binary linear codes. The rate of the systematic code that corresponds to the modified setting is $(m+\gamma)/(m+1)$.

3.5.2.1 Proof of Theorem 3.3

For the error of the first kind, we can write:

$$\begin{aligned}
\overline{P}_e(1) &= \sum_{m_a} \frac{1}{M_a} \sum_{\underline{s}} p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}) && (3.66)\\
&\qquad \cdot \mathbb{1}\left[\bigcup_{i=1}^{m} ((\underline{b}_i(m_a), \underline{y}) \notin \mathcal{A}_\varepsilon^n) \bigcup ((\underline{s}, \underline{y}) \notin \mathcal{A}_\varepsilon^n)\right] \\
&\le \sum_{m_a} \frac{1}{M_a} \sum_{\underline{s}} \sum_{\underline{y}} p(\underline{s}, \underline{y}|\underline{\boldsymbol{b}}(m_a)) \mathbb{1}[(\underline{\boldsymbol{b}}(m_a), \underline{s}, \underline{y}) \notin \mathcal{A}_\varepsilon^n] && (3.67)\\
&= \sum_{m_a} \frac{1}{M_a} \Pr\left\{(\underline{\boldsymbol{b}}(m_a), \underline{S}, \underline{Y}) \notin \mathcal{A}_\varepsilon^n \middle| \underline{\boldsymbol{B}} = \underline{\boldsymbol{b}}(m_a)\right\} && (3.68)\\
&\le \sum_{m_a} \frac{\varepsilon}{M_a} && (3.69)\\
&= \varepsilon, && (3.70)
\end{aligned}$$

where we used $\underline{\boldsymbol{b}}(m_a)$ to denote $(\underline{b}_1(m_a), \underline{b}_2(m_a), \ldots, \underline{b}_m(m_a))$ in (3.66) and $\underline{\boldsymbol{B}}$ to denote $(\underline{B}_1, \underline{B}_2, \ldots, \underline{B}_m)$ in (3.68). Then, we used $p(\underline{s})p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}) = p(\underline{s}, \underline{y}|\underline{\boldsymbol{b}}(m_a))$ in (3.67). Here, (3.67) follows from the fact that if at least one of $\underline{b}_1(m_a), \underline{b}_2(m_a), \ldots, \underline{b}_m(m_a)$ or $\underline{s}$ is not jointly typical with $\underline{y}$, then $(\underline{\boldsymbol{b}}(m_a), \underline{s}, \underline{y})$ is not jointly typical. Then, (3.69) is a direct consequence of Definition 3.1 since $\underline{\boldsymbol{b}}(m_a) \in \mathcal{B}^n_{SY,\varepsilon}(B_1 B_2 \cdots B_m)$ for $m_a = 1, 2, \cdots, M_a$.

For the error of the second kind, we can write:

$$
\begin{aligned}
&\overline{P}_e(2) \\
&\le \sum_{m_a} \frac{1}{M_a} \sum_{\underline{s}} p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}) \\
&\qquad \cdot \sum_{k_a \neq m_a} \sum_{\tilde{\underline{s}}} p(\tilde{\underline{s}}) \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i(k_a), \underline{y}) \in \mathcal{A}^n_\varepsilon) \bigcap ((\tilde{\underline{s}}, \underline{y}) \in \mathcal{A}^n_\varepsilon) \right] \\
&= M_a \sum_{m_a} \sum_{\underline{s}} \frac{p(\underline{s})}{M_a} \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}) \\
&\qquad \cdot \sum_{k_a \neq m_a} \sum_{\tilde{\underline{s}}} \frac{p(\tilde{\underline{s}})}{M_a} \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i(k_a), \underline{y}) \in \mathcal{A}^n_\varepsilon) \bigcap ((\tilde{\underline{s}}, \underline{y}) \in \mathcal{A}^n_\varepsilon) \right] \\
&\le M_a 2^{6n\varepsilon} \sum_{m_a} \sum_{\underline{s}} p(\underline{\boldsymbol{b}}(m_a)) p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}) \qquad (3.71) \\
&\qquad \cdot \sum_{k_a \neq m_a} \sum_{\tilde{\underline{s}}} p(\tilde{\underline{s}}) p(\underline{\boldsymbol{b}}(k_a)) \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i(k_a), \underline{y}) \in \mathcal{A}^n_\varepsilon) \bigcap ((\tilde{\underline{s}}, \underline{y}) \in \mathcal{A}^n_\varepsilon) \right] \\
&\le M_a 2^{6n\varepsilon} \sum_{\underline{\boldsymbol{b}} \in \{0,1\}^{mn}} \sum_{\underline{s}} p(\underline{\boldsymbol{b}}) p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}, \underline{s}) \qquad (3.72) \\
&\qquad \cdot \sum_{\tilde{\underline{\boldsymbol{b}}} \in \{0,1\}^{mn}} \sum_{\tilde{\underline{s}}} p(\tilde{\underline{s}}) p(\tilde{\underline{\boldsymbol{b}}}) \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i, \underline{y}) \in \mathcal{A}^n_\varepsilon) \bigcap ((\tilde{\underline{s}}, \underline{y}) \in \mathcal{A}^n_\varepsilon) \right] \\
&= M_a 2^{6n\varepsilon} \sum_{\underline{y}} p(\underline{y}) \sum_{\tilde{\underline{\boldsymbol{b}}}, \tilde{\underline{s}}} p(\tilde{\underline{\boldsymbol{b}}}, \tilde{\underline{s}}) \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i, \underline{y}) \in \mathcal{A}^n_\varepsilon) \bigcap ((\tilde{\underline{s}}, \underline{y}) \in \mathcal{A}^n_\varepsilon) \right] \qquad (3.73) \\
&\le 2^{n(H(\boldsymbol{B})+7\varepsilon)} |\mathcal{A}^n_\varepsilon(Y)| 2^{-n(H(Y)-\varepsilon)} \left(\prod_{i=1}^{m} |\mathcal{A}^n_\varepsilon(B_i|\underline{y})| \right) |\mathcal{A}^n_\varepsilon(S|\underline{y})| 2^{-n(H(\boldsymbol{B},S)-\varepsilon)} \qquad (3.74) \\
&\le 2^{n(H(\boldsymbol{B})+7\varepsilon)} 2^{n(H(Y)+\varepsilon)} 2^{-n(H(Y)-\varepsilon)} \\
&\qquad \cdot 2^{n\left(\left(\sum_{i=1}^{m} H(B_i|Y)\right)+H(S|Y)+2(m+1)\varepsilon\right)} 2^{-n(H(\boldsymbol{B},S)-\varepsilon)} \qquad (3.75) \\
&= 2^{n\left((H(\boldsymbol{B})-H(\boldsymbol{B},S)+\left(\sum_{i=1}^{m} H(B_i|Y)\right)+H(S|Y)+(12+2m)\varepsilon\right)} \qquad (3.76)
\end{aligned}
$$

where we used $\underline{\boldsymbol{b}}$ to denote $(\underline{b}_1, \underline{b}_2, \ldots, \underline{b}_m)$ and $\tilde{\underline{\boldsymbol{b}}}$ to denote $(\tilde{\underline{b}}_1, \tilde{\underline{b}}_2, \ldots, \tilde{\underline{b}}_m)$ in (3.72). We also used $\boldsymbol{B}$ to denote $(B_1, B_2, \ldots, B_m)$ in (3.74). Finally, we simplified the notation by

replacing $\tilde{\underline{\boldsymbol{b}}} \in \{0,1\}^{mn}$ by $\tilde{\underline{b}}$ in (3.73). Then

(3.71) follows for n sufficiently large and for $\underline{\boldsymbol{b}} \in \mathcal{B}^n_{SY,\varepsilon}(\boldsymbol{B})$ from $1/M_a \leq 2^{3n\varepsilon} p(\underline{\boldsymbol{b}})$, which can be shown in a similar way to (3.43),

(3.72) follows from summing over $\underline{\boldsymbol{b}} \in \{0,1\}^{mn}$ instead of over $\underline{\boldsymbol{b}}(m_a) \in \mathcal{B}^n_\varepsilon$, and over $\tilde{\underline{\boldsymbol{b}}} \in \{0,1\}^{mn}$ instead of over $\underline{\boldsymbol{b}}(k_a) \in \mathcal{B}^n_\varepsilon$ for $k_a \neq m_a$,

(3.73) is obtained by working out the summations over $\underline{b}_1, \underline{b}_2, \ldots, \underline{b}_m$, and $\underline{s}$,

(3.74) follows from $M_a = |\mathcal{B}^n_\varepsilon(\boldsymbol{B})| \leq 2^{n(H(\boldsymbol{B})+\varepsilon)}$, i.e., the $\mathcal{B}$-typicality property P_3, from (3.6), and from (3.11), and

(3.75) follows from (3.5) and (3.13).

The conclusion from (3.76) is that for

$$\begin{aligned} H(\boldsymbol{B}) &< H(\boldsymbol{B},S) - H(S|Y) - \left(\sum_{i=1}^{m} H(B_i|Y)\right) - (12+2m)\varepsilon \\ &= R_{\text{BMD}}(p(\boldsymbol{b},s)) - (12+2m)\varepsilon, \end{aligned}$$

the error probability of the second kind

$$\overline{P}_e(2) \leq \varepsilon \tag{3.77}$$

for n large enough. Using (3.70) and (3.77) in (3.25), we find that the error probability averaged over all sign-codewords in the sign-codebook, and averaged over all sign-codebooks $\overline{P}_e \leq 2\varepsilon$ for n large enough. This implies the existence of a sign-code with total error probability $P_e = \Pr\{\hat{M}_a \neq M_a\} \leq 2\varepsilon$. This holds for all $\varepsilon > 0$, and thus, the rate

$$R = H(\boldsymbol{B}) \leq R_{\text{BMD}} \tag{3.78}$$

is achievable with sign-coding and BMD, which concludes the proof of Theorem 3.3.

3.5.2.2 Proof of Theorem 3.4

For the error of first kind, we can write:

$$
\begin{aligned}
\overline{P}_e(1) &= \sum_{m_a} \frac{1}{M_a} \sum_{m_s} \frac{1}{2^{n_1}} \sum_{\underline{s}''} p(\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\quad \cdot \mathbb{1}\left[\bigcup_{i=1}^{m} ((\underline{b}_i(m_a), \underline{y}) \notin \mathcal{A}_\varepsilon^n) \bigcup ((\underline{s}'(m_s)\underline{s}'', \underline{y}) \notin \mathcal{A}_\varepsilon^n)\right] \\
&= \sum_{m_a} \frac{1}{M_a} \sum_{m_s} \sum_{\underline{s}''} 2^{-n} \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\quad \cdot \mathbb{1}\left[\bigcup_{i=1}^{m} ((\underline{b}_i(m_a), \underline{y}) \notin \mathcal{A}_\varepsilon^n) \bigcup ((\underline{s}'(m_s)\underline{s}'', \underline{y}) \notin \mathcal{A}_\varepsilon^n)\right] \qquad (3.79) \\
&\leq \sum_{m_a} \frac{1}{M_a} \sum_{m_s} \sum_{\underline{s}''} \sum_{\underline{y}} p(\underline{s}'(m_s)\underline{s}'', \underline{y}|\underline{\boldsymbol{b}}(m_a)) \\
&\qquad\qquad \cdot \mathbb{1}[(\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'', \underline{y}) \notin \mathcal{A}_\varepsilon^n] \qquad (3.80) \\
&= \sum_{m_a} \frac{1}{M_a} \Pr\{(\underline{\boldsymbol{b}}(m_a), \underline{S}, \underline{Y}) \notin \mathcal{A}_\varepsilon^n | \underline{\boldsymbol{B}} = \underline{\boldsymbol{b}}(m_a)\} \\
&\leq \sum_{m_a} \frac{\varepsilon}{M_a} \qquad (3.81) \\
&= \varepsilon. \qquad (3.82)
\end{aligned}
$$

Here, to obtain (3.79), we used the fact that $\underline{S}''$ is uniform, more precisely, $p(\underline{s}'') = 2^{-n_2}$. Then, we used $2^{-n} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'') = p(\underline{s}'(m_s)\underline{s}'', \underline{y}|\underline{\boldsymbol{b}}(m_a))$ in (3.80). Furthermore, (3.80) also follows from the fact that if at least one of $\underline{b}_1(m_a), \underline{b}_2(m_a), \ldots, \underline{b}_m(m_a)$ or $\underline{s}'(m_s)\underline{s}''$ is not jointly typical with $\underline{y}$, then $(\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'', \underline{y})$ is not jointly typical. Then, (3.81) is a direct consequence of Definition 3.1 since $\underline{\boldsymbol{b}}(m_a) \in \mathcal{B}^n_{SY,\varepsilon}(B_1 B_2 \cdots B_m)$ for $m_a = 1, 2, \cdots, M_a$.

For the error of second kind, we can write:

$$
\begin{aligned}
&\overline{P}_e(2) \\
&\leq \sum_{m_a} \frac{1}{M_a} \sum_{m_s} \frac{1}{2^{n_1}} \sum_{\underline{s}''} p(\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\qquad \cdot \sum_{(k_a,k_s)\neq(m_a,m_s)} \sum_{\tilde{\underline{s}}''} p(\tilde{\underline{s}}'') \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i(k_a), \underline{y}) \in \mathcal{A}_\varepsilon^n) \bigcap ((\underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}_\varepsilon^n)\right] \\
&= M_a 2^{n_1} \sum_{m_a,m_s,\underline{s}''} \frac{2^{-n}}{M_a} \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'') \qquad (3.83) \\
&\qquad \cdot \sum_{(k_a,k_s)\neq(m_a,m_s)} \sum_{\tilde{\underline{s}}''} \frac{2^{-n}}{M_a} \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i(k_a), \underline{y}) \in \mathcal{A}_\varepsilon^n) \bigcap ((\underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}_\varepsilon^n)\right] \\
&= M_a 2^{n_1} \sum_{m_a,m_s,\underline{s}''} \frac{2^{-n}}{M_a} \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\qquad \cdot \sum_{k_a\neq m_a,k_s,\tilde{\underline{s}}''} \frac{2^{-n}}{M_a} \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i(k_a), \underline{y}) \in \mathcal{A}_\varepsilon^n) \bigcap ((\underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}_\varepsilon^n)\right] \\
&\quad + 2^{n_1} \sum_{m_a,m_s,\underline{s}''} \frac{2^{-n}}{M_a} \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'') \qquad (3.84) \\
&\qquad \cdot \sum_{k_s\neq m_s,\tilde{\underline{s}}''} 2^{-n} \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i(m_a), \underline{y}) \in \mathcal{A}_\varepsilon^n) \bigcap ((\underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}_\varepsilon^n)\right], \\
&\leq M_a 2^{n_1} 2^{6n\varepsilon} \sum_{m_a,m_s,\underline{s}''} p(\underline{\boldsymbol{b}}(m_a)) p(\underline{s}'(m_s)\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\quad \cdot \sum_{k_a\neq m_a,k_s,\tilde{\underline{s}}''} p(\underline{\boldsymbol{b}}(k_a)) p(\underline{s}'(k_s)\tilde{\underline{s}}'') \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i(k_a), \underline{y}) \in \mathcal{A}_\varepsilon^n) \bigcap ((\underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}_\varepsilon^n)\right] \\
&\quad + 2^{n_1} 2^{3n\varepsilon} \sum_{m_a,m_s,\underline{s}''} p(\underline{\boldsymbol{b}}(m_a)) p(\underline{s}'(m_s)\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{\boldsymbol{b}}(m_a), \underline{s}'(m_s)\underline{s}'') \\
&\qquad \cdot \sum_{k_s\neq m_s,\tilde{\underline{s}}''} p(\underline{s}'(k_s)\tilde{\underline{s}}'') \mathbb{1}\left[\bigcap_{i=1}^{m} ((\underline{b}_i(m_a), \underline{y}) \in \mathcal{A}_\varepsilon^n) \bigcap ((\underline{s}'(k_s)\tilde{\underline{s}}'', \underline{y}) \in \mathcal{A}_\varepsilon^n)\right] \qquad (3.85)
\end{aligned}
$$

where (3.83) follows from $n = n_1 + n_2$ and from the fact that $\underline{S}''$ is uniform, more precisely, $p(\underline{s}'') = 2^{-n_2}$. Then, (3.84) is obtained by splitting $(k_a, k_s) \neq (m_s, m_s)$ into $\{k_a \neq m_s, k_s\}$ and $\{k_a = m_a, k_s \neq m_s\}$. Next, (3.85) follows for n sufficiently large and for $\underline{\boldsymbol{b}} \in \mathcal{B}^n_{SY,\varepsilon}(\boldsymbol{B})$ from $1/M_a \leq 2^{3n\varepsilon} p(\underline{\boldsymbol{b}})$ and from $p(\underline{s}'\underline{s}'') = 2^{-n}$.

From (3.85), we obtain:

$$
\begin{aligned}
&\overline{P}_e(2) \\
&\quad\leq\; M_a 2^{n_1} 2^{6n\varepsilon} \sum_{\underline{b},\underline{s}'\underline{s}''} p(\underline{b})p(\underline{s}'\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{b},\underline{s}'\underline{s}'') \sum_{\tilde{\underline{b}},\tilde{\underline{s}}'\tilde{\underline{s}}''} p(\tilde{\underline{b}})p(\tilde{\underline{s}}'\tilde{\underline{s}}'') \\
&\qquad\qquad \cdot\mathbb{1}\left[\bigcap_{i=1}^{m}((\tilde{\underline{b}}_i,\underline{y})\in\mathcal{A}_\varepsilon^n)\bigcap((\tilde{\underline{s}}'\tilde{\underline{s}}'',\underline{y})\in\mathcal{A}_\varepsilon^n)\right] \\
&\qquad + 2^{n_1}2^{3n\varepsilon} \sum_{\underline{b},\underline{s}'\underline{s}''} p(\underline{b})p(\underline{s}'\underline{s}'') \sum_{\underline{y}} p(\underline{y}|\underline{b},\underline{s}'\underline{s}'') \sum_{\tilde{\underline{s}}'\tilde{\underline{s}}''} p(\tilde{\underline{s}}'\tilde{\underline{s}}'') \\
&\qquad\qquad \cdot\mathbb{1}\left[\bigcap_{i=1}^{m}((\underline{b}_i,\underline{y})\in\mathcal{A}_\varepsilon^n)\bigcap((\tilde{\underline{s}}'\tilde{\underline{s}}'',\underline{y})\in\mathcal{A}_\varepsilon^n)\right] \qquad (3.86) \\
&\quad=\; M_a 2^{n_1} 2^{6n\varepsilon} \sum_{\underline{b},\underline{s}} p(\underline{b})p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{b},\underline{s}) \sum_{\tilde{\underline{b}},\tilde{\underline{s}}} p(\tilde{\underline{b}})p(\tilde{\underline{s}}) \\
&\qquad\qquad \cdot\mathbb{1}\left[\bigcap_{i=1}^{m}((\tilde{\underline{b}}_i,\underline{y})\in\mathcal{A}_\varepsilon^n)\bigcap((\tilde{\underline{s}},\underline{y})\in\mathcal{A}_\varepsilon^n)\right] \\
&\qquad + 2^{n_1}2^{3n\varepsilon} \sum_{\underline{b},\underline{s}} p(\underline{b})p(\underline{s}) \sum_{\underline{y}} p(\underline{y}|\underline{b},\underline{s}) \sum_{\tilde{\underline{s}}} p(\tilde{\underline{s}}) \\
&\qquad\qquad \cdot\mathbb{1}\left[\bigcap_{i=1}^{m}((\underline{b}_i,\underline{y})\in\mathcal{A}_\varepsilon^n)\bigcap((\tilde{\underline{s}},\underline{y})\in\mathcal{A}_\varepsilon^n)\right], \qquad (3.87) \\
&\quad=\; M_a 2^{n_1} 2^{6n\varepsilon} \sum_{\underline{y}} p(\underline{y}) \sum_{\tilde{\underline{b}},\tilde{\underline{s}}} p(\tilde{\underline{b}},\tilde{\underline{s}})\mathbb{1}\left[\bigcap_{i=1}^{m}((\tilde{\underline{b}}_i,\underline{y})\in\mathcal{A}_\varepsilon^n)\bigcap((\tilde{\underline{s}},\underline{y})\in\mathcal{A}_\varepsilon^n)\right] \\
&\qquad + 2^{n_1}2^{3n\varepsilon} \sum_{\underline{b},\underline{y}} p(\underline{b},\underline{y}) \sum_{\tilde{\underline{s}}} p(\tilde{\underline{s}})\mathbb{1}\left[\bigcap_{i=1}^{m}((\underline{b}_i,\underline{y})\in\mathcal{A}_\varepsilon^n)\bigcap((\tilde{\underline{s}},\underline{y})\in\mathcal{A}_\varepsilon^n)\right] \qquad (3.88)
\end{aligned}
$$

where

(3.86) follows from summing over $\underline{b}\in\{0,1\}^{mn}$ instead of over $\underline{b}(m_a)\in\mathcal{B}_\varepsilon^n$ and over $\tilde{\underline{b}}\in\{0,1\}^{mn}$ instead of $\underline{b}(k_a)\in\mathcal{B}_\varepsilon^n$ for $k_a\neq m_a$. Moreover, it follows from summing over $\underline{s}'\in\mathcal{S}^{n_1}$ instead of $\underline{s}'(k_s)$ for $k_s=1,2,\cdots,M_s$ and $k_s\neq m_s$,

(3.87) follows from substituting $\underline{s}$ for $\underline{s}'\underline{s}''$ and $\tilde{\underline{s}}$ for $\tilde{\underline{s}}'\tilde{\underline{s}}''$, and

(3.88) is obtained by working out the summations over $\underline{b}_1,\underline{b}_2,\ldots,\underline{b}_m,\underline{s}$ in the first part and $\underline{s}$ in the second part.

Finally, from (3.88), we obtain:

$$\begin{aligned}
\overline{P}_e(2) \le\ & 2^{n(H(\boldsymbol{B})+\varepsilon)}2^{n\gamma}2^{6n\varepsilon}|\mathcal{A}_\varepsilon^n(Y)|2^{-n(H(Y)-\varepsilon)} \\
& \cdot\left(\prod_{i=1}^m |\mathcal{A}_\varepsilon^n(B_i|\underline{y})|\right)|\mathcal{A}_\varepsilon^n(S|\underline{y})|2^{-n(H(B_1B_2\cdots B_mS)-\varepsilon)} \\
& + 2^{n\gamma}2^{3n\varepsilon}|\mathcal{A}_\varepsilon^n(Y)|2^{-n(H(\boldsymbol{B}Y)-\varepsilon)}2^{-n(H(S)-\varepsilon)} \\
& \cdot\left(\prod_{i=1}^m |\mathcal{A}_\varepsilon^n(B_i|\underline{y})|\right)|\mathcal{A}_\varepsilon^n(S|\underline{y})| \qquad (3.89) \\
\le\ & 2^{n(H(\boldsymbol{B})+\varepsilon)}2^{n\gamma}2^{6n\varepsilon}2^{n(H(Y)+\varepsilon)}2^{-n(H(Y)-\varepsilon)} \\
& \cdot\left(\prod_{i=1}^m 2^{n(H(B_i|Y)+2\varepsilon)}\right)2^{n(H(S|Y)+2\varepsilon)}2^{-n(H(\boldsymbol{B}S)-\varepsilon)} \\
& + 2^{n\gamma}2^{3n\varepsilon}2^{n(H(Y)+\varepsilon)}2^{-n(H(\boldsymbol{B}Y)-\varepsilon)}2^{-n(H(S)-\varepsilon)} \\
& \cdot\left(\prod_{i=1}^m 2^{n(H(B_i|Y)+2\varepsilon)}\right)2^{n(H(S|Y)+2\varepsilon)} \qquad (3.90) \\
=\ & 2^{n\left(H(\boldsymbol{B})+\gamma+\left(\sum_{i=1}^m H(B_i|Y)\right)+H(S|Y)-H(\boldsymbol{B}S)+(12+2m)\varepsilon\right)} \\
& + 2^{n\left(\gamma+H(Y)-H(\boldsymbol{B}Y)-H(S)+\left(\sum_{i=1}^m H(B_i|Y)\right)+H(S|Y)+(8+2m)\varepsilon\right)}. \qquad (3.91)
\end{aligned}$$

Here, we substituted $n_1 = n\gamma$ in (3.89). Then

(3.89) is obtained using for the first part that $M_a = |\mathcal{B}_\varepsilon^n(\boldsymbol{B})| \le 2^{n(H(\boldsymbol{B})+\varepsilon)}$, i.e., the $\mathcal{B}$-typicality property P_3, (3.6) for $p(\underline{y})$, and (3.11) for $p(\underline{\tilde{\boldsymbol{b}}}, \underline{\tilde{s}})$. For the second part, we used (3.6) for $p(\underline{\tilde{s}})$ and (3.11) for $p(\underline{\boldsymbol{b}}, \underline{y})$, and

(3.90) follows from (3.5) and (3.13).

The conclusion from (3.91) is that for n large enough, for

$$H(\boldsymbol{B}) + \gamma \le R_{\text{BMD}} - (12+2m)\varepsilon \qquad (3.92)$$

and for

$$\gamma \le H(\boldsymbol{B}Y) + H(S) - H(Y) - \left(\sum_{i=1}^m H(B_i|Y)\right) - H(S|Y) - (8+2m)\varepsilon, \qquad (3.93)$$

the error probability of the second kind

$$\overline{P}_e(2) \le \varepsilon. \qquad (3.94)$$

The second constraint (3.93) is already implied by the first constraint (3.92) since

$$\begin{aligned}\gamma &\le H(\boldsymbol{B}Y) + H(S) - H(Y) - \left(\sum_{i=1}^{m} H(B_i|Y)\right) - H(S|Y) - (8+2m)\varepsilon \\ &= H(\boldsymbol{B}Y) + H(S) - H(Y) - \left(\sum_{i=1}^{m} H(B_i|Y)\right) - H(S|Y) \\ &\quad + H(\boldsymbol{B}S) - H(\boldsymbol{B}S) - (8+2m)\varepsilon \\ &= H(\boldsymbol{B}Y) + H(S) - H(Y) + R_{\text{BMD}} - H(\boldsymbol{B}) - H(S) - (8+2m)\varepsilon \\ &= H(\boldsymbol{B}|Y) + R_{\text{BMD}} - H(\boldsymbol{B}) - (8+2m)\varepsilon. \end{aligned} \tag{3.95}$$

Using (3.82) and (3.94) in (3.25), we find that the error probability averaged over all sign-codewords in the modified sign-codebook, and averaged over all modified sign-codebooks $\overline{P}_e \le 2\varepsilon$ for n large enough. This implies the existence of a modified sign-code with total error probability $P_e = \Pr\{(\hat{M}_a, \hat{M}_s) \neq (M_a, M_s)\} \le 2\varepsilon$. This holds for all $\varepsilon > 0$, and thus, the rate

$$R = H(\boldsymbol{B}) + \gamma \le R_{\text{BMD}} \tag{3.96}$$

is achievable with modified sign-coding, which concludes the proof of Theorem 3.4.

3.6 Wachsmann Curves: Parameter Selection for PAS

In the previous sections of this chapter, we introduced random sign-coding arguments and showed that sign-coding, and hence PAS, achieve the capacity of any memoryless channel, given that the capacity-achieving distribution is symmetric. In this section, we study the optimum shaping and FEC coding rates for PAS using AIRs. Therefore, we consider the case where $N \to \infty$, which implies (for asymptotically optimum amplitude shaping architectures that will be discussed in Ch. 4) that the input blocklength of the shaper $k = NH(A)$, and consequently, $R_{\text{t}} = H(A) + \gamma$.

3.6.1 Shaping and Coding Redundancy

In the PAS architecture, to obtain a target rate $R_{\text{t}} = H(A) + \gamma$ using the 2^m-ASK constellation, a total of $N(m - R_{\text{t}})$ redundant bits are added to a channel input sequence by shaping and coding operations combined. Shaping is responsible for $N(m - 1 - H(A))$ redundant bits, whereas coding adds $N(H(A) + 1 - R_{\text{t}}) = N(1 - \gamma)$. This is illustrated in Fig. 3.6 where the content of a channel input sequence that is produced by the PAS architecture of Fig. 3.1 (top) is shown. The striped areas represent the information carried in signs (red) which is γN bits, and in amplitudes (blue) which is $k = NH(A)$ bits. Dotted areas show the redundant bits in a sequence. When $\gamma = 0$, i.e., $R_{\text{c}} = (m-1)/m$, all signs are selected by

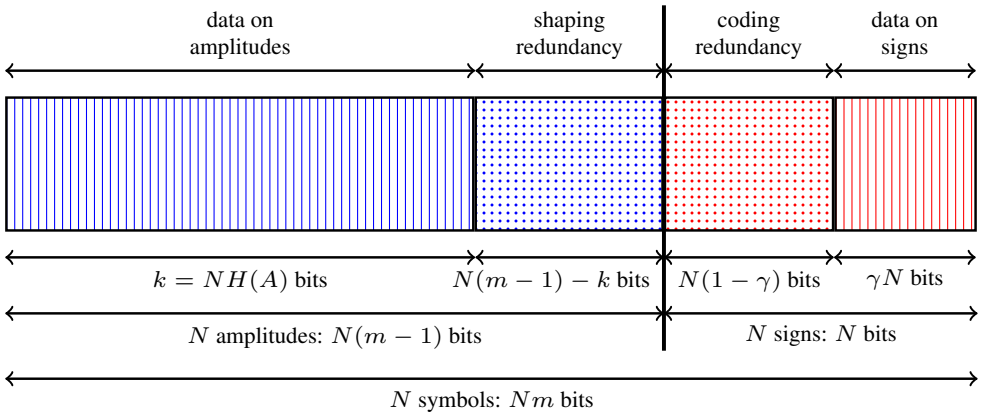

Figure 3.6: Content of a channel input sequence produced by PAS.

parity bits and thus, the striped red area in Fig. 3.6 vanishes. When $H(A) = m - 1$, the amplitudes are uniformly distributed, i.e., there is no shaping, and thus, the dotted blue area in Fig. 3.6 disappears. We note that a similar illustration was provided for a single ASK symbol in [93, Fig. 9]. In Table 3.2, the content of a sequence at the output of a PAS transmitter (in accordance with Fig. 3.6) is tabulated for Example 2.1 where $N = 216$.

Table 3.2: Content of an amplitude sequence as in Fig. 3.6 based on Example 2.1

Parameter	Formula (per N-sequence)	Value per 1-D (Example 2.1)	Value per 216-D (Example 2.1)
Data on amp.	$NH(A)$	1.75	378
Data on sign	$N\gamma$	0.50	108
Shap. redundancy	$N(m-1-H(A))$	0.25	54
Cod. redundancy	$N(1-\gamma)$	0.50	108
Redundancy	$N(m-R_{\mathrm{t}})$	0.75	162
Data, NR_{t}	$N(H(A)+\gamma)$	2.25	486

3.6.2 Gap-to-capacity

When the input is constrained to be MB-distributed, $H(X) = H(A) + 1$ can be used as a design parameter which tunes the balance between shaping and coding redundancies at a fixed rate R_{t}. More specifically, the entropy $H(A)$ of the MB distribution is controlled by λ as in (2.24). Thus by changing λ, the amount of shaping redundancy in an amplitude can be adjusted. The question is then how to choose the optimum λ. Following Wachsmann, Fischer

and Huber [27, 94], we use the gap-to-capacity, i.e., normalized SNR,

$$\begin{aligned}\Delta\text{SNR} &= 10\log\left(\text{SNR}\right)\Big|_{R_{\text{BMD}}=R_{\text{t}}} - 10\log\left(\text{SNR}\right)\Big|_{C=R_{\text{t}}} \\ &= 10\log\left(\frac{\text{required SNR such that } R_{\text{BMD}}=R_{\text{t}}}{2^{2R_{\text{t}}}-1}\right)\end{aligned} \tag{3.97}$$

as the metric to be minimized when searching for the optimum MB distribution for a fixed transmission rate and constellation size.[4] The only difference with respect to [27, eq. (55)] is that we use R_{BMD} instead of MI in (3.97). The numerator in (3.97) is the SNR value at which $R_{\text{BMD}} = R_{\text{t}}$ for a given $p(x)$, and the denominator is the SNR value at which the capacity $C = R_{\text{t}}$. Observing that $\gamma = R_{\text{t}} - H(A)$, the rate of the FEC code that should be employed in PAS to obtain a transmission rate R_{t} for a given constellation entropy $H(X)$ is

$$\begin{aligned}R_{\text{c}} &= \frac{m-1+\gamma}{m} \\ &= \frac{m-1+R_{\text{t}}-H(A)}{m} \\ &= \frac{m+R_{\text{t}}-H(X)}{m}.\end{aligned} \tag{3.98}$$

Example 3.2 (Optimal PAS parameters: The AWGN Channel). In Fig. 3.7, the entropy $H(X)$ of an MB-distributed 8-ASK input X vs. ΔSNR is plotted for $R_{\text{t}} = 2.25$ bit/1-D. On the top horizontal axis, the corresponding FEC code rates in (3.98) are also shown. The rightmost point (indicated by a square) corresponds to uniform signaling where the target rate of 2.25 bit/1-D is obtained by using an FEC code of rate $R_{\text{c}} = R_{\text{t}}/m = 3/4$. In this trivial case, all 0.75 bits of redundancy are added by the coding operation, and the gap-to-capacity ΔSNR is 1.04 dB. The leftmost part of the curve where $H(X)$ goes to R_{t} belongs to the uncoded signaling case, i.e., $R_{\text{c}} = 1$, where R_{t} is attained by shaping the constellation such that $H(X) = R$. Here ΔSNR is infinite, since without coding, reliable communication is only possible over a noiseless channel. The minimum ΔSNR in Fig. 3.7 is obtained with $H(X) = 2.745$, which corresponds to $R_{\text{c}} = 0.835$ from (3.98). In DVB-S2 [77] and IEEE 802.11 [16], the code rate that is closest to 0.835 is $5/6 \approx 0.833$. Accordingly, the best performance is expected to be provided by the FEC rate $5/6$, with an SNR gain over uniform signaling that amounts according to this analysis to approximately 0.82 dB.[5]

To provide more insight into how the gap-to-capacity curve is plotted in Fig. 3.7, we now consider the distribution $p(x)$ that leads to the minimum (circle marker) in Fig. 3.7

[4] In general, gap-to-capacity can be computed for any parametric family of distributions. Here we only consider the MB distributions since they have been shown to perform very close to the capacity of ASK constellations over the AWGN channel [23] and maximize the energy efficiency [47].

[5] We note here that it is not always possible to have an FEC code with the desired rate, especially in cases where existing codes are reused. In such cases, the FEC code rate which is closest to the optimum value should be employed. However, we argue that the loss due to this suboptimal parameter selection will be negligible since ΔSNR differs only slightly around the minimum in Fig. 3.7.

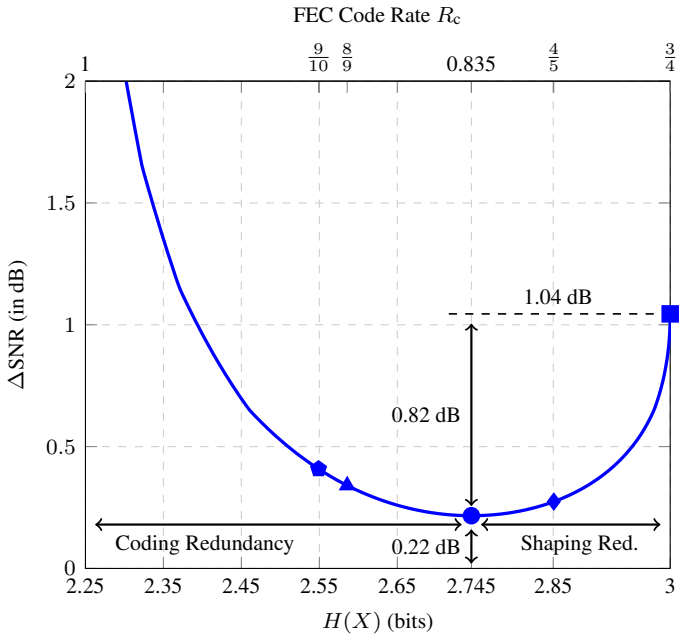

Figure 3.7: Channel input entropy vs. gap-to-capacity for 8-ASK at the target rate of $R_\mathrm{t} = 2.25$ bit/1-D. The x-axis above shows the corresponding FEC code rates.

where $H(X) = 2.745$ bits. In Fig. 3.8, R_BMD is plotted vs. SNR for this distribution, along with the curve that corresponds to uniform signaling. The (blue) shaped curve converges to $R_\mathrm{BMD} = H(X) = 2.745$ bit/1-D for asymptotically large SNR, while the (red) uniform curve converges to $R_\mathrm{BMD} = m = 3$ bit/1-D. The 0.255 bit/1-D difference between these values is the shaping redundancy. The difference $H(X) - R_\mathrm{t} = 2.745 - 2.25 = 0.495$ bit/1-D is the coding redundancy. The difference in SNR required to obtain $R_\mathrm{BMD} = R_\mathrm{t} = 2.25$ bit/1-D for uniform and shaped signaling is 0.82 dB. The remaining gap-to-capacity for shaped signaling is 0.22 dB, which corresponds to the ΔSNR value at $H(X) = 2.745$ bit/1-D (circle marker) in Fig. 3.7. All these parameters or values are indicated both in Fig. 3.7 and Fig. 3.8.

Example 3.3 (Optimal PAS parameters: Fading Channels). In Fig. 3.9, gap-to-capacity is plotted for 8-ASK at $R_\mathrm{t} = 1.5$ bit/1-D for the AWGN channel, the Rician fading channel with $K = 10$, and the Rayleigh fading channel, see Sec. 2.2.2. Again, on the top horizontal axis, the corresponding FEC code rates in (3.98) are shown. We see that as the channel becomes more and more dynamic, i.e., changes first to Rician and then to Rayleigh, (1) the optimum point shifts towards uniform signaling, and the required coding redundancy increases, and (2) the maximum capacity gain decreases. For instance, in the Rayleigh fading case, the

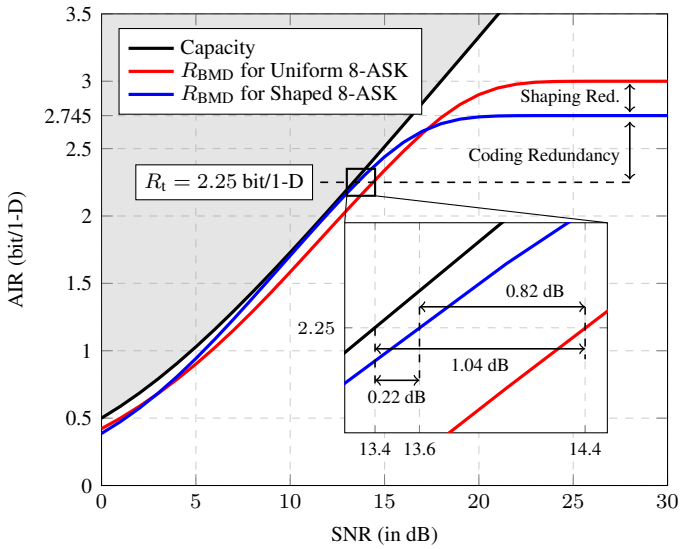

Figure 3.8: SNR vs. R_{BMD} for the MB-distributed 8-ASK with $H(X) = 2.745$ bit/1-D, and for uniform 8-ASK.

optimum point is around $H(X) = 2.65$ and thus, the optimum FEC code rate is $R_{\mathrm{c}} = 0.62$. Whereas the optimum code rate is around 0.75 for the AWGN channel. The corresponding capacity gain is 0.56 dB for the Rayleigh fading channel, which is 0.41 dB smaller than that of the AWGN channel. Therefore, we conclude that although the gains are smaller, shaping increases the maximum AIR over fading channels as well. Besides, in such cases, the total redundancy should be distributed more in favor of coding and less in favor of shaping.

3.7 Conclusion

In this chapter, we searched for an answer to the following research question.

RQ-1 What are the AIRs of PAS for symbol-metric decoding (SMD) and bit-metric decoding (BMD)? Is it possible to achieve the capacity of memoryless channels with PAS? What are the optimum shaping and coding rates in PAS that maximize AIR gains?

We developed the random sign-coding framework to compute achievable rates of PAS. We demonstrated that it is possible to obtain achievability results for all PAS settings in this framework. We showed that PAS achieves the capacity of memoryless channels with symmetric capacity-achieving distributions. Furthermore, unlike most proofs of Shannon's channel

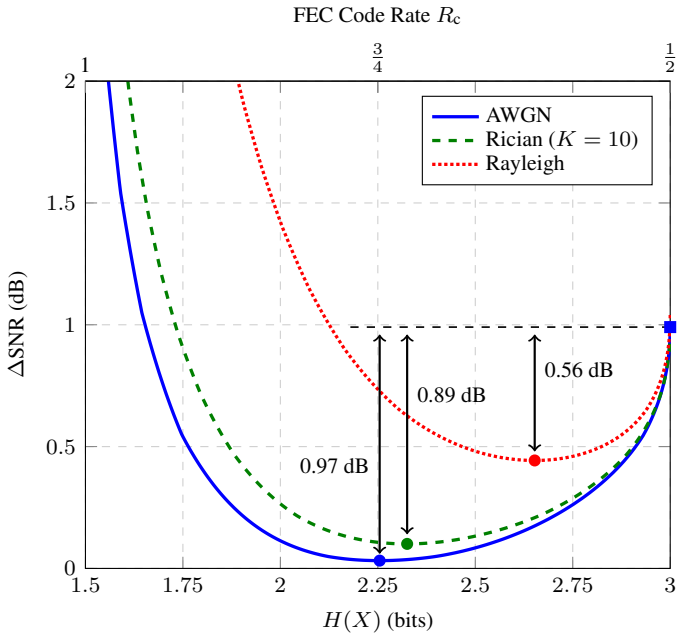

Figure 3.9: Channel input entropy vs. gap-to-capacity for 8-ASK at the target rate of $R_t = 1.5$ bit/1-D. The x-axis above shows the corresponding FEC code rates. Filled circles specify the minima of their corresponding curves.

coding theorem, most of the code that is used to prove achievability is generated constructively in our random sign-coding experiment. Thus, random sign-coding proofs can at least partially be used to identify "good" codes. We also showed that capacity can be achieved using binary linear codes in PAS. Finally, we used gap-to-capacity plots, i.e., Wachsmann curves [27], to obtain the optimum combination of shaping and coding rates in PAS. We observed that this optimum shifts in favor of increased coding redundancy for fading channels.

CHAPTER 4

Amplitude Shaping for Short Blocklengths

Parts of this chapter are published in:

Y. C. Gültekin and F. M. J. Willems, "Comparison of enumerative and probabilistic shaping for short block lengths," *Eur. School of Inf. Theory (ESIT)*, Madrid, Spain, May 2017. (Abstract submission & poster presentation)

Y. C. Gültekin, W. J. van Houtum, and F. M. J. Willems, "On constellation shaping for short block lengths," in *Proc. Symp. on Inf. Theory and Signal Process. in the Benelux (SITB)*, Enschede, The Netherlands, June 2018, pp. 86-96.

Y. C. Gültekin, T. Fehenberger, A. Alvarado, and F. M. J. Willems, "Probabilistic shaping for finite blocklengths: Distribution matching and sphere shaping," *Entropy*, vol. 19, no. 5: 581, May 2020.

4.1 Introduction

In Chapter 3, we derived achievable rates for PAS by using random sign-coding arguments, and we showed that PAS achieves capacity. For our proofs, we assumed that the shaping layer produces i.i.d. amplitudes, and the signaling blocklength $N \to \infty$. In this chapter, we will search for an answer to the research question **RQ-2**: What is the "best" amplitude shaping strategy for finite blocklengths? We will focus on the shaping layer of PAS, and we will investigate the performance of several amplitude shaping methods for practical values of N, which are often very small. For this purpose, we will define two performance criteria: rate loss and shaping gain.

As introduced in Sec. 2.6.3, the function of an amplitude shaper is to map k-bit strings to N-amplitude sequences in an invertible manner. This can be accomplished by first specifying a set of amplitude sequences, which we call the *shaping set*. Then an algorithm that maps binary strings to these sequences assuming a specific way of ordering, which is called the *shaping algorithm*, must be defined. In this chapter, we restrict our attention to the selection of shaping sets. Shaping algorithms and their complexity will be discussed in Chapter 7.

In PAS, channel inputs are created by adding signs to the amplitudes outputted by the shaper. These signs are either produced by the channel code in the form of parity bits, or they are information bits. In both cases, we assume that the signs are uniform. Consequently, the properties of the amplitude sequences, such as their symbol distribution or average energy, determine the performance of PAS. A careful selection of the set of amplitude sequences that can be outputted by the shaper, i.e., the shaping set or shaping code, may result in improvement in overall performance. The shaping set can be constructed with the aim of matching a target distribution (direct method) or obtaining an energy-efficient signal space (indirect method). We will consider both methods in this chapter. The direct method, we refer to as constant composition coding. The indirect method, we refer to as sphere coding. In the remainder of this section, we will define the parameters and metrics that will be used throughout this thesis to assess the performance of amplitude shaping architectures.

4.1.1 Fundamental Parameters & Performance Metrics

Let $\mathcal{A}_\mathrm{s}$ be a shaping code which consists of amplitude sequences $a_j^N = (a_{j,1}, a_{j,2}, \cdots, a_{j,N})$ where $a_{j,n} \in \mathcal{A}$ for $n = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, J$. Here $J = |\mathcal{A}_\mathrm{s}|$ is the number of codewords, i.e., sequences of length N, in the code. We assume that all codewords occur uniformly, i.e., each with probability $1/J$. Therefore, the shaping rate of the code $\mathcal{A}_\mathrm{s}$ is

$$R_\mathrm{s} = \frac{\log_2 J}{N} \tag{4.1}$$

in bit/1-D.

The energy of a codeword a^N is defined as

$$e\left(a^N\right) = \sum_{n=1}^{N} a_n^2. \tag{4.2}$$

The energy E_{av} averaged over all codewords in $\mathcal{A}_{\text{s}}$ and over all time indices n is

$$E_{\text{av}} = \frac{1}{NJ} \sum_{j=1}^{J} e\left(a_j^N\right) = \frac{1}{J} \sum_{j=1}^{J} \frac{1}{N} \sum_{n=1}^{N} a_{j,n}^2. \tag{4.3}$$

The shaping gain of the code $\mathcal{A}_{\text{s}}$ over uniform signaling is defined as

$$G_{\text{s}} = 10 \log_{10} \left(\frac{2^{2(R_{\text{s}}+1)} - 1}{3E_{\text{av}}} \right) \tag{4.4}$$

in dB. Similar to [56, Sec. II-A], we assumed in (4.4) that the average symbol energy expression $(2^{2m} - 1)/3$ for uniform 2^m-ASK constellations works as a good approximation for noninteger m. The shaping rate R_{s} is increased by one in (4.4) to account for the signed combinations of codewords.

The composition of a codeword a^N is defined as $C = [n_1, n_2, \cdots, n_{|\mathcal{A}|}]$, where n_i denotes the number of times the i^{th} element of $\mathcal{A}$ occurs in a^N, i.e.,

$$n_i = \sum_{n=1}^{N} \mathbb{1}[a_n = \mathcal{A}(i)] \tag{4.5}$$

for $i = 1, 2, \cdots, |\mathcal{A}|$ where obviously, $\sum_i n_i = N$. Then the empirical distribution, i.e., the type, of a^N is given by

$$p_{a^N}\left(\mathcal{A}(i)\right) = \frac{n_i}{N} \tag{4.6}$$

for $i = 1, 2, \cdots, |\mathcal{A}|$. The number of codewords with the same composition C is given by the multinomial coefficient (MC)

$$\text{MC}(C) = \frac{N!}{\prod_{i=1}^{|\mathcal{A}|} n_i!}. \tag{4.7}$$

The marginal distribution $p_n(a)$ averaged over all codewords in $\mathcal{A}_{\text{s}}$ and over the time index n is defined as

$$p_n(a) = \frac{1}{J} \sum_{j=1}^{J} \mathbb{1}\left[a_{j,n} = a\right], \tag{4.8}$$

for $n = 1, 2, \cdots, N$ and for $a \in \mathcal{A}$. The distribution $p(a)$ averaged over all codewords in $\mathcal{A}_\mathrm{s}$ and over all time indices n is defined as

$$\begin{aligned} p(a) &= \frac{1}{J}\sum_{j=1}^{J}\frac{1}{N}\sum_{n=1}^{N}\mathbb{1}[a_{j,n} = a] \\ &= \frac{1}{N}\sum_{n=1}^{N} p_n(a) \end{aligned} \tag{4.9}$$

for $a \in \mathcal{A}$. We note that in general, $p_n(a) \neq p_l(a)$ for $n \neq l$.

Consider a shaping code $\mathcal{A}_\mathrm{s}$ with rate R_s, average distribution $p(a)$, and average energy E_av. It can be shown for a random variable A with distribution $p(a)$ and for an MB-distributed random variable A_MB with $\mathbb{E}[|A_\mathrm{MB}|^2] = E_\mathrm{av}$ that

$$H(A_\mathrm{MB}) \overset{\text{(a)}}{\geq} H(A) \overset{\text{(b)}}{\geq} R_\mathrm{s} \tag{4.10}$$

where (a) is due to the fact that the MB distribution maximizes entropy for a fixed average energy [26, Ch. 12], and (b) is due to the finite blocklength N. The inequalities in (4.10) will be discussed in detail in Sec. 4.4.1, and they show that the finite length rate loss

$$R_\mathrm{loss} = H(A_\mathrm{MB}) - R_\mathrm{s} \tag{4.11}$$

is always nonnegative for finite N.

Consider again a shaping code $\mathcal{A}_\mathrm{s}$ with rate R_s and average energy E_av. Furthermore, consider an MB-distributed random variable A_MB, now with $H(A_\mathrm{MB}) = R_\mathrm{s}$. In [95], Cho defined the energy gap

$$E_\mathrm{gap} = \frac{E_\mathrm{av}}{\mathbb{E}\left[A_\mathrm{MB}^2\right]} \tag{4.12}$$

to evaluate the performance of variable length shaping codes that he proposed.

Definition 4.1 (Optimum shaping codes). A shaping code $\mathcal{A}_\mathrm{s}$ is called asymptotically optimum if

$$R_\mathrm{loss} \to 0 \text{ for } N \to \infty, \tag{4.13}$$

or equivalently, if

$$E_\mathrm{gap} \to 1 \text{ for } N \to \infty. \tag{4.14}$$

As we discussed in Sec. 2.5.1, the slope of the SNR (in dB) versus capacity curve for the AWGN channel is approximately 6 dB/bit. Thus for optimum shaping codes, E_gap and R_loss can be related as $10\log_{10} E_\mathrm{gap} \approx 6R_\mathrm{loss}$.

4.2 Constant Composition Distribution Matching

Definition 4.2 (Constant composition codes). A code is a constant composition code if for a given composition C, all codewords with the same composition C are in the code, and no other codewords.

For the initial proposal of PAS, constant composition distribution matching (CCDM) was used as the amplitude shaping architecture [28]. CCDM is a direct method where the aim is to mimic a target distribution $p(a^*)$. The basic principle of CCDM is to utilize amplitude sequences having a fixed empirical distribution, which is information-theoretically close to the target distribution. To this end, first a target distribution $p(a^*)$ is obtained. This is usually accomplished by finding the MB distribution that maximizes the AIR at a given SNR. Then this distribution is quantized to obtain $p(a)$ such that $Np(a)$ is an integer for $a \in \mathcal{A}$. The distributions of this form are called N-type. This quantization can be accomplished either by a simple rounding or by minimizing the Kullback–Leibler (KL) divergence between $p(a^*)$ and $p(a)$ as in [96]. Finally, the constant composition code with $C = Np(a)$ is considered as the shaping code. We denote this constant composition code by $\mathcal{A}_{\mathrm{s}}^{\mathrm{cc}}$ where its cardinality is $|\mathcal{A}_{\mathrm{s}}^{\mathrm{cc}}| = \mathrm{MC}(C)$ as in (4.7). The set $\mathcal{A}_{\mathrm{s}}^{\mathrm{cc}}$ is called the type class of $p(a)$. It is shown in [28] that as $N \to \infty$, the KL divergence between $p(a^*)$ and $p(a)$ vanishes. Consequently, constant composition codes are optimum according to Definition 4.1, i.e., both (a) and (b) in (4.10) are satisfied with equality when $N \to \infty$. We note that due to symmetry with respect to time indices n, p_n is the same for $n = 1, 2, \cdots, N$ for constant composition codes.

Example 4.1 (CCDM). Consider a target shaping rate of 1.75 bit/1-D with $\mathcal{A} = \{1, 3, 5, 7\}$. We use the target MB distribution $p(a^*) = [0.3918, 0.3117, 0.1972, 0.0993]$ with $H(A^*) = 1.8466$ at $N = 96$. The composition that is obtained with the quantization rule proposed in [96, Algorithm 2] is $C = [37, 30, 19, 10]$. The rate and the average energy of the corresponding constant composition code are $R_{\mathrm{s}} = 1.7575$ bit/1-D and $E_{\mathrm{av}} = 13.25$, respectively. The rate loss is $R_{\mathrm{loss}} = 0.0955$ bit/1-D. At $N = 216$, we use the target MB distribution $p(a^*) = [0.4140, 0.3169, 0.1857, 0.0833]$ with $H(A^*) = 1.8019$. The composition that is obtained with the quantization rule proposed in [96, Algorithm 2] is $C = [89, 69, 40, 18]$. The rate and the average energy of the corresponding constant composition code are $R_{\mathrm{s}} = 1.7507$ and $E_{\mathrm{av}} = 12.00$, respectively. The rate loss is $R_{\mathrm{loss}} = 0.0516$ bit/1-D. We observe that at this target rate, the rate loss is nearly halved when N is increased from 96 to 216, and the average energy is reduced by 0.43 dB.

4.3 Sphere Shaping

Definition 4.3 (Sphere codes). A code is a sphere code if there exist no sequences, not in the code, with energy smaller than the energy of a codeword. More precisely, we define the

sphere code as

$$\mathcal{A}^{\bullet} = \left\{ a^N = (a_1, a_2, \cdots, a_N) \middle| \sum_{n=1}^{N} |a_n|^2 \leq E^{\bullet} \right\} \tag{4.15}$$

for $a \in \mathcal{A}$ where $E^{\bullet}$ is the maximum *sequence* energy.

Sphere shaping where a sphere code is considered as the shaping code $\mathcal{A}_{\rm s}$ is an indirect method where the aim is to construct the codebook with the least average energy for a given shaping rate $R_{\rm s}$. We will show in Sec. 4.4 that the sphere codes are also optimum according to Definition 4.1. This hints that $p(a)$ approaches an MB distribution asymptotically for large N for sphere codes, which will also be shown in Sec. 4.4. Moreover, we will show that the sphere codes have the smallest rate loss for a fixed shaping rate of $R_{\rm s}$ at any blocklength N.

Example 4.2 (Sphere shaping). As in Example 4.1, we target a shaping rate of 1.75 bit/1-D with $\mathcal{A} = \{1, 3, 5, 7\}$. At $N = 96$, we use the maximum energy $E^{\bullet} = 1120$. The rate and the average energy of the corresponding sphere code are $R_{\rm s} = 1.7503$ and $E_{\rm av} = 11.4263$, respectively. The rate loss is $R_{\rm loss} = 0.0232$ bit/1-D. At $N = 216$, we use the maximum energy $E^{\bullet} = 2456$. The rate and the average energy of the corresponding sphere code are $R_{\rm s} = 1.7520$ and $E_{\rm av} = 11.2649$, respectively. The rate loss is $R_{\rm loss} = 0.0129$ bit/1-D. In comparison to the constant composition codes in Example 4.1, sphere codes are 0.6 and 0.28 dB more energy efficient at $N = 96$ and 216, respectively.

4.4 Achievability of Shaping Rates

Definition 4.4 (Achievability). The rate-energy pair (R, E) is called achievable if for each $\epsilon > 0$ and for all N large enough, there exists a code with shaping rate and average energy satisfying

$$R_{\rm s} \geq R - \epsilon, \tag{4.16}$$

$$E_{\rm av} \leq E + \epsilon. \tag{4.17}$$

Finally we define the rate-energy function as follows:

$$R(E) \triangleq \max\{R : (R, E) \text{ is achievable}\}. \tag{4.18}$$

Theorem 4.1. The maximum achievable rate for average energy E is

$$R(E) = \max_{A:\mathbb{E}[A^2] \leq E} H(A). \tag{4.19}$$

The proof consists of a converse part and the corresponding achievability proof.

4.4.1 Converse

Consider a shaping code $\mathcal{A}_s$. The shaping rate can be upper-bounded as

$$\begin{aligned} R_s \overset{(4.1)}{=} \frac{\log_2 J}{N} = \frac{1}{N} H\left(A_1 A_2 \cdots A_N\right) &\overset{(c)}{=} \frac{1}{N} \sum_{n=1}^{N} H(A_n | A_{n-1}, \cdots, A_1) \\ &\overset{(d)}{\leq} \frac{1}{N} \sum_{n=1}^{N} H\left(A_n\right) \\ &\overset{(e)}{\leq} H(A) \\ &\overset{(e)}{\leq} H(A_{\text{MB}}) \end{aligned} \tag{4.20}$$

where (c) is an application of the chain rule for entropy [26, Th. 2.5.1], (d) follows from the fact that conditioning cannot increase entropy [26, Th. 2.5.6], and (e) is due to (4.9) and the convexity of entropy. We note that (e) in (4.20) implies (b) in (4.10). Next, we observe that

$$\frac{1}{J} \sum_{j=1}^{J} \frac{1}{N} \sum_{n=1}^{N} |a_{j,n}|^2 = \frac{1}{N} \sum_{n=1}^{N} \sum_{a \in \mathcal{A}} p_n(a) |a|^2 = \sum_{a \in \mathcal{A}} p(a) |a|^2 = \mathbb{E}[|A|^2]. \tag{4.21}$$

We now conclude that for an achievable rate-energy pair (R, E), for all $\epsilon > 0$, and for all large enough N, there exists a random variable A distributed over $\mathcal{A}$ such that

$$R \leq \frac{\log_2 J}{N} + \epsilon \leq H(A) + \epsilon, \tag{4.22}$$

$$E \geq \frac{1}{J} \sum_{j=1}^{J} \frac{1}{N} \sum_{n=1}^{N} a_{j,n}^2 - \epsilon = \mathbb{E}[A^2] - \epsilon. \tag{4.23}$$

If we let, $\epsilon \downarrow 0$ we obtain that

$$R(E) \leq \max_{A: \mathbb{E}[A^2] \leq E} H(A). \tag{4.24}$$

4.4.2 Achievability Based on Constant Composition Codes

Fix an energy E and assume that the random variable A^* maximizes the entropy $H(A^*)$ while satisfying the energy constraint $\mathbb{E}[(A^*)^2] \leq E$. Denote by $\{p(a^*), a \in \mathcal{A}\}$ the distribution—which is MB—corresponding to this random variable. For all large enough N, we now take a composition C that satisfies

$$|C - Np(a^*)| \leq 1 \tag{4.25}$$

where $n_j \geq 0$ for all $j = 1, 2, \cdots, |\mathcal{A}|$, and $\sum_j n_j = N$.

It can be shown that the probabilities $p(a^*) > 0$, see [26, Example 12.2.3]. Therefore, the normalized composition $\{C/N, a \in \mathcal{A}\}$ approaches entropy $H(A^*)$ for increasing N.

Now for fixed N, consider a code consisting of all sequences having the composition $\{C, a \in \mathcal{A}\}$. It can be shown that the rate of this constant composition code approaches the entropy $H(C/N)$ of the normalized composition for increasing N, see again [26, Example 12.2.3], where Stirling approximation is used.

We conclude that the rate of the constant composition code approaches the entropy of the normalized composition, which approaches entropy $H(A^*)$, for N large. Therefore,

$$R(E) = \max_{A:\mathbb{E}[A^2]\leq E} H(A) \tag{4.26}$$

is achievable for all E.

4.4.3 Achievability Based on Sphere Codes

Theorem 4.2. For each code with rate R and average energy E, there is a sphere code with rate $R_\circ$ and average energy $E_\circ$ such that

$$R_\circ = R \text{ and } E_\circ \leq E. \tag{4.27}$$

Proof. Just replace codewords by sequences outside the code with lower energy until the code is a sphere code. □

Theorem 4.2, along with the optimality of constant composition codes, leads to the conclusion that sphere codes achieve the maximum rate as well.

4.4.4 Maxwell-Boltzmann Distribution

The distribution that achieves maximum entropy under an energy constraint is the MB distribution [26, Ch. 12]. It is straightforward to show that the maximum entropy distribution is unique. This follows directly from the strict convexity of the entropy function. Since sphere codes result in maximum entropy under an energy constraint, the corresponding average distribution $p(a)$ approaches the MB distribution as $N \to \infty$. This shows that the sphere codes are optimum according to Definition 4.1. Furthermore, it is shown in [97] that at a finite blocklength N, a sphere code of rate R and average distribution $p(a)$ has the minimum KL divergence between $p(a)$ and the MB distribution with entropy R among all possible codes, see [98, Corollary 4.6].

4.5 Comparison

For comparison purposes, we define an additional shaping code.

Definition 4.5 (Single-shell codes). A code is a single-shell code if all codewords with energy NE_{av} are included, and no other codewords.

Example 4.3 (Single-shell shaping). As in Examples 4.1 and 4.2, we consider a target shaping rate of 1.75 bit/1-D with $\mathcal{A} = \{1, 3, 5, 7\}$. At $N = 96$, we use $NE_{\text{av}} = 1176$. The rate of the corresponding single-shell code is $R_{\text{s}} = 1.7534$ bit/1-D. The rate loss is $R_{\text{loss}} = 0.0568$ bit/1-D. At $N = 216$, we use $NE_{\text{av}} = 2496$. The rate of the corresponding single-shell code is $R_{\text{s}} = 1.7506$ bit/1-D. The rate loss is $R_{\text{loss}} = 0.0277$ bit/1-D. When compared to Examples 4.1 and 4.2, single-shell codes have larger rate loss than sphere codes and smaller rate loss than constant composition codes. This is because sphere codes include all sequences on all possible N-shells, while constant composition codes include only *some* of the sequences on a single N-shell since there are other compositions with the same energy.

4.5.1 Finite Length Rate Loss

In Fig. 4.1, we have compared constant composition, single-shell, and sphere codes for short blocklengths in terms of the rate loss. We fix the target shaping rate $R_{\text{s}} = 1.75$ bit/1-D, and for each N, we choose the composition C, the sequence energy NE_{av}, and the maximum sequence energy $E^{\bullet}$ for constant composition, single-shell, and sphere codes, respectively, such that these codes contain at least $2^{NR_{\text{s}}}$ codewords in the most energy efficient manner. We use the amplitude alphabet $\mathcal{A} = \{1, 3, 5, 7\}$ of 8-ASK. Our observation is twofold.

- As discussed in Sec. 4.4, the rate loss decreases with increasing N (vanishes when $N \to \infty$) for both constant composition and sphere codes. This also holds for single-shell codes, which can be shown straightforwardly. Therefore, all these codes are optimum according to Definition 4.1.
- Sphere codes, since they construct the most energy-efficient codebook, have the minimum rate loss at any blocklength N. Equivalently, for a given rate loss, sphere codes require the minimum blocklength N. As an example, for a target rate loss of 0.05 bit/1-D, constant composition codes require approximately 7 times larger blocklengths than sphere codes as shown in Fig. 4.1.

4.5.2 Shaping Gain & Signal Space Structure

In Fig. 4.2, we visualize the signal space structures that correspond to constant composition, single-shell, and sphere codes to comprehend their energy efficiency. For constant composition codes, each codeword in $\mathcal{A}_{\text{s}}^{\text{cc}}$ has the same energy $E^{\text{cc}} = NE_{\text{av}}$, and consequently, they all are located on the N-shell of squared radius E^{cc}. However, since there are multiple

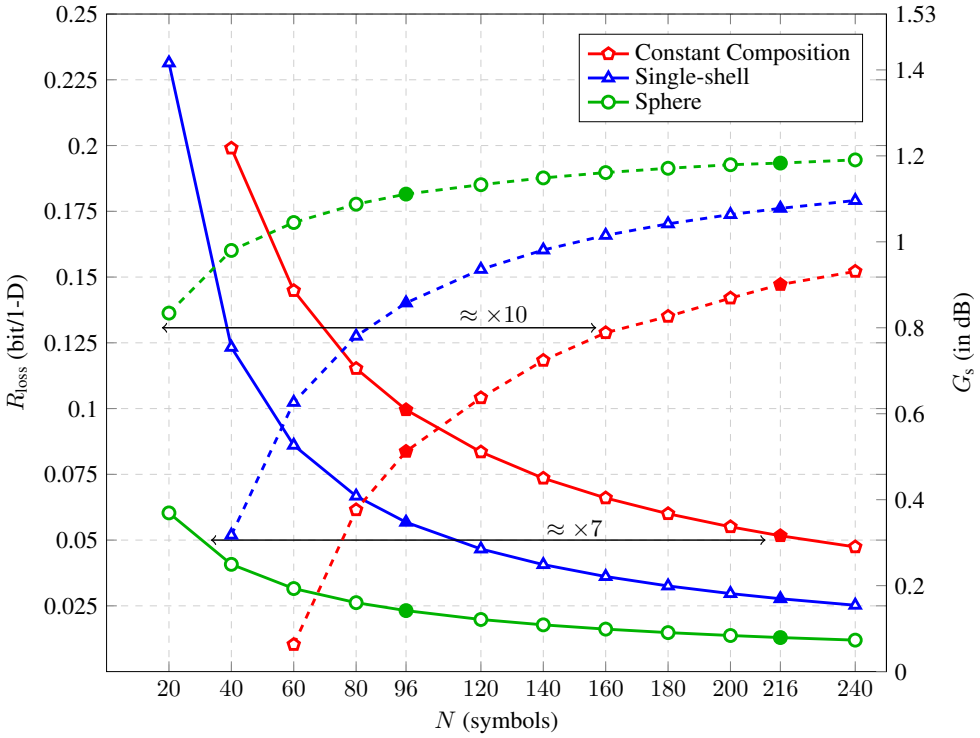

Figure 4.1: Rate loss in (4.11) (solid) and shaping gain in (4.4) (dashed) versus blocklength N for constant composition, single-shell and sphere codes.

compositions that have the same energy, the N-shell is only partially utilized by the constant composition code as shown in Fig. 4.2. On the other hand, single-shell codes consist of all signal points on the N-shell of square radius NE_{av}. Finally, sphere codes $\mathcal{A}^{\bullet}$ consists of all codewords located in or on the surface of the N-sphere of squared radius $E^{\bullet}$.

We see from Fig. 4.2 that for a given rate, i.e., a fixed number of codewords, sphere codes are the most energy-efficient, while constant composition codes are the least energy-efficient among the codes discussed in this chapter. This can also be deducted by observing their shaping gains, which is a metric directly related to E_{av} from (4.4) and thus, to energy efficiency. In Fig. 4.1, shaping gains of constant composition, single-shell, and sphere codes are also plotted. We see that similar to their rate loss minimizing behavior, sphere codes maximize the shaping gain, and thus energy efficiency, at any dimension N as discussed in Theorem 4.2. As an example, for a target shaping gain of 0.8 dB, constant composition codes require approximately 10 times larger blocklengths than sphere codes as shown in Fig. 4.1. In

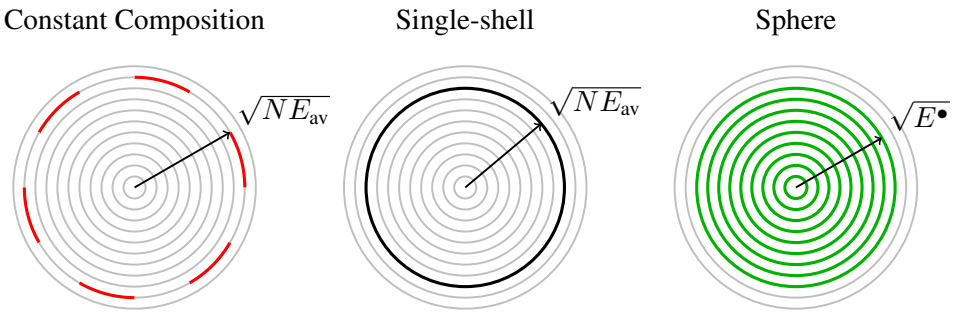

Figure 4.2: The illustration of the employed N-D signal points by constant composition, single-shell, and sphere codes. Each circle represents an N-D shell. Colored portions of the shells indicate the signal points which are included in the corresponding shaping code.

Table 4.1: Parameters Computed in Examples 4.1, 4.2, and 4.3

N	Shaping Code	R_{s}	E_{av}	R_{loss}	G_{s}
96	Constant composition	1.7575	13.2500	0.0995	0.5124
	Single-shell	1.7534	12.2500	0.0568	0.8280
	Sphere	1.7503	11.4263	0.0232	1.1112
216	Constant composition	1.7507	12.0000	0.0516	0.9009
	Single-shell	1.7506	11.5556	0.0277	1.0642
	Sphere	1.7520	11.2649	0.0129	1.1834

the following example, we focus on the blocklengths used in the IEEE 802.11 standard [16].

Example 4.4 (Shaping codes for IEEE 802.11). There are two modes in the IEEE 802.11 standard where there are $N = 96$ or 216 real dimensions reserved for data in a single OFDM symbol [16]. In Table 4.1, parameters of the shaping codes discussed in Examples 4.1, 4.2, and 4.3 are tabulated which were computed at these blocklengths. The corresponding markers are filled in Fig. 4.1. At $N = 96$, sphere codes are 0.6 dB more energy-efficient than constant composition codes. At $N = 216$, the advantage of sphere codes over constant composition codes drops to 0.28 dB. Since both codes are optimum, this difference will vanish as $N \to \infty$.

4.6 Conclusion

In this chapter, we searched for an answer to the following research question.

RQ-2 What is the "best" amplitude shaping strategy for finite values of the blocklength N? What are the metrics to be used to assess the "goodness" of different amplitude shaping approaches?

We demonstrated through rate loss and shaping gain analyses that sphere shaping is the best amplitude shaping strategy, especially for short blocklengths, i.e., smaller than a couple of hundred symbols. Motivated by this, we will discuss how to realize sphere shaping in the following chapters.

4

4

Part III

Enumerative Sphere Shaping Techniques

4

CHAPTER 5

Enumerative Sphere Shaping

Parts of this chapter are published in:

Y. C. Gültekin, W. J. van Houtum, S. Şerbetli, and F. M. J. Willems, "Constellation shaping for IEEE 802.11," in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Commun. (PIMRC),* Montreal, QC, Canada, Oct. 2017.

Y. C. Gültekin, W. J. van Houtum, A. G. C. Koppelaar, and F. M. J. Willems, "Enumerative sphere shaping for wireless communications with short packets," *IEEE Trans. Wireless Commun.,* vol. 19, no. 2, pp. 1098-1112, Feb. 2020.

5.1 Introduction

In Chapter 4, sphere shaping is shown to be an efficient solution to the problem of amplitude shaping for finite (especially for short) blocklengths. This efficiency is in the sense that sphere shaping minimizes the rate loss, or equivalently, minimizes the average energy of the shaping set for a given shaping rate at any blocklength. However, realizing sphere shaping is not a straightforward task since the problem of mapping information bits to channel inputs is multidimensional. We note that this problem, i.e., symbol mapping, is solved straightforwardly in a dimension-by-dimension manner with LUTs of negligible size for uniform signaling.

Example 5.1 (Indexing N-D sequences). Consider a PAS-based transmission strategy where shaping blocklength is $N = 16$, employed modulation is 8-ASK, i.e., $\mathcal{A} = \{1, 3, 5, 7\}$, and the target shaping rate is $R_{\mathrm{s}} = 1.5$ bit/1-D. The output set of the corresponding amplitude shaper consists of $2^{NR_{\mathrm{s}}} = 2^{24}$ amplitude sequences, each of which is of 16-symbols-long. Since each amplitude in $\mathcal{A}$ can be represented using $\log_2 |\mathcal{A}| = m_{\mathrm{a}} = 2$ bits, a LUT-based shaper implementation would require 67.11 megabytes (MB) of dedicated memory.

We conclude from Example 5.1 that even at very small blocklengths, using a LUT for shaping is impractical due to very large storage requirements, and thus, we need constructive algorithms. In this chapter, we will investigate the research question **RQ-3** which deals with the algorithmic implementation and end-to-end decoding performance of sphere shaping. We will introduce *enumerative sphere shaping (ESS)* as an efficient solution. Building upon the foundation established in [29], we will investigate the performance of ESS in the PAS framework for both the AWGN channel and frequency selective channels. Furthermore, the required storage and computational complexity of ESS will be analyzed and compared to alternative sphere shaping algorithms proposed by Laroia, Farvardin, and Tretter in [30].

5.2 Enumerative Sphere Shaping (ESS)

5.2.1 Lexicographical Ordering

ESS starts from the assumption that the amplitude sequences (in a sphere code $\mathcal{A}^{\bullet}$) can be ordered lexicographically.

Definition 5.1 (Lexicographical ordering). A sequence $a^N = (a_1, a_2, \cdots, a_N) \in \mathcal{A}^{\bullet}$ is "larger" than $b^N = (b_1, b_2, \cdots, b_N) \in \mathcal{A}^{\bullet}$ if there exists an integer n such that $a_j = b_j$ for $1 \leq j < n$ and $a_n > b_n$. Then we write $a^N > b^N$, and define the index

$$i\left(a^N\right) \triangleq \left|\left\{b^N \in \mathcal{A}^{\bullet} : a^N > b^N\right\}\right|. \tag{5.1}$$

The mapping from sequences in $\mathcal{A}^{\bullet}$ to indices is one-to-one, and thus,

$$a^N\left(i\right) = a^N \text{ if } i\left(a^N\right) = i. \tag{5.2}$$

Table 5.1: Sphere Shaping Set for $N = 4$, $\mathcal{A} = \{1, 3, 5, 7\}$ and $E^\bullet = 28$ as Ordered by ESS, LA1 and SM

	Lexicographical		**Energy-Based**			
	ESS		LA1		LA2 (SM)	
index i	$a^N(i)$	e	$a^N(i)$	e	$a^N(i)$	e
0	$(1,1,1,1)$	4	$(1,1,1,1)$	4	$(1,1,1,1)$	4
1	$(1,1,1,3)$	12	$(1,1,1,3)$	12	$(1,1,1,3)$	12
2	$(1,1,1,5)$	28	$(1,1,3,1)$	12	$(1,1,3,1)$	12
3	$(1,1,3,1)$	12	$(1,3,1,1)$	12	$(1,3,1,1)$	12
4	$(1,1,3,3)$	20	$(3,1,1,1)$	12	$(3,1,1,1)$	12
5	$(1,1,5,1)$	28	$(1,1,3,3)$	20	$(1,1,3,3)$	20
6	$(1,3,1,1)$	12	$(1,3,1,3)$	20	$(1,3,1,3)$	20
7	$(1,3,1,3)$	20	$(1,3,3,1)$	20	$(1,3,3,1)$	20
8	$(1,3,3,1)$	20	$(3,1,1,3)$	20	$(3,1,1,3)$	20
9	$(1,3,3,3)$	28	$(3,1,3,1)$	20	$(3,1,3,1)$	20
10	$(1,5,1,1)$	28	$(3,3,1,1)$	20	$(3,3,1,1)$	20
11	$(3,1,1,1)$	12	$(1,1,1,5)$	28	$(1,1,1,5)$	28
12	$(3,1,1,3)$	20	$(1,1,5,1)$	28	$(1,1,5,1)$	28
13	$(3,1,3,1)$	20	$(1,3,3,3)$	28	$(1,3,3,3)$	28
14	$(3,1,3,3)$	28	$(1,5,1,1)$	28	$(3,1,3,3)$	28
$2^k - 1 = 15$	$(3,3,1,1)$	20	$(3,1,3,3)$	28	$(3,3,1,3)$	28
16	$(3,3,1,3)$	28	$(3,3,1,3)$	28	$(3,3,3,1)$	28
17	$(3,3,3,1)$	28	$(3,3,3,1)$	28	$(1,5,1,1)$	28
18	$(5,1,1,1)$	28	$(5,1,1,1)$	28	$(5,1,1,1)$	28

We can use lexicographical ordering to create a mapping that transforms a message (index) into a sequence, and vice versa, as shown in Example 5.2.

Example 5.2 (Lexicographically ordered sphere shaping set). Consider the parameters $N = 4$, $\mathcal{A} = \{1, 3, 5, 7\}$ and $E^\bullet = 28$. In Table 5.1, we see the corresponding $a^N \in \mathcal{A}^\bullet$ lexicographically ordered, and their index i. We note that the amplitude 7 never occurs in Table 5.1 since $E^\bullet = 28$ does not allow any sequence to include it.

5.2.2 Backward Amplitude Trellis

To represent lexicographically-ordered amplitude sequences from within a sphere, we build a trellis as shown in Fig. 5.1 for the same set of parameters used in Example 5.2. In this trellis, nodes in column $n = 0, 1, 2, \cdots, N$ represent accumulated energy of amplitude sequences

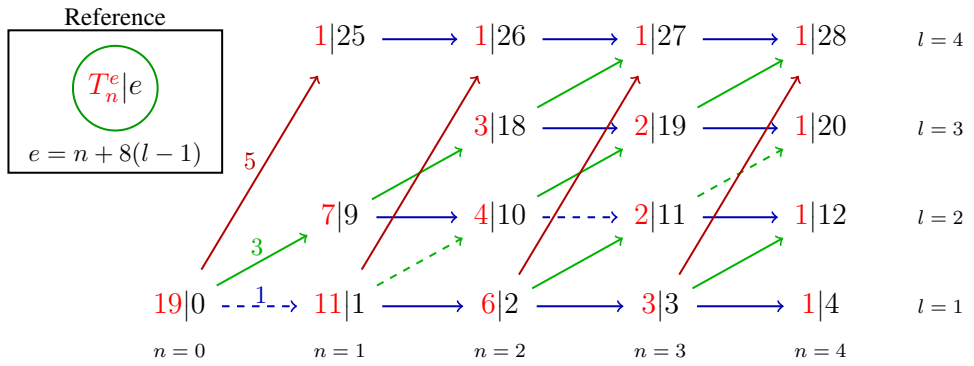

Figure 5.1: Enumerative (backward) trellis for $N = 4$, $\mathcal{A} = \{1, 3, 5, 7\}$, and $E^\bullet = 28$.

over the first n dimensions, more precisely,

$$e\left(a_1, a_2, \cdots, a_n\right) = \sum_{j=1}^{n} a_j^2. \tag{5.3}$$

These energy values are shown by black numbers in Fig. 5.1. We use the pair (n, e) to address a specific node. Branches connecting a node in column $n-1$ with a node in column n are the n^{th} components of the amplitude sequences $a_n \in \mathcal{A}$ for $n = 1, 2, \cdots, N$. Amplitudes are differentiated by color-coding and indicated for $n = 1$ in Fig. 5.1. Each amplitude sequence is represented by a path that consists of N branches, starts in the zero-energy node (the bottom left), and ends in a final node (from the rightmost column, $n = N$). As an example, the sequence $(1, 3, 1, 3)$ is shown with dashed lines in Fig. 5.1.

The number written in red in a node (n, e) is the number of possible ways to reach a final node starting from (n, e), and it is denoted by T_n^e. These can be computed in a recursive manner for $n = N-1, N-2, \cdots, 0$ as

$$T_n^e \triangleq \sum_{a \in \mathcal{A}: e + a^2 \le E^\bullet} T_{n+1}^{e+a^2} \tag{5.4}$$

where the initialization is[1]

$$T_N^e = \begin{cases} 1: & e \le E^\bullet, \\ 0: & \text{otherwise}. \end{cases} \tag{5.5}$$

Note that we only consider states with energy levels that can be reached. For ASK alphabets as in 2.7, possible states in column n have energy values $n + 8(l-1)$ for $l = 1, 2, \cdots, L$,

[1] Since the trellis computation starts from $n = N$, i.e., from the final column, this trellis is also called the enumerative *backward* trellis.

not exceeding $E^\bullet$. Then the maximum energy can be written as

$$E^\bullet = N + 8(L-1) \tag{5.6}$$

where L is the number of possible energy levels at $n = N$, or equivalently, the number of N-shells that are represented in the trellis.

According to the definition of T_n^e, the number of sequences represented in the trellis is given by $|\mathcal{A}^\bullet| = T_0^0$, and for Fig. 5.1, $T_0^0 = 19$, which is consistent with Table 5.1. As defined in (4.1), the shaping rate of this trellis is

$$R_{\mathrm{s}} = \frac{\log_2 |\mathcal{A}^\bullet|}{N} = \frac{\log_2 T_0^0}{N} \tag{5.7}$$

in bit/1-D, and for Fig. 5.1, $R_{\mathrm{s}} = 1.06$ bit/1-D.

Lemma 5.1 (Distribution symmetry). In an enumerative backward trellis, consider all T_n^e sequences $(a_{n+1}(i), a_{n+2}(i), \cdots, a_N(i))$ starting from (n, e) that are indexed by $i = 0, 1, \cdots, T_n^e - 1$. For **this** set of sequences and for $j = n+1, n+2, \cdots, N$,

$$p_j(a) \triangleq \frac{1}{T_n^e} \sum_{i=0}^{T_n^e - 1} \mathbb{1}\left[a_j(i) = a\right], \text{ for } a \in \mathcal{A}. \tag{5.8}$$

Then again for this set of sequences and for $a \in \mathcal{A}$,

$$p_{n+1}(a) = p_{n+2}(a) = \cdots = p_N(a) = \frac{T_{n+1}^{e+a^2}}{\sum_a T_{n+1}^{e+a^2}} \overset{(5.4)}{=} \frac{T_{n+1}^{e+a^2}}{T_n^e} \tag{5.9}$$

i.e., the distribution of amplitudes is the same for all time indices $n+1, n+2, \cdots, N$. The reason for this symmetry is that the set of sequences is permutation invariant.

Lemma 5.1 implies that the average amplitude distribution $p(a)$ of the sphere codebook $\mathcal{A}^\bullet$ (as defined in (4.9)) is given by

$$p(a) = \frac{T_1^{a^2}}{\sum_{b \in \mathcal{A}} T_1^{b^2}}, \quad \text{for } a \in \mathcal{A}. \tag{5.10}$$

For the trellis in Fig. 5.1, $p = [T_1^1, T_1^9, T_1^{25}, T_1^{49}]/T_0^0 = [11/19, 7/19, 1/19, 0]$.

Example 5.3 (The amount of shaping). For a given N and $\mathcal{A}$, decreasing $E^\bullet$, i.e., decreasing the radius of the N-spherical signal structure, leads to a "more shaped" $p(a)$ that has smaller entropy, and vice versa. In Table 5.2, the induced distribution is tabulated for different values of $E^\bullet$ while $N = 4$ and $\mathcal{A} = \{1, 3, 5, 7\}$. Here, the third row corresponds to Example 5.2 and Fig. 5.1, and the last row to uniform signaling.

Remark 5.1 (The relation between L and $E^\bullet$). We see from the sphere-hardening result which is discussed, e.g., by Wozencraft and Jacobs in [35, Sec. 5.5], that for large N, $E^\bullet \approx N E_{\mathrm{av}}$. Following Laroia *et al.* [30, Sec. III-A] and approximating the required average energy to transmit R bit/1-D by $c 2^{2R}$, we can write $E^\bullet \approx N c 2^{2R}$ where c is some constant. Therefore, as seen from (5.6), L is a linear function of N for fixed R_{s}.

Table 5.2: Average Distribution for $N = 4$, $\mathcal{A} = \{1, 3, 5, 7\}$

$E^\bullet$	$p(1)$	$p(3)$	$p(5)$	$p(7)$	$H(A)$
4	1	0	0	0	0
12	0.8	0.2000	0	0	0.7219
28	0.5789	0.3684	0.0526	0	1.2108
60	0.4268	0.3171	0.2073	0.0488	1.7329
196	0.2500	0.2500	0.2500	0.2500	2

5.2.3 Shaping Algorithms

Assuming lexicographical ordering, finding the index i of a sequence a^N is equivalent to count the number of sequences that are lexicographically smaller than a^N. This can be realized by considering the path representing a^N in the backward trellis and counting the number of paths that branch off to "lower" nodes. This leads to Cover's indexing formula for sequences in a sphere [99]

$$i\left(a^N\right) = \sum_{n=1}^{N} \sum_{b<a_n} T_n^{b^2+\sum_{j=1}^{n-1} a_j^2}. \tag{5.11}$$

Example 5.4 (Enumerative indexing). Consider $a^N = (1, 3, 1, 3)$ which has the path passing through nodes $(0, 0)$, $(1, 1)$, $(2, 10)$, $(3, 11)$ and $(4, 20)$ in Fig. 5.1, i.e., the path drawn with dashed lines. At each transition for which there is a possible transition with a smaller amplitude, i.e., the second and the fourth transitions, we add the red numbers in the corresponding lower nodes, i.e., $T_2^2 = 6$ and $T_4^{12} = 1$, to find the index $i(a^N)$ which is 7. This mapping is consistent with Table 5.1.

Algorithm 5.1: Enumerative Shaping

1 Given that $i < T_0^0$, initialize the algorithm by setting the *local index* $I_1 = i$.

2 For $n = 1, 2, \cdots, N$, take a_n be such that

$$\sum_{b<a_n} T_n^{b^2+\sum_{j=1}^{n-1} a_j^2} \le I_n < \sum_{b\le a_n} T_n^{b^2+\sum_{j=1}^{n-1} a_j^2} \tag{5.12}$$

and

$$I_{n+1} = I_n - \sum_{a<a_n} T_n^{b^2+\sum_{j=1}^{n-1} a_j^2}. \tag{5.13}$$

3 Finally output a^N.

The indexing formula (5.11), i.e., the procedure of finding the index of an amplitude sequence, is called *deshaping*. The inverse function that determines from a message index the sequence in a sphere is called *shaping*. These shaping and deshaping algorithms can be implemented recursively, and they are outlined in Algorithms 5.1 and 5.2, respectively.

Algorithm 5.2: Enumerative Deshaping

1 Given a^N, initialize the algorithm by setting the *local index* $J_{N+1} = 0$.
2 For $n = N, N-1, \cdots, 1$, update the local index as

$$J_n = \sum_{b<a_n} T_n^{b^2+\sum_{j=1}^{n-1} a_j^2} + J_{n+1}. \tag{5.14}$$

3 Finally output $i = J_1$.

5.3 Laroia's Sphere Shaping Algorithms

5.3.1 Energy-based Ordering

Laroia, Farvardin, and Tretter provided two algorithms to realize sphere shaping in [30], both of which sort the sequences in a sphere based on their energy, i.e., based on the index l of the N-D shell that they are located on for $l = 1, 2, \cdots, L$. Sequences on the same shell can then be ordered in different ways, e.g., lexicographically as done by Laroia *et al.* in [30, Algorithm 1] which is shown in the middle column of Table 5.1. We denote this algorithm by LA1 here.

5.3.2 Forward Amplitude Trellis

To represent energy-based-ordered amplitude sequences from within a sphere, again a trellis is constructed as shown in Fig. 5.2 for the same set of parameters used in Fig. 5.1. Similar to the backward trellis, black numbers represent energy levels, branches indicate amplitudes, and we use the pair (n, e) to indicate a specific node in Fig. 5.2

The red number written in a node (n, e) is the number of n-sequences with energy e for $n = 1, 2, \cdots, N$ and $e \leq E^\bullet$, and it is denoted by F_n^e. These can be computed in a recursive manner for $n = 1, 2, \cdots, N$ as

$$F_n^e \triangleq \sum_{a \in \mathcal{A}: e-a^2 \leq E^\bullet} F_{n-1}^{e-a^2} \tag{5.15}$$

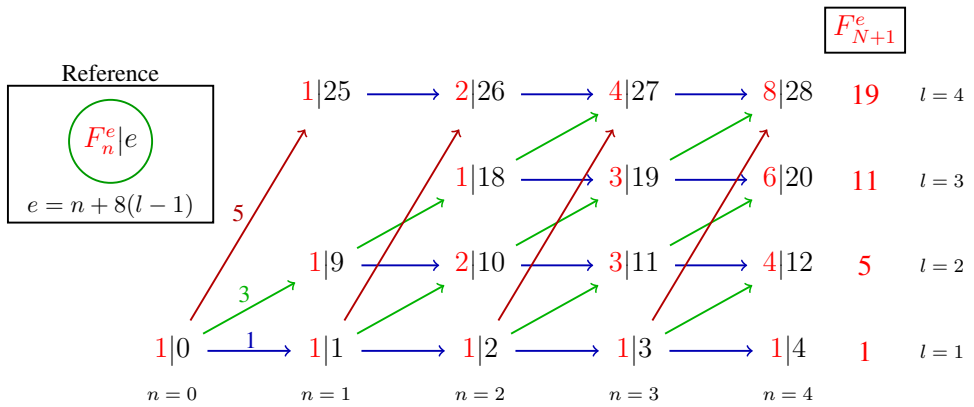

Figure 5.2: Enumerative (forward) trellis for $N = 4$, $\mathcal{A} = \{1, 3, 5, 7\}$, and $E^{\bullet} = 28$.

where the initialization is[2]

$$F_0^e = \begin{cases} 1: & e = 0, \\ 0: & \text{otherwise}. \end{cases} \tag{5.16}$$

In this forward amplitude trellis, the numbers F_N^e in the last column are the number of sequences located on the N-shell of squared radius $e = N, N + 8, \cdots, E^{\bullet}$. For Fig 5.2, and consequently for Fig. 5.1, there are 1 sequence of energy 4, 4 sequences of energy 12, 6 sequences of energy 20, and 8 sequences of energy 28 as shown in Table 5.1.

Remark 5.2 (Alternative way of computing the forward trellis). If N is an integer power of two, the n^{th} column of the forward trellis can be computed for $n = 1, 2, 4, \cdots, N$ as

$$F_n^e = \sum_{b \le e} F_{n/2}^b F_{n/2}^{e-b} \tag{5.17}$$

where the initialization is still as in (5.16). We note that unlike (5.15), multiplications are necessary when (5.17) is used. This way of computing *some* columns of the forward trellis will be particularly useful when we discuss shell mapping in Sec. 5.3.3.4.

Following [30, Sec. II-B], we define an additional column for the forward trellis in which the number of sequences with energy no greater than e

$$F_{N+1}^e = \sum_{b \le e} F_N^b \tag{5.18}$$

[2]Since the trellis computation starts from $n = 0$, i.e., from the zero-energy node, this trellis is also called the enumerative *forward* trellis.

is stored for $e \leq E^{\bullet}$. Clearly, $F_{N+1}^{E^{\bullet}} = T_0^0$, from which the shaping rate can be computed as in (5.7). Similar to (5.10), the average distribution can be computed as[3]

$$p(a) = \frac{\sum_{l=1}^{L} F_{N-1}^{N+8(l-1)-a^2}}{\sum_{b\in\mathcal{A}} \sum_{l=1}^{L} F_{N-1}^{N+8(l-1)-b^2}}, \qquad \text{for } a \in \mathcal{A}. \tag{5.19}$$

5.3.3 Shaping Algorithms

To realize the energy-based ordering, first the N-shell that the output sequence is located on, then the index of the sequence within this shell must be found during shaping. We call the former; *extra step*, the latter; *shaping within the N-shell*. Both of these steps can be realized in two different ways.

5.3.3.1 Finding the N-shell: The Extra Step with No Storage

The extra step can be realized by successive comparisons of the index i to F_N^e for $e = N, N+8, \cdots, E^{\bullet}-8$, and updating the index by subtracting F_N^e from it whenever the index is larger. In this way of implementing the extra step, there can be up to $L-1$ comparisons and subtractions, and F_{N+1}^e is **not** required to be stored. Since a high portion of the sequences is located near the surface of the N-sphere due to sphere hardening, the average number of comparisons required to find the N-shell is very close to $L-1$ with this approach.

Example 5.5 (Extra step with no storage). Consider the forward trellis in Fig. 5.2 and input index $i = 9$. To determine which N-shell the corresponding amplitude sequence is located on, we first compare i to $F_N^N = F_4^4 = 1$ which is the number of sequences on the first, i.e., innermost, shell. Since $i > F_4^4$, we update $i \leftarrow i - F_4^4 = 8$. Then we compare i to $F_N^{N+8} = F_4^{12} = 4$ which is the number of sequences on the second shell. Since $i > F_4^{12}$, we update $i \leftarrow i - F_4^{12} = 4$. Then we compare i to $F_N^{N+16} = F_4^{20} = 6$ which is the number of sequences on the third shell. Since $i < F_4^{20}$, we decide that our sequence is located on the third shell of squared radius 20, and its index within this shell is $I_N(a^N) = 4$.

5.3.3.2 Finding the N-shell: The Extra Step with Storage

The extra step can also be realized by successive comparisons of the index i to F_{N+1}^e for $e = E^{\bullet}-8, E^{\bullet}-16, \cdots, N$, and only updating the index by subtracting F_{N+1}^e the first time the index is larger. In this way of implementing the extra step, there can be up to $L-1$ comparisons, but only a single subtraction. Since a high portion of the sequences is located near the surface of the N-sphere due to sphere hardening, only a couple of comparisons are required on average to find the N-shell with this approach. However, F_{N+1}^e must be stored.

[3] To write (5.19), we used the fact that when rotated 180°, the forward trellis in Fig. 5.2 represents L separate single shell backward trellises as discussed in Example 5.7.

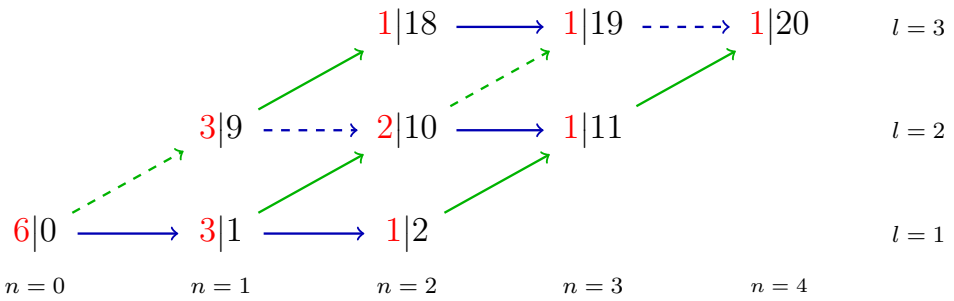

Figure 5.3: The subtrellis of Fig. 5.2 that belongs to the shell of squared radius 20.

Example 5.6 (Extra step with storage). Consider again the forward trellis in Fig. 5.2 and input index $i = 9$. To determine which N-shell the corresponding amplitude sequence is located on, we first compare i to $F_{N+1}^{E^\bullet-8} = F_5^{20} = 11$ which is the number of sequences on the three innermost shells. Since $i < F_5^{20}$, we compare the index to $F_{N+1}^{E^\bullet-16} = F_5^{12} = 5$ which is the number of sequences on the two innermost shells. Since $i > F_5^{12}$, we decide that our sequence is located on the third shell of squared radius 20, and its index within this shell is $I_N(a^N) = i - F_5^{12} = 4$.

5

5.3.3.3 Shaping within the N-shell: Laroia's Algorithm 1

LA1 is similar to ESS from both algorithmic and complexity perspectives. After the index l of the N-shell that the output sequence is located on, and the local index $I_N(a^N)$ within this shell are found, enumerative shaping procedure in Algorithm 5.1 is run for $I_N(a^N)$ over the part of the forward trellis that corresponds to the l^{th} N-shell as explained in the following example. With this algorithm, the sequences on the same shell are sorted lexicographically as shown in Table 5.1. We note that F_n^e must be stored in memory for $n = 0, 1, \cdots, N$ and $e \leq E^\bullet$ to realize shaping and deshaping algorithms based on LA1.

Example 5.7 (LA1: Shaping within the N-shell). Consider again the forward trellis in Fig. 5.2 and input index $i = 9$. We found in Examples 5.5 and 5.6 that the corresponding sequence is on the third shell of squared radius 20, and it has index $I_N(a^N) = 4$ within this shell. Then Algorithm 5.1 is run with the *backward* trellis shown in Fig. 5.3 to find that the sequence in this shell with index 4 is $(3, 1, 3, 1)$ which is shown with dashed lines. This (rotated) backward subtrellis is the part of the forward trellis in Fig. 5.2 that represents only the sequences located on the third shell.

5.3.3.4 Shaping within the N-shell: Shell Mapping

The second sphere shaping algorithm that is based on the forward trellis provided by Laroia *et al.* in [30, Algorithm 2] is the well-known shell mapping (SM). We also denote SM by LA2 to be consistent with our denotation of LA1. SM is based on the divide-and-conquer (D&C) principle as used in [100]. The sequences on a given N-shell are ordered with respect to the index of their first half, and the ones having identical first halves with respect to the index of their second half as shown in Table 5.1. This principle is applied recursively such that the n-D problem is successively divided into two $n/2$-D problems. In the end, the 2-D problem can be solved easily by a LUT [101, Example 8.2]. Here we assume that N is an integer power of two. An efficient way of implementing SM as discussed in [30] and [101] is formulated in Algorithms 5.3 and 5.4. We note that F_n^e must be stored in memory for $n = 0, 1, 2, 4, \cdots, N$ and $e \leq E^\bullet$ to realize shaping and deshaping based on SM. These columns of the forward trellis can be computed using (5.17).

Algorithm 5.3: SM Shaping

For $n = N, N/2, \cdots, 4$:

1 The energy e_1 of the first half a_1^n of a^n (and consequently the energy e_2 of the second half a_2^n) is determined by taking e_1 such that

$$\sum_{b<e_1} F_{n/2}^b F_{n/2}^{e(a^n)-b} \leq I_n(a^n) < \sum_{b\leq e_1} F_{n/2}^b F_{n/2}^{e(a^n)-b}, \tag{5.20}$$

and then setting $e_2 = e(a^n) - e_1$.

2 Residual offset D_s follows from

$$D_\mathrm{s} = I_n\left(a^n\right) - \sum_{b<e_1} F_{n/2}^b F_{n/2}^{e(a^n)-b}. \tag{5.21}$$

3 The local offsets $I_{n/2}\left(a_1^n\right)$ and $I_{n/2}\left(a_2^n\right)$

$$I_{n/2}\left(a_1^n\right) = \left\lfloor \frac{D_\mathrm{s}}{F_{n/2}^{e_2}} \right\rfloor, \tag{5.22a}$$

$$I_{n/2}\left(a_2^n\right) = D_\mathrm{s} - I_{n/2}\left(a_1^n\right) F_{n/2}^{e_2} \tag{5.22b}$$

are computed.

Finally, mapping from depth-2 offsets to symbols is straightforward [101].

Algorithm 5.4: SM Deshaping

Note that $J_1\left(a_1^2\right) = J_1\left(a_2^2\right) = 0$.

1 For $n = 2, 4, \cdots, N$:

$$D_{\mathrm{d}} = J_{n/2}\left(a_1^n\right) F_{n/2}^{e\left(a_2^n\right)} + J_{n/2}\left(a_2^n\right), \tag{5.23a}$$

$$J_n\left(a^n\right) = \sum_{k<e\left(a_1^n\right)} F_{n/2}^{k} F_{n/2}^{e\left(a^n\right)-k} + D_{\mathrm{d}}. \tag{5.23b}$$

Finally output $I_N(a^N) = J_N$.

5.4 Required Storage and Computational Complexity

5.4.1 Operational Input Length

In most practical systems, information that is to be transmitted, and thus, a data string to be inputted to the amplitude shaper is binary. Accordingly, we define the operational input length of a sphere shaper as

$$k \triangleq \lfloor \log_2 |\mathcal{A}^{\bullet}| \rfloor = \lfloor N R_{\mathrm{s}} \rfloor \tag{5.24}$$

in bits. Then sphere shaping creates an invertible mapping from k-bit strings to N-amplitude sequences. We note that this means only the first 2^k sequences on the lexicographical or energy-based list are transmitted, and the rest is unused. For ESS, these sequences are **not** necessarily from the outermost shell as shown in Table 5.1. Therefore, the operational average energy for the first 2^k sequences can be smaller, but also larger than E_{av}. For Laroia's sphere shaping, the unused sequences are from the outermost shell as shown in Table 5.1, and the operational average energy is smaller than E_{av}. Furthermore, due to the same reasoning, Laroia's algorithms construct the most energy efficient signal set for a given k. However, as it will be discussed in Chapter 7, the difference in operational average energy for ESS and Laroia's algorithms is negligible for blocklengths larger than a few dozen. Moreover, the backward trellis can be modified using the ideas presented in Sec. 7.2.3 such that ESS constructs roughly the most energy efficient signal set also for very small N.

Remark 5.3 (Exact operational amplitude distribution for sphere shaping). Sphere shaping considers the innermost L shells of an N-D ASK lattice which consists of $2^{NR_{\mathrm{s}}}$ signal points. We call the amplitude distribution averaged over all these sequences the *sphere distribution*. However, only 2^k sequences are transmitted by a communication system. The amplitude distribution averaged over these 2^k sequences—which we call the *operational distribution*—obviously depends on which subset of sequences are used from the initial complete sphere. Therefore, to obtain this distribution, we first need to specify a sphere shaping algorithm that creates an ordering for the amplitude sequences. Then we need to consider

the first 2^k sequences. However, in Fig. 4.1, we assumed that the sphere distribution is an accurate approximation for the operational distribution, and we computed the rate loss accordingly. It will be shown in Chapter 7 that this assumption is valid for $N > 50$, and it creates a negligible difference in rate loss for smaller blocklengths. However, in the context of demonstrating the superiority of sphere shaping over CCDM, it plays no significant role.

5.4.2 Enumerative Sphere Shaping

To realize ESS as explained in Sec. 5.2.3, the backward trellis T_n^e must be stored for $n = 0, 1, \cdots, N$ and $e \leq E^\bullet$. This trellis can be stored in the form of an L-by-$(N+1)$ array $\mathbf{T}$ where its n^{th} column $\mathbf{t}_n$ consists of T_n^e for $e \leq E^\bullet$, and each element is at most $\lceil NR_{\text{s}} \rceil$-bit long. Therefore, the required storage is upper-bounded by $L(N+1)\lceil NR_{\text{s}} \rceil$ bits, or equivalently, $L(N+1)(k+1)$ bits, as shown in Table 5.3. Considering Remark 5.1, we see that the storage complexity is $\mathcal{O}(N^3)$ as a function of N for a fixed shaping rate R_{s}.

Table 5.3: Required Storage and Computational Complexity of Sphere Shaping

Technique	Storage (bits)	Bit Oper./1-D
ESS	$L(N+1)(k+1)$	$(n_{\text{a}}-1)(k+1)$
LA1	$L(N+1)(k+1)+\Lambda_{\text{sto}}$	$(n_{\text{a}}-1)\,(k+1)+\Lambda_{\text{comp}}$
SM	$L\,(\log_2 N+1)\,(k+1)+\Lambda_{\text{sto}}$	$L(k+1)^2+\Lambda_{\text{comp}}$

To compute the backward trellis with (5.4), we need to compute LN numbers, each of which requires at most $(|\mathcal{A}|-1)$ additions. Therefore, at most $LN(n_{\text{a}}-1)(k+1)$ bit operations[4] (bit/oper.) must be carried out as shown in Table 5.4.

Table 5.4: Complexity of Computing Forward and Backward Trellises

Technique	Bit Operations
Backward Trellis with (5.4)	$LN(n_{\text{a}}-1)(k+1)$
Forward Trellis with (5.15)	$LN(n_{\text{a}}-1)(k+1)$
Forward Trellis with (5.17)	$L^2\log_2 N(k+1)^2$

Shaping and deshaping algorithms of ESS demand at most $(n_{\text{a}}-1)$ subtractions and additions of numbers from the backward trellis per dimension, respectively.[5] Therefore, their computational requirement is upper-bounded by $(n_{\text{a}}-1)(k+1)$ bit oper./1-D as shown in

[4] We assume that k-bit additions and subtractions are k-bit operations, and k-bit multiplications and divisions are k^2-bit operations similar to [30].

[5] For Algorithm 5.1, before each subtraction, first a comparison must be carried out. Then depending on the result of the comparison, a subtraction may be necessary. We assume that this can be implemented by realizing a subtraction, and outputting the minuend if the output is negative, and the difference if the output is nonnegative.

Table 5.3. Considering Remark 5.1, we see that the computational complexity behaves as $\mathcal{O}(N)$ as a function of N for a fixed shaping rate R_s.

Example 5.8 (Complexity of ESS for the IEEE 802.11 Standard). The backward trellis constructed with $N = 96$, $\mathcal{A} = \{1, 3, 5, 7\}$, and $L = 129$ has $R_\mathrm{s} = 1.7503$ bit/1-D and $k = 168$ bits. To store this trellis, $L(N + 1)(k + 1) = 264.34$ kilobytes (kB) of memory is required. The corresponding shaping and deshaping algorithms require 507 bit oper./1-D.

5.4.3 Laroia's Sphere Shaping

To compute the forward trellis with (5.15), we need the same amount of bit oper. as computing the backward trellis with (5.4) as shown in Table 5.4. When on the other hand (5.17) is used to compute it, $L \log_2 N$ numbers must be computed, where each computation requires at most L multiplications and additions. Therefore (neglecting the complexity of additions), at most $L^2 \log_2 N(k + 1)^2$ bit oper. must be carried out as shown in Table 5.4. Here we also neglected the complexity of computing F^e_{N+1} in case the extra step is realized with storage as explained in Sec. 5.3.3.2, since it only requires additions.

5.4.3.1 The Extra Step

The extra step with no storage requires at most $L - 1$ comparisons and subtractions (or additions) of numbers from the trellis *per* N*-D*, and thus, its computational requirement is at most $(L - 1)(k + 1)/N$ bit oper./1-D as shown in Table 5.5.

Table 5.5: Complexity of the Extra Step

	Λ_comp	Λ_sto
With no storage	$(L - 1)(k + 1)/N$	0
With storage	$(L - 1)(k + 1)/N$	$L(k + 1)$

The extra step with storage requires at most $L - 1$ comparisons and 1 subtraction (or addition) of numbers from the trellis *per* N*-D*, and thus, its computational requirement is also at most $(L - 1)(k + 1)/N$ bit oper./1-D. However, an additional column needs to be stored using $L(k + 1)$ bits along with the part of the forward trellis that is stored for the subsequent shaping algorithm. The additional computational requirement and (possible) increase in required storage due to the extra step are indicated by Λ_comp and Λ_sto in Tables 5.3 and 5.5, respectively.

Remark 5.4 (Complexity of the extra step). From Table 5.5, it seems that the extra step should be realized using the approach with no storage. However, we note that the computational requirements tabulated in Table 5.5 are the worst-case values. The extra step with

no storage starts processing the number of sequences on the innermost shell, working outwards. On the other hand, the extra step with storage starts from the outermost shell, working inwards. Therefore, since—due to sphere hardening—most sequences are on the outermost shell, the extra step with no storage requires more computations on the average.

5.4.3.2 Laroia's Algorithm 1 (LA1)

The procedure of shaping within the N-shell by LA1 has the same required storage and computational complexity as ESS. Considering the extra step, LA1 is slightly more complex than ESS as indicated in Table 5.3 with Λ_{comp}. Depending on the way the extra step is realized, LA1 may also require more storage than ESS as shown in Table 5.3 with Λ_{sto}.

5.4.3.3 Shell Mapping (LA2)

SM demands the storage of $\log_2 N + 1$ columns of the forward trellis, and thus, the required storage is upper-bounded by $L(\log_2 N+1)(k+1)$ bits excluding Λ_{sto}, as shown in Table 5.3. Considering Remark 5.1, we see that the storage complexity is $\mathcal{O}(N^2 \log N)$ as a function of N for a fixed shaping rate R_{s}.

Shaping and deshaping algorithms of SM require up to L multiplications (or divisions) of numbers from the forward trellis at each step [102]. Unlike ESS, the SM algorithm consists of $\log_2 N$ steps. However, due to the nature of the D&C principle, SM repeats the n^{th} step 2^{n-1} times for $n = 1, 2, \cdots, \log_2 N$. Therefore, as shown in Table 5.3, the computational complexity of SM is upper-bounded by

$$\frac{1}{N} \sum_{n=1}^{\log_2 N} 2^{n-1} L \left\lceil N R_{\text{s}} \right\rceil^2 = \frac{1}{N} L \left\lceil N R_{\text{s}} \right\rceil^2 (N-1) \leq L \left\lceil N R_{\text{s}} \right\rceil^2 = L(k+1)^2 \quad (5.25)$$

bit oper./1-D excluding Λ_{comp}, which has complexity $\mathcal{O}(N^3)$ as a function of N for a fixed shaping rate R_{s}.

Example 5.9 (Complexity of SM for the IEEE 802.11 standard). To implement SM for the same parameters as in Example 5.8 where $N = 96$, the required number of bit oper./1-D is on the order of millions. Thus, we consider $N = 32$ with $L = 48$ which leads to a shaping rate of $R_{\text{s}} = 1.7557$ bit/1-D and an input length of $k = 56$ bits. Three parallel shell mappers can be used shape over 96 dimensions in this case, however with a small loss of efficiency. With these parameters, SM requires at most 155952 bit oper./1-D. The part of the forward trellis that needs to be stored in this case requires 2.05 kB of memory.

5.4.4 Conclusion: Which Algorithm to Use?

To choose among ESS, LA1, and SM, we first consider the computational complexity. Although SM requires smaller storage than the other algorithms, a very large number of multiplications and divisions are necessary, which makes it impractical to implement SM for

blocklengths larger than a few dozen. For instance in the V.34 high speed modem standard [11], SM is realized for $N = 16$. This leaves us with a choice between ESS and LA1.

ESS and LA1 have the same required storage in principle. Due to the extra step necessary to find the N-shell the sequence is located on, LA1 is slightly more complex than ESS. To be fair, we note that LA1 constructs a more energy-efficient signal set than ESS in general. However, this difference in energy efficiency can only be significant for very short blocklengths, and thus, we base our comparison only on complexity. We conclude from this discussion that ESS is an efficient way of realizing sphere shaping, and we prefer to use it.

5.5 End-to-end Decoding Performance

5.5.1 Simulation Settings

5.5.1.1 General Parameters

In this section, we evaluate the performance of ESS in the PAS framework by Monte Carlo simulations. For comparison, uniform signaling, CCDM, and LA1 are also simulated. We note that SM is not simulated, since we expect that it will perform identically to LA1 for the AWGN channel. As the channel input constellation, 4-, 8- and 16-ASK are considered, i.e., $m \in \{2, 3, 4\}$. However, before transmission over the communication channel, two ASK symbols are combined to a single quadrature amplitude modulation (QAM) symbol. At the PAS transmitter, the BRGC is applied by the symbol mapper, and the same mapping is used to label amplitudes at the output of the amplitude shaper. As the FEC code, rate-R_c systematic LDPC codes of length n_c bits are used from the IEEE 802.11 standard [16], where $R_\mathrm{c} \in \{1/2, 2/3, 3/4, 5/6\}$ and $n_\mathrm{c} \in \{648, 1296\}$. Each LDPC codeword corresponds to $N_\mathrm{total} = n_\mathrm{c}/m$ real symbols. At the PAS receiver, the soft-demapper computes LLRs using (2.8). For FEC decoding, the built-in LDPC decoder of MATLAB is used with at most 50 iterations. To assess the performance, frame error rate (FER) curves are plotted where a frame is equivalent to an LDPC codeword, and we declare a frame error when at least one of the information bits is estimated incorrectly. For simulations over the AWGN channel, single carrier transmission is realized whereas, over fading channels, OFDM is implemented.

5.5.1.2 Fair Comparison

Our objective is to compare uniform and shaped signaling structures fairly as discussed in [103], i.e., at a fixed transmission rate R_t. Shaping decreases the entropy $H(X)$ of the constellation, and consequently, the transmission rate. Therefore, to compensate for this effect and to operate at the same transmission rate R_t as the compared uniform signaling scheme, either the constellation size 2^m and/or the FEC code rate R_c of the PAS structure must be higher as discussed in the following example.

Example 5.10 (Fair comparison). Consider a uniform signaling system that employs a rate $R_\mathrm{c} = 3/4$ FEC code followed by an 8-ASK symbol mapper. The transmission rate of this system is $R_\mathrm{t} = mR_\mathrm{c} = 2.25$ bit/1-D. To obtain the same transmission rate with PAS, one of the following approaches can be taken. First, the constellation size can be increased to 16-ASK while keeping the FEC code rate fixed at 3/4, which leads to $\gamma = mR_\mathrm{c} - (m-1) = 0$. Then the parameters of the amplitude shaper can be adjusted such that $k/N = 2.25 < m-1$ and consequently, $R_\mathrm{t} = k/N + \gamma = 2.25$. Second, the FEC code rate can be increased to 5/6 while keeping the constellation size fixed at $m = 3$, which leads to $\gamma = mR_\mathrm{c} - (m-1) = 0.5$. Then the parameters of the amplitude shaper can be selected such that $k/N = 1.75 < m-1$ and consequently, $R_\mathrm{t} = k/N + \gamma = 2.25$.

5.5.1.3 Amplitude Shaping

Shaping is realized over N real dimensions. Note that when $N_\mathrm{total} = N$, shaping and FEC coding blocklengths are the same. When on the other hand $N_\mathrm{total} = \alpha N$ for some integer $\alpha > 1$, each FEC frame consists of α shaped sequences. At each target rate R_t and constellation size 2^m, we choose an FEC code rate R_c for the PAS based on the discussion in Sec. 3.6, i.e., on Wachsmann curves. This FEC code rate results in $\gamma = mR_\mathrm{c} - (m-1)$. Then for sphere shaping, $E^\bullet$ is selected as the smallest value that satisfies $k/N + \gamma \geq R_\mathrm{t}$. For CCDM, the most energy-efficient composition that has at least 2^k sequences is selected.

5.5.1.4 Frequency-selective Channels

Frequency-selective fading realizations are produced using the type-D HiperLAN/2 channel model which is based on a Rician-modeled tapped delay line [104]. Doppler spread is taken to be zero. Perfect CSI is assumed to be available at the receiver. For simulations over fading channels, OFDM is used as the modulation format as specified in the IEEE 802.11 standard. The bandwidth is set to 40 MHz, and it is separated into 128 subcarriers among which 108 are used for data, 6 are occupied by pilots, and the remaining 14 are empty, see [16, Sec. 21.3.7.2] for the actual subcarrier mapping. The cyclic prefix length is taken to be 25 % of an OFDM symbol duration. As the constellation, 4- and 8-ASK are used which leads to $N_\mathrm{total} \in \{648, 432\}$ for $n_\mathrm{c} = 1296$ bits, respectively. Thus a codeword, i.e., a frame, consists of three or two OFDM symbols for schemes based on 4-ASK and 8-ASK, respectively. Shaping is always realized over an OFDM symbol, i.e., $N = 216$.

5.5.2 The AWGN Channel

5.5.2.1 Performance of ESS at Different Blocklengths

In Fig. 5.4, FER is plotted versus SNR for PAS and uniform signaling with 16-ASK and $n_\mathrm{c} = 648$ bits. The transmission rate is $R_\mathrm{t} = 3$ bit/1-D. For this constellation and transmission rate, the FEC code rate that minimizes ΔSNR in (3.97) is 0.85 which should be combined with

Table 5.6: Parameters of ESS for Fig. 5.4 and 5.5

N	$E^\bullet$	k/N (bit/1-D)	R_{loss} (bit/1-D)	E_{av}	G_{s} (in dB)
6	374	2.667	0.1181	46.83	0.57
54	2302	2.667	0.0365	41.02	1.15
162	6514	2.667	0.0169	39.69	1.29
486	19086	2.667	0.0073	39.10	1.36

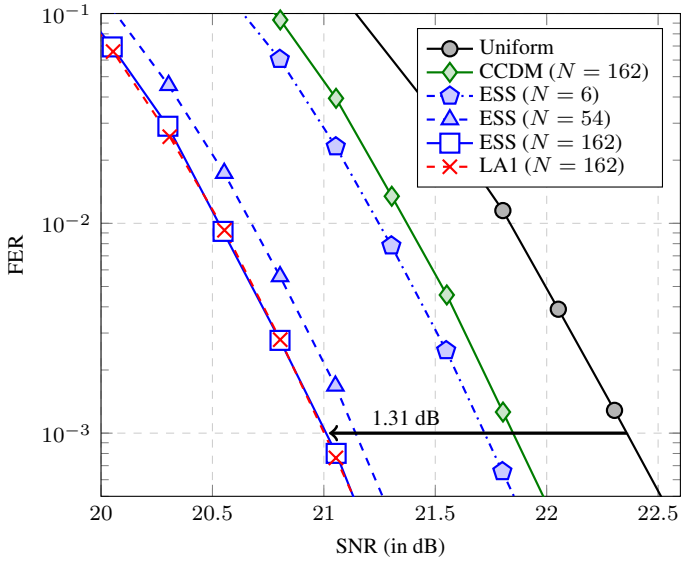

Figure 5.4: 648-bit LDPC-coded FER vs. SNR with 16-ASK at $R_{\text{t}} = 3$ bit/1-D.

the MB distribution that has $H(X) = 3.6$. Therefore, we combine ESS and CCDM with the rate-$5/6$ FEC code which is the closest to 0.85 in the IEEE 802.11 standard [16]. This leads to $\gamma = R_{\text{c}}m - (m-1) = 1/3$. Shaping is realized over $N \in \{6, 54, 162\}$ dimensions for ESS leading to $\alpha = \{27, 3, 1\}$ shaping blocks inside a single FEC codeword, respectively. In Table 5.6, corresponding parameters and metrics for ESS are tabulated. Uniform signaling is realized with the FEC code of rate $R_{\text{c}} = R_{\text{t}}/m = 3/4$.

We see from Fig. 5.4 that at an FER of 10^{-3} and at $N = 6$, 54, and 162, ESS is 0.59, 1.16, and 1.31 dB more power-efficient than uniform signaling, respectively. The first observation is that these improvements are in agreement with the shaping gain results in Table 5.6. Secondly, ESS provides more than half a dB gain even at a very small blocklength of $N = 6$ which enables a trade-off between shaping gain and complexity. More precisely, when the primary objective is not to maximize the gain but to provide a granular set of transmission

rates, one can achieve this with ESS over only a couple of dimensions while still having a significant SNR improvement. For comparison, multiset-partition distribution matching (MPDM) [93] needs at least 20 16-ASK symbols to only perform as good as uniform signaling where CCDM requires even more [105]. Finally, we note that 95 % of the gain achieved at $N = 162$ can be reaped already at $N = 54$ which exhibits diminishing returns. Therefore, performance-wise, it is possible to secure most of the possible shaping gain with sphere shaping, without increasing the blocklength well above a couple of hundreds. Finally, the performance of PAS using LA1 for amplitude shaping at $N = 162$ is also shown in Fig. 5.4. We see that ESS and LA1 perform virtually identical since the difference in their energy efficiency is negligible as discussed in Sec. 5.4.1.[6]

Remark 5.5 (**Sphere shaping at $N = 6$ using a LUT).** To realize sphere shaping at $N = 6$ which is shown to provide 0.59 dB gain over uniform signaling in Fig. 5.4, a LUT which consists of $2^k = 2^{16}$ entries is necessary. Considering that each entry includes 6 amplitudes, and amplitudes of 16-ASK can be represented with 3 bits, the size of the LUT is computed to be more than 147 kB. However, if the corresponding backward amplitude trellis is computed using (5.4), only 5593 bits of memory is required by ESS for storage.

5.5.2.2 Performance of CCDM at Different Blocklengths

In Fig. 5.5, FER is plotted versus SNR for the same set of parameters as in Fig. 5.4, except the LDPC codeword length is $n_{\mathrm{c}} = 1944$ bits now, and consequently, $N_{\mathrm{total}} = 486$. Both CCDM and ESS are considered at blocklengths $N = 162$ and $N = 486$. In the former case, there are three shaped codewords inside a single FEC frame, whereas, in the latter, shaping and FEC blocklengths are the same. The first observation from Fig. 5.5 is that when N increased from 162 to 486, the performance of ESS improves by less than 0.1 dB. This is because, for sphere shaping, the shaping gain already approaches the maximum at relatively small N, and further increasing it provides only marginal improvement due to diminishing returns. Secondly, we see that the same increase in blocklength improves the performance of CCDM by more than 0.5 dB. This is because CCDM performs poorly for relatively small N, and it requires larger blocklengths than sphere shaping to have its shaping gain approach the maximum. For the AWGN channel, the performance of CCDM will converge to that of sphere shaping as $N \to \infty$.

5.5.2.3 Transmission Rate Granularity with ESS

One important advantage of PAS is that the rate adaptation is handled in the amplitude shaping block. This way, instead of having many different strong FEC codes with different coding rates to obtain rate granularity, one can fix the FEC code, and adapt the transmission rate by modifying the outer shaping code. For sphere shaping, the transmission rate can be changed

[6] As will be discussed in Chapter 7, LA1 is at least as energy efficient as ESS. Thus, for the same set of parameters, we do not expect its SNR efficiency in the PAS framework to be worse than that of ESS.

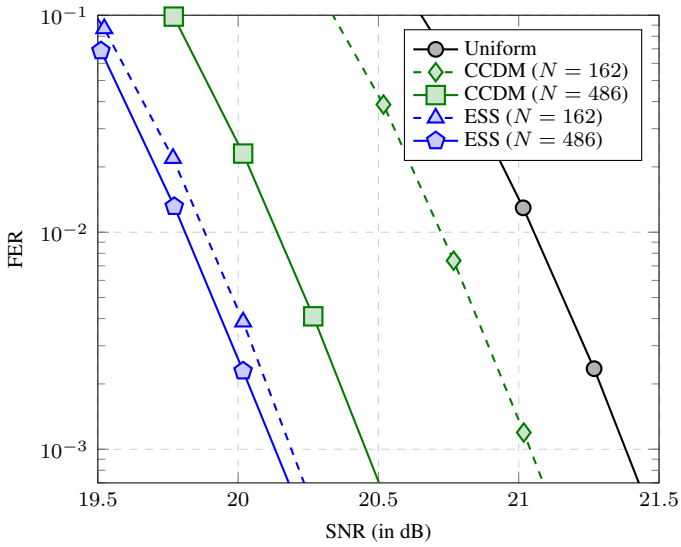

Figure 5.5: 1944-bit LDPC-coded FER vs. SNR with 16-ASK at $R_\mathrm{t} = 3$ bit/1-D.

easily by changing $E^\bullet$, and consequently, k/N. The granularity of this rate adaptation is $1/N$ which is the best possible.

In Fig. 5.6, FER is plotted versus SNR for ESS-shaped 8-ASK (top) and 16-ASK (bottom) at various transmission rates. As the FEC code, the 648-bit rate-$R_\mathrm{c} = 5/6$ LDPC code from the IEEE 802.11 standard is used. For 8-ASK transmission, $N = 216$, while for 16-ASK transmission, $N = 162$. The performance of uniform signaling with FEC codes of rate $\{1/2, 2/3, 3/4, 5/6\}$ is also shown (black). For comparison, ESS-shaped signaling with 16-ASK at $R_\mathrm{t} = 2$ bit/1-D is also shown (dashed) in the top figure. We see that by fixing the FEC code with a coding rate of $R_\mathrm{c} = 5/6$, and only using two constellations, it is possible to operate at a large interval of transmission rates, i.e., $R_\mathrm{t} < 3.33$ bit/1-D, only by changing the parameters of the amplitude shaper.

Remark 5.6 (Performance prediction for shaped signaling). Consider a uniform signaling strategy that combines 2^m-ASK with rate-R_c FEC coding. The corresponding signal structure has the shape of an N-D cube. Now consider PAS with ESS using the same constellation and FEC code, with average energy E_av. The corresponding signal structure can be thought of as the same N-D cube, with corners removed such that it reduces to an N-sphere, enclosing a smaller amount of signal points, and hence, having a smaller rate. Consequently, shaping does not change the "distance profile" of the signal set which is determined by the FEC code but rather leaves out high-energy signal points. We define the reduction in average energy

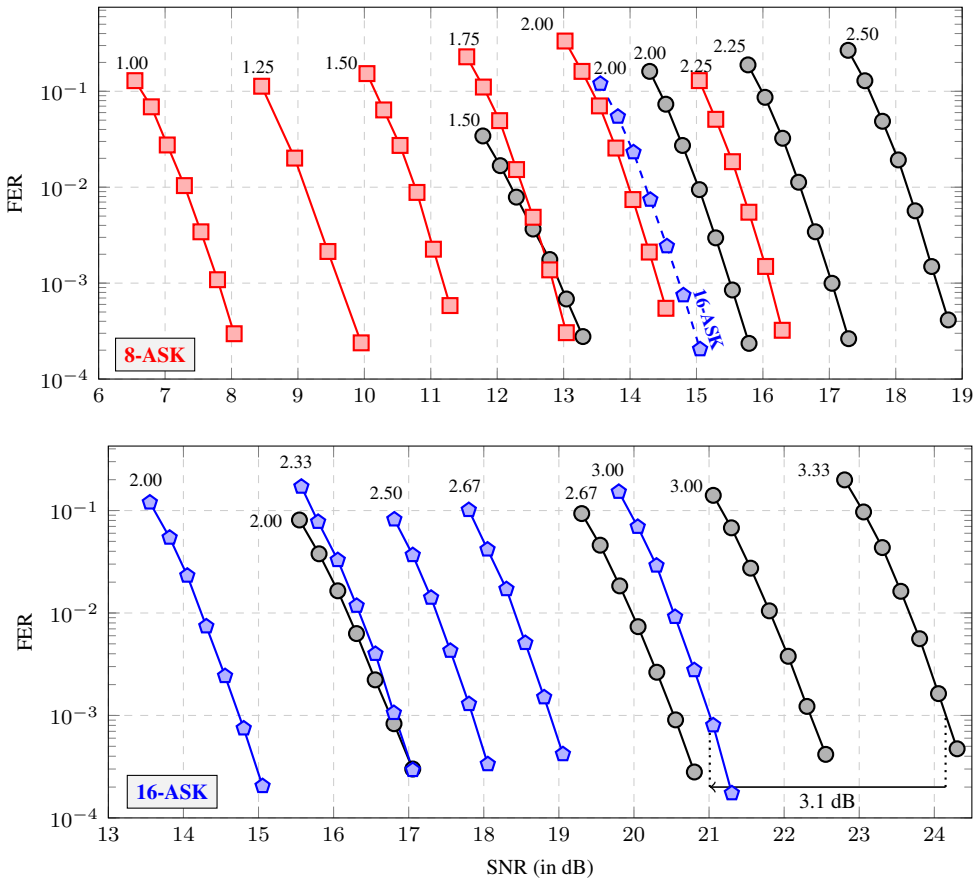

Figure 5.6: 648-bit rate-$R_c = 5/6$ LDPC-coded FER vs. SNR with 8-ASK (top, red) and with 16-ASK (bottom, blue) at indicated transmission rates (in bit/1-D). Uniform transmission with code rates $R_c \in \{1/2, 2/3, 3/4, 5/6\}$ are shown in black in both figures.

due to shaping with respect to uniform signaling (in dB) as

$$\Delta E_{\text{av}} = 10 \log_{10} \left(\frac{2^{2m} - 1}{3} \frac{1}{E_{\text{av}}} \right), \tag{5.26}$$

and we expect the FER vs. SNR behavior of the shaped scheme to be roughly the behavior of the unshaped scheme, shifted towards left by ΔE_{av} dB. As an example, consider ESS of 16-ASK with rate-$R_c = 5/6$ FEC code at rate $R_t = 3$ bit/1-D shown in Fig. 5.6 (bottom). Here $\Delta E_{\text{av}} = 3.31$ dB, and the improvement in performance with respect to the uniform

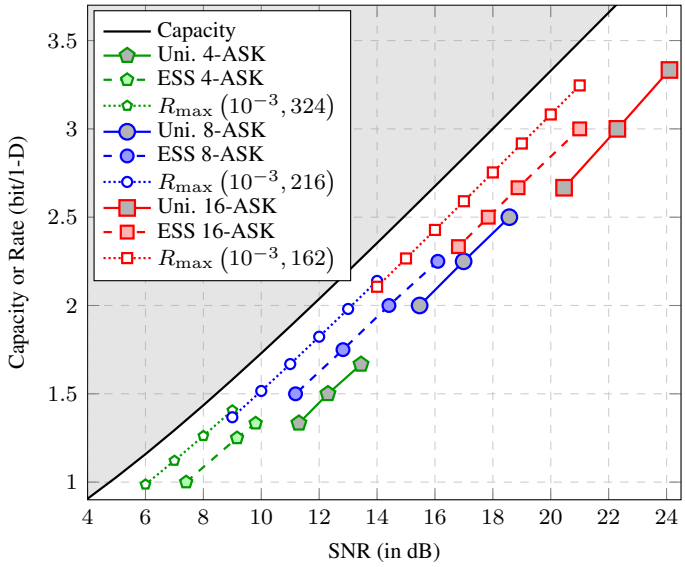

Figure 5.7: SNR values at which the FER of 10^{-3} is achieved by ESS. Shaped 8- and 16-ASK are combined with the $R_{\mathrm{c}} = 5/6$ LDPC code. Shaped 4-ASK is followed by the $R_{\mathrm{c}} = 3/4$ code. The codes are of length $n_{\mathrm{c}} = 648$. This corresponds to $N = 324$, $N = 216$, and $N = 162$ for 4-, 8-, and 16-ASK, respectively. Values for uniform signaling are also shown for $R_{\mathrm{c}} \in \{2/3, 3/4, 5/6\}$.

scheme is 3.1 dB, which are roughly in agreement.

Finally, we see from Fig. 5.6 (top) that at $R_{\mathrm{t}} = 2$ bit/1-D, ESS of 8-ASK performs better than that of 16-ASK. This observation can be explained considering the difference in N, in ΔE_{av}, and coding gains of the same FEC code combined with different constellations. This explanation is beyond the scope of this thesis, but as a thumb rule, we propose to use the smallest possible constellation which is large enough to transmit at a given rate.

5.5.2.4 SNR Gap to Polyanskiy's Approximation

In Fig. 5.7, we show the SNR values at which an FER of 10^{-3} is obtained over the AWGN channel by ESS of 4-, 8-, and 16-ASK at different transmission rates. ESS of 16- and 8-ASK employs the rate-5/6 code whereas ESS of 4-ASK is combined with the rate-3/4 code. We note that $n_{\mathrm{c}} = 648$ corresponds to $N = 324$, 216, and 162 for 4-, 8-, and 16-ASK, respectively. In the same figure, we also present the SNR values for uniform signaling with different FEC code rates. For comparison, we show the normal approximation to the maximal achievable rate (2.5) for the AWGN channel at finite blocklengths, i.e., $N \in \{162, 216, 324\}$,

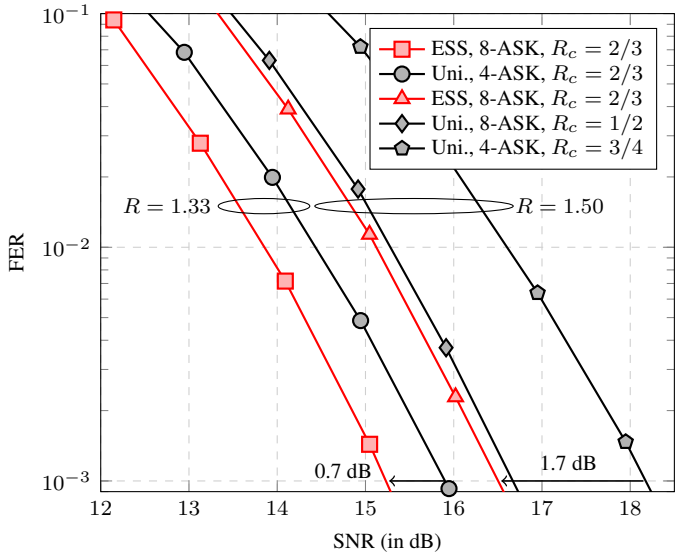

Figure 5.8: FER versus SNR behavior of ESS and uniform signaling over the HiperLAN/2-D channel. LDPC codes of length $n_c = 1296$ are employed. Shaping is over one OFDM symbol, i.e., $N = 216$.

which was derived by Polyanskiy *et al.* in [37]. We observe that for rates $R_t \in [1, 3]$ bit/1-D, it is possible to operate less than 1.5 dB away from this approximation. For instance, at $R_t = 2.67$ bit/1-D, ESS performs 1.58 dB more efficiently than uniform signaling, and it is 1.4 dB away from the approximation at $N = 162$. Figure 5.7 can be applied to predict the performance of an ESS-shaped scheme at a given rate which may help the upper layers in a communication system selecting R_t depending on the channel conditions. It becomes unnecessary to apply FEC codes of different rates to provide rate granularity by moving this functionality to the shaping block and tuning $E^\bullet$.

5.5.3 Frequency-selective Channels

In Fig. 5.8, FER is plotted versus SNR for PAS and uniform signaling for the fading channel modeled by type-D HiperLAN/2 [104]. PAS uses 8-ASK whereas the uniform scheme uses both 4- and 8-ASK. Length $n_c = 1296$-bit LDPC code is employed from the IEEE 802.11. We argued in Sec. 3.6 that as the channel starts to have a fading nature, the coding redundancy should be increased relative to the shaping redundancy. Therefore, we only used the smallest possible FEC code rate for the PAS scheme based on 8-ASK which is $R_c = 2/3$.

We see from Fig. 5.8 that at rate $R_t = 1.5$ bit/1-D, PAS with ESS is 0.2 dB more efficient

than uniform 8-ASK with $R_c = 1/2$. Furthermore, it outperforms uniform 4-ASK with $R_c = 3/4$ by 1.7 dB. At rate $R_t = 1.33$ bit/1-D, PAS with ESS requires 0.7 dB less SNR than uniform 4-ASK with $R_c = 2/3$. We note that the IEEE 802.11 standard does not provide any modulation order - coding rate combination that leads to $R_t = 1.33$ [16]. The increase in gain here as R_t decreases is because the FEC code rate R_c of the corresponding uniform setting also increases from $1/2$ to $2/3$, which degrades its performance with respect to the shaped scheme. From Fig. 5.8, we can conclude that shaping provides gains in fading scenarios as well.

Remark 5.7 (Allowed m - R_c combinations in the IEEE 802.11 standard). In Fig. 5.8, we provide FERs of two different uniform signaling settings at $R_t = 1.5$. Among these, ESS provides 1.7 dB gain over the rate-3/4-coded uniform 4-ASK which is a combination that is supported by the IEEE 802.11 standard [16]. On the other hand, the gain is 0.2 dB over the rate-1/2-coded uniform 8-ASK which is a combination that the IEEE 802.11 standard does not allow, but is simulated in this work for the sake of fairness.

5.6 Conclusion

In this chapter, we searched for an answer to the following research question.

> **RQ-3** How can sphere shaping be realized algorithmically? Which algorithm provides high performance with low complexity? What is the end-to-end decoding performance of PAS using sphere shaping over the AWGN and frequency selective channels?

We explained enumerative sphere shaping (ESS), an efficient algorithm to realize sphere shaping, and we compared ESS with two competitive algorithms: LA1 [30, Algorithm 1] and SM. We demonstrated that SM is significantly more complex than ESS, while LA1 is slightly more complex. Then we demonstrated using end-to-end decoding results that PAS with ESS provides more than 1 dB gains in power-efficiency over uniform signaling for the AWGN channel for a large range of transmission rates. Furthermore, if the shaping redundancy is kept limited, it is also possible to obtain gains up to 0.7 dB using PAS with ESS for frequency-selective channels.

CHAPTER 6

Case Study: Shaping for the IEEE 802.11 Standard

Parts of this chapter are published in:

Y. C. Gültekin, W. J. van Houtum, S. Şerbetli, and F. M. J. Willems, "Constellation shaping for IEEE 802.11," in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Commun. (PIMRC),* Montreal, QC, Canada, Oct. 2017.

Y. C. Gültekin, W. J. van Houtum, A. G. C. Koppelaar, and F. M. J. Willems, "Enumerative sphere shaping for wireless communications with short packets," *IEEE Trans. Wireless Commun.,* vol. 19, no. 2, pp. 1098-1112, Feb. 2020.

Y. C. Gültekin, F. M. J. Willems, W. J. van Houtum, and S. Şerbetli, "Encoder input selector," U.S. Patent 10 530 630 B2, Jan., 7, 2020.

6.1 Introduction

In this chapter, we will study the research question **RQ-4** which is about the use of nonsystematic FEC codes in the PAS framework. One of the prominent advantages of PAS is that it allows amplitude shaping to be included in existing communication systems as an outer code. However, the way amplitude shaping is combined with channel coding in PAS requires a systematic FEC code. In what follows, we will explain how the nonsystematic convolutional code applied in the IEEE 802.11 standard can be used in PAS, together with an outer amplitude shaping block. To the best of our knowledge, this is the only work in which a nonsystematic code is used in PAS.

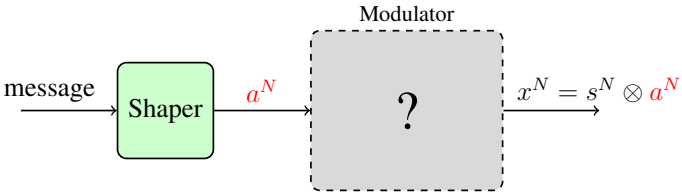

Figure 6.1: Functional diagram of the basic PAS structure explained in Sec. 2.6.3.1. The modulator must include FEC capability.

As illustrated in Fig. 6.1, the fundamental goal of PAS is to generate a channel input x^N, of which the amplitudes a^N are pre-determined by a shaping block. In the most common approach, the modulator in Fig. 6.1 consists of a systematic FEC encoder and a symbol mapper as shown in Fig. 2.6. This way, the input amplitude sequence is mapped to the unique channel input sequence in the encoder's codebook that has identical amplitudes.[1] With this way of perceiving the functionality of the modulator in Fig. 6.1, we can reformulate the problem at hand: We need to devise a method to map the input amplitude sequence to a unique channel input sequence in a nonsystematic FEC encoder's codebook. In this chapter, we explain how to achieve this for the nonsystematic convolutional codes (CCs) from the IEEE 802.11 standard [16].

6.2 PAS with a Nonsystematic FEC Code

The physical layer (PHY) of the IEEE 802.11 standard is based on BICM where interleaving is over a single orthogonal frequency-division multiplexing (OFDM) symbol, not over a FEC frame which may span multiple OFDM symbols [16].[2] Gray-coded M^2-QAM symbol map-

[1] Since symbol mapping $\{0,1\}^{mN} \to \{\pm 1, \pm 3, \cdots, \pm(2^m - 1)\}^N$ is a one-to-one function, we consider the image of the encoder's codebook under the symbol mapping function as the codebook of the encoder.

[2] The reason behind this is to be able to start Viterbi decoding as soon as the first OFDM symbol arrives, and thus, to limit the latency. Interleaving over frequency is necessary to avoid the detrimental effect resulting from adjacent coherent subcarriers being in deep fade [31, Sec. 3.3.2].

ping is used which is in the form of the Cartesian product of two M-ASK constellations. For FEC, both nonsystematic CCs and systematic LDPC codes are included. In this chapter, our focus is on the CCs.

The nonsystematic 64-state mother CC used in the IEEE 802.11 standard has rate $R_c = 1/2$, and the corresponding encoder outputs the pair $(v_1[t], v_2[t])$ at time instance t for the input bit $u[t]$. Generator polynomials of this code are $g_0 = 133_8$ and $g_1 = 171_8$. Here the subscript "8" indicates that the polynomials are expressed in octal. Examining [16, Fig. 17-8] where the structure of the corresponding encoder is shown, the output equations can be written as

$$v_1[t] = u[t] \oplus u[t-2] \oplus u[t-3] \oplus u[t-5] \oplus u[t-6], \tag{6.1}$$
$$v_2[t] = u[t] \oplus u[t-1] \oplus u[t-2] \oplus u[t-3] \oplus u[t-6] \tag{6.2}$$

where $\oplus$ denotes the XOR operation.

The finite state machine (FSM) model of this CC shows that in a given state, the output pair either belongs to the set $\{(0,0),(1,1)\}$ or to $\{(0,1),(1,0)\}$ depending on the input bit $u[t]$ of the encoder. Thus, by inverting the input bit, the output pair will also be inverted. This enables us to make half of the outputs equal to pre-determined values, i.e., the amplitude bits and the extra information bits which are used as some of the signs. We explain how this is achieved through three different cases.

6.2.1 Rate-1/2 Encoding

The only constellation that can be combined with the rate-$R_c = 1/2$ CC in the PAS framework is 4-ASK. In this case $\gamma = 0$. We consider the 4-ASK alphabet labeled with the BRGC, and we neglect the interleaver present in the IEEE 802.11 standard for now. As shown in Fig. 6.2, the output pairs of the encoder consist of a sign bit and an amplitude bit (B_1, B_2) for $t \geq 1$.[3] We aim to set half of the outputs to the prescribed amplitude bits, i.e., $V_2 = B_2$. From (6.2) , we get

$$\begin{aligned} u[t] &= v_2[t] \oplus u[t-1] \oplus u[t-2] \oplus u[t-3] \oplus u[t-6] \\ &= b_2 \oplus u[t-1] \oplus u[t-2] \oplus u[t-3] \oplus u[t-6] \\ &= f(b_2, s) \end{aligned} \tag{6.3}$$

where s and b_2 are the encoder state and the amplitude bit at time t, respectively. We call $f(\cdot)$ the *input select function*. Using this function, for each amplitude bit b_2, the *input selector* in Fig. 6.2 finds the input $u[t]$ to the convolutional encoder that will make the encoder output the prescribed amplitude bit in its corresponding position, i.e., on the v_2 branch. The other output is determined by (6.1), and it is used as the sign bit $B_1 = V_1$.

[3] In Figures 6.2, 6.3, and 6.4, $b_{i,j}$ denotes the i^{th} bit-level of the j^{th} ASK symbol x_j.

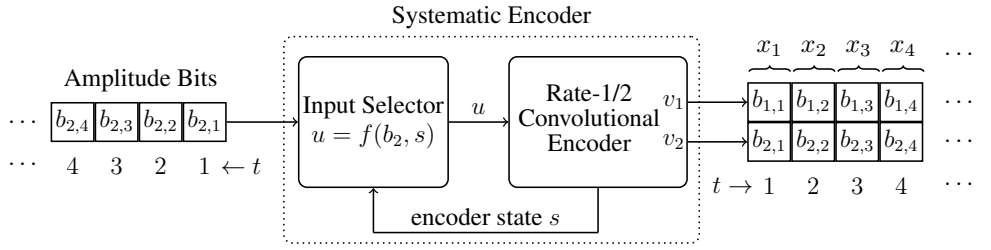

Figure 6.2: Block diagram of the PAS architecture employing the rate-1/2 nonsystematic CC of the IEEE 802.11 standard [16]. The encoder is preceded by the proposed input selector that realizes (6.3). Combined, these two operate as a systematic encoder, i.e., the input stream appears unchanged at the positions that correspond to amplitude bits at the output.

6.2.2 Rate-$(m-1)/m$ Encoding

The minimum code rate that can be combined with 2^m-ASK for $m > 2$ in the PAS framework is $R_\mathrm{c} = (m-1)/m > 1/2$. In this case, possible FEC code rates in the 802.11 standard are $R_\mathrm{c} \in \{2/3, 3/4, 5/6\}$. These code rates are obtained by puncturing the mother code, and thus, the bits that will be punctured after encoding must also be taken into account. We explain this through the following example.

Example 6.1 (Input select functions for 8-ASK and $R_\mathrm{c} = 2/3$ CC). We consider the 8-ASK alphabet labeled with the BRGC, and the CC with rate $R_\mathrm{c} = 2/3$, i.e., $\gamma = 0$. The puncturing pattern is $[1, 1, 1, 0]$. As shown in Fig. 6.3, the output pairs of the encoder consist of a sign bit and an amplitude bit (B_1, B_2) for odd time indices t. The pairs consist of an amplitude bit and a bit that will be punctured (B_3, P) for even time indices. We aim to set half of the outputs to the prescribed amplitude bits, i.e., $V_2 = B_2$ and $V_1 = B_3$ for odd and even time indices, resp. From (6.2) and (6.1), we get

$$u[t] = b_2 \oplus u[t-1] \oplus u[t-2] \oplus u[t-3] \oplus u[t-6] = f_\mathrm{o}(b_2, s) \tag{6.4}$$
$$u[t] = b_3 \oplus u[t-2] \oplus u[t-3] \oplus u[t-5] \oplus u[t-6] = f_\mathrm{e}(b_3, s) \tag{6.5}$$

for odd and even t, resp. We call $f_\mathrm{o}(\cdot)$ and $f_\mathrm{e}(\cdot)$ the odd and even input select functions, resp. Using these functions, for each amplitude bit b_2 or b_3, the input selector in Fig. 6.3 finds the input $u[t]$ to the convolutional encoder that will make the encoder output the prescribed amplitude bits in their corresponding positions, i.e., on the v_2 branch for odd t, on the v_1 branch for even t. The other output is determined either by (6.1) or by (6.2), and it is used as the sign bit $B_1 = V_1$ or punctured for odd and even time indices, resp.

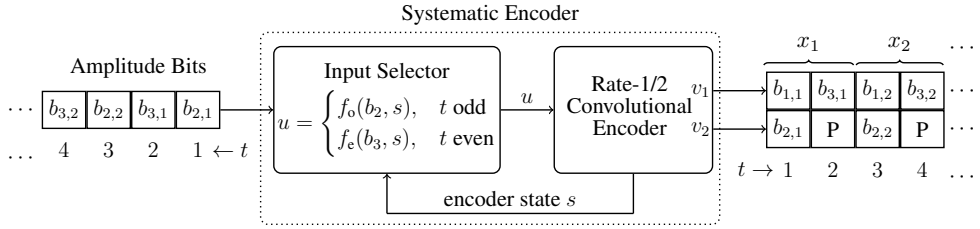

Figure 6.3: Block diagram of the PAS architecture employing 8-ASK and the rate-2/3 nonsystematic CC of the IEEE 802.11 standard [16]. The encoder is preceded by the proposed input selector that realizes (6.4) and (6.5). Combined, these two operate as a systematic encoder, i.e., the input stream appears unchanged at the positions that correspond to amplitude bits at the output.

6.2.3 Rate-$(m-1+\gamma)/m$ Encoding

When a rate $R_\mathrm{c} > (m-1)/m$ FEC code is combined with 2^m-ASK, i.e., $\gamma > 0$, some of the signs are selected directly by information bits. In this case, the increase in the number of punctured bits (with respect to the rate $R_\mathrm{c} = (m-1)/m$ code) should be used to make these sign bits appear at the output of the FEC encoder as well. We explain this through the following example.

Example 6.2 (Input select functions for 8-ASK and $R_\mathrm{c} = 5/6$ CC). We consider the 8-ASK alphabet labeled with the BRGC, and the CC with rate $R_\mathrm{c} = 5/6$, i.e., $\gamma = 1/2$. The puncturing pattern is $[0,0,0,1,1,0,0,1,1,0]$. As shown in Fig. 6.4, the output pairs of the encoder always consist of a bit that we need to set to a pre-determined value (amplitude bits B_2 or B_3, or sign bits B_1 for $t = 3, 8, \cdots$ that are equal to information bits), and a bit that we do not care about (a bit that will be punctured P, or sign bits B_1 for $t = 1, 6, \cdots$ that are parity). Again using (6.4) and (6.5) for each amplitude bit b_2 and b_3, and for each (information) sign bit b_1, the input selector in Fig. 6.4 finds the input $u[t]$ to the convolutional encoder that will make the encoder output the prescribed amplitude bits or (information) sign bits in their corresponding positions. The other output is determined either by (6.1) or by (6.2), and used as the (parity) sign bit $B_1 = V_1$ for $t = 1, 6, \cdots$ or it is punctured.

Remark 6.1 (On the universality of input-select and input-deselect). For other combinations of m and R_c, the puncturing pattern and the positions of the amplitude bits and (information) sign bits at the encoder output may change. For such settings, the input select functions can be modified straightforwardly. For some combinations, it may be necessary to consider not only the current input bit and the encoder state but also the next ones. For some other combinations, it may also be necessary to feed the shift register in [16, Fig. 17-8] from the opposite direction. However, it is possible to obtain input-select functions for all combinations of modulation formats and coding rates (the ones which are possible to combine

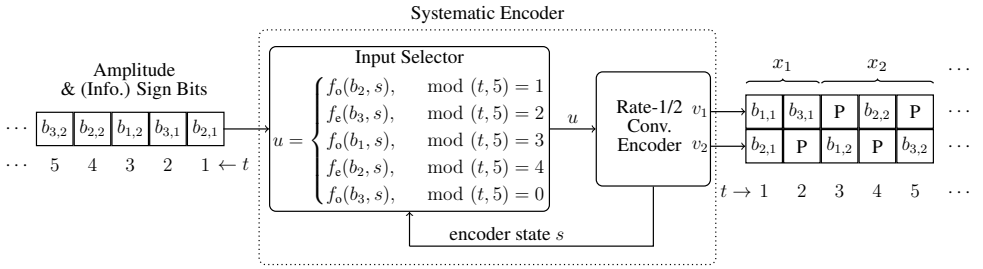

Figure 6.4: Block diagram of the PAS architecture employing 8-ASK and the rate-5/6 non-systematic CC of the IEEE 802.11 standard [16]. The encoder is preceded by the proposed input selector that realizes (6.4) and (6.5). Combined, these two operate as a systematic encoder, i.e., the input stream appears unchanged at the positions that correspond to amplitude bits and (information) sign bits at the output.

in PAS) specified in the IEEE 802.11 standard. The block that realizes the inverse function of the input selector at the receiver can be implemented similarly. This block, given that the preceding FEC decoder correctly estimated the complete frame, does not introduce any errors.

Remark 6.2 (**On the availability of encoder state s in the input selector).** We note that since the FSM model of the encoder and its starting state (zero) are known to the input selector, the feedback links that carry the encoder state s in Figures 6.2, 6.3 and 6.4 are unnecessary in practice, and they are only included here to emphasize that the selection process depends on s.

Remark 6.3 (**On the definition of systematic codes).** A systematic, rate k/n convolutional encoder is usually defined to have k of its n output branches reserved for systematic bits as in [106, Sec. 2.10], [107, Sec. 11.1], and [108, Sec. 8.1.9]. Since in our proposal, the systematic bits may appear in different branches during encoding, the effective systematic encoders in Figures 6.2, 6.3 and 6.4 differ from the common systematic encoders in the literature in general.

6.2.4 Effect of Interleaving

The bit-level interleaver present in 802.11 permutes coded bits in a pre-determined way before the symbol mapping. Due to the deterministic nature of the interleaver, we can determine which output bits of the encoder will be used by the following mapper as amplitude and sign bits. As an example in Fig. 6.5, whether the bits at the output of the encoder are amplitude or sign bits, and the index of the channel inputs that they belong are shown for 8-ASK and $R_c = 2/3$. In this case, the bits at the input of the input selector should be shuffled according to the pre-interleaver order of the bits shown in Fig. 6.5. We note here that the specific

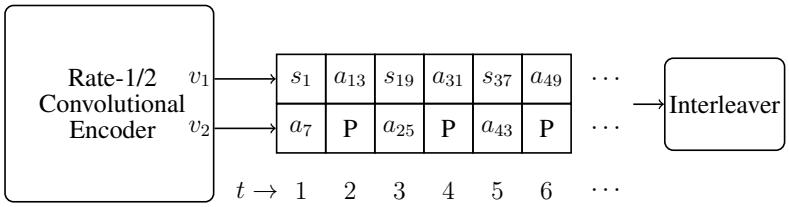

Figure 6.5: The functions of the bits at the output of the encoder which will be punctured to rate $R_c = 2/3$, interleaved, and mapped onto 8-ASK symbols. Here, a_j and s_j represent amplitude bits and sign bits of the j^{th} QAM symbol by the mapper (after puncturing and interleaving).

interleaver defined in [16, Sec. 17.3.5.7] permutes the coded bits in a way that their function, i.e., amplitude or sign bit, does not change, while adjacent coded bits are mapped to non-adjacent subcarriers as shown in Fig. 6.5. With another type of interleaver which leads to a pre-interleaver coded bit order that has long runs of a certain bit-level, deriving input-select functions would not be possible.

6.2.5 Effect of Code Termination

In the IEEE 802.11 standard, trellis termination is realized during convolutional encoding by appending 6 zeros to the end of the encoder input to make sure that the encoder returns to state zero [16, Sec. 17.3.5.3]. This can be taken into account during input-select by allowing a negligible decrease in shaping gain. In this case, a couple of symbols (the exact number depending on the constellation size) at the end of the frame stay uniform due to zero padding which terminates the trellis.

6.3 End-to-end Decoding Performance

In this section, Monte Carlo simulation results are provided to evaluate the performance of PAS using the nonsystematic CCs of the IEEE 802.11 standard [16]. FEC codes are of rate R_c and length n_c-bits.

For amplitude shaping, ESS is combined with the input-select functions explained in Sec. 6.2. Shaping is always over an OFDM symbol, i.e., $N = 96$. For a given target transmission rate R_t, constellation size 2^m, and FEC code rate R_c, $E^\bullet$ is selected as the smallest value that satisfies $k/N + \gamma \geq R_t$. Here again, $\gamma = R_c m - (m-1)$.

For the simulations over frequency-selective channels, the bandwidth is now set to 10 MHz [16, Table 17-5], and it is separated into 64 subcarriers among which 48 are used for data, 4 occupied by pilots, and the remaining 12 are empty [16, Sec. 17.3.5.10]. All other

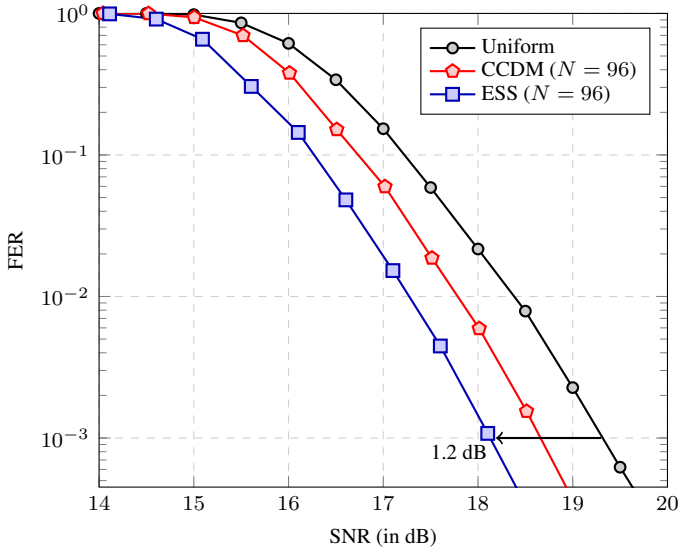

Figure 6.6: FER vs. SNR for ESS, CCDM, and uniform signaling with 8-ASK at $R_\mathrm{t} = 2.25$ bit/1-D.

parameters and settings related to simulations over fading channels are the same as that of Sec. 5.5.

Encoding with trellis termination, puncturing, and interleaving are implemented as in the IEEE 802.11 standard [16, Sec. 17.3.5]. The BRGC is applied by the symbol mapper. The same mapping is used to label the amplitudes at the output of the shaper. At the receiver side, the demapper computes LLRs using (2.8), and then Viterbi decoding is realized.

6.3.1 The AWGN Channel

In Fig. 6.6, FER is plotted versus SNR for PAS and uniform signaling with 8-ASK. The transmission rate is $R_\mathrm{t} = 2.25$ bit/1-D. For this constellation and transmission rate, the FEC code rate that minimizes ΔSNR in (3.97) is approximately $5/6$ which should be combined with the MB distribution that has $H(X) = 2.75$. Thus, ESS and CCDM are combined with the rate-5/6 code. This leads to $\gamma = R_\mathrm{c} m - (m-1) = 1/2$ with $k/N = 1.75$ bit/1-D. We take 8 shaping blocks of 96 amplitudes, i.e., 8 OFDM symbols, inside a single FEC codeword which consists of $n_\mathrm{c} = 2304$ bits. The ESS is realized with $E^\bullet = 1120$ which leads to $R_\mathrm{s} = 1.7503$ bit/1-D and $E_\mathrm{av} = 11.4264$. For this shaping scheme, rate loss $R_\mathrm{loss} = 0.0234$ bit/1-D and shaping gain $G_\mathrm{s} = 1.11$ dB. Uniform signaling is realized with the CC of rate $R_\mathrm{c} = R_\mathrm{t}/m = 3/4$ We see from Fig.6.6 that at an FER of 10^{-3}, ESS is 1.2 dB

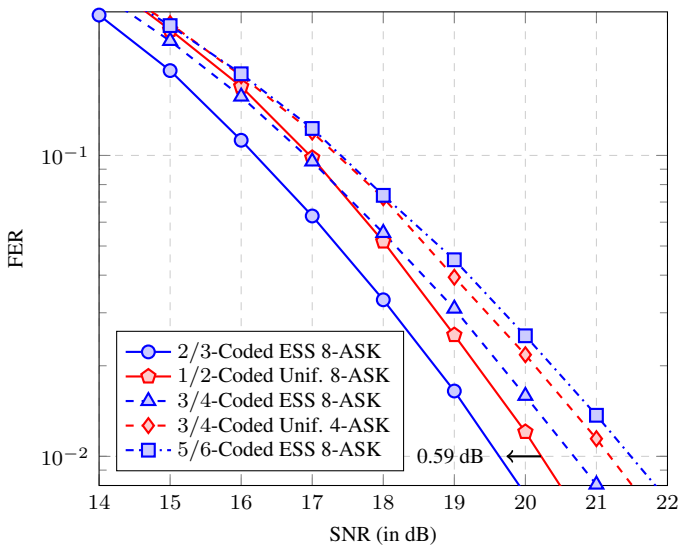

Figure 6.7: FER vs. SNR for ESS and uniform signaling at $R_\mathrm{t} = 1.5$ bit/1-D.

more power-efficient than uniform signaling. This is in rough agreement with the computed shaping gain. In this setting, ESS outperforms CCDM by 0.55 dB.

6.3.2 Frequency Selective Channels

In Fig. 6.7, the FER is plotted versus the SNR for PAS and uniform signaling for the frequency selective fading channel which is modeled by type-D HiperLAN/2 [104]. PAS uses 8-ASK whereas the uniform scheme uses both 4- and 8-ASK. CCs from IEEE 802.11 are employed where for each combination of modulation formats and coding rates, n_c is selected such that 12 OFDM symbols are filled. We see from Fig. 6.7 that rate-2/3 coded ESS is almost 0.6 dB more efficient than uniform 8-ASK with $R_\mathrm{c} = 1/2$. Furthermore, it outperforms uniform 4-ASK with $R_\mathrm{c} = 3/4$ by almost 1.5 dB. Here, we attribute the increase in gain to the increased code rate for 4-ASK which degrades the performance over fading channels. Another observation is that when the code rate increases, the efficiency of ESS is immediately lost. This is in agreement with our discussion in Sec. 3.6.2 and observations in Sec. 5.5.3 in which we stated the coding redundancy should be kept relatively high for fading channels.

6.4 Conclusion

In this chapter, we searched for an answer to the following research question.

RQ-4 Can PAS be incorporated into existing communication systems that are based on the IEEE 802.11 standard? Can PAS be combined with the nonsystematic convolutional codes used in 802.11 [16] which are a mandatory part of the standard?

We devised an input-select block that shuffles the amplitude bits generated by the shaper. When placed in between the shaper and the nonsystematic channel encoder of the IEEE 802.11 standard, this block preserves the temporal structure of the shaped amplitudes through the nonsystematic channel encoding. Thus, it enables PAS to be realized also with convolutional codes of the IEEE 802.11 (in addition to the LDPC-coded mode discussed in Chapter 5), and it completes the integration of PAS into the 802.11.

CHAPTER 7

Practical Implementation Aspects

"In theory, there is no difference between theory and practice,
while in practice, there is."

Parts of this chapter are published in:

Y. C. Gültekin, F. M. J. Willems, W. J. van Houtum, and S. Şerbetli, "Approximate enumerative sphere shaping," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, June 2018, pp. 676-680.

Y. C. Gültekin and F. M. J. Willems, "Building the optimum enumerative shaping trellis," in *Proc. Symp. on Inf. Theory and Signal Process. in the Benelux (SITB)*, Gent, Belgium, May 2019, p. 34. (Abstract & poster presentation)

Y. C. Gültekin, F. M. J. Willems, W. J. van Houtum, and S. Şerbetli, "Approximate enumerative sphere shaping," U.S. Patent 10 523 474 B1, Dec., 31, 2019.

Y. C. Gültekin, W. J. van Houtum, A. G. C. Koppelaar, and F. M. J. Willems, "Low-complexity enumerative coding techniques with applications to amplitude shaping," *IEEE. Commun. Lett.*, Sep. 2020.

Y. C. Gültekin, W. J. van Houtum, A. G. C. Koppelaar, and F. M. J. Willems, "Comparison and optimization of enumerative coding techniques for amplitude shaping," Nov. 2020 (submitted to IEEE Commun. Lett.).

7.1 Introduction

In this chapter, we will examine the research question **RQ-5** which revolves around the problem of "implementing ESS as effectively as possible". This effectiveness can be evaluated through 5 fundamental qualities that are not necessarily independent: Energy efficiency, storage complexity, computational complexity, latency, and required arithmetic precision.

In Chapter 4, we showed that at any blocklength N, sphere shaping minimizes the rate loss and maximizes the shaping gain for a fixed rate. Later in Chapters 5 and 6, we demonstrated through end-to-end decoding results that PAS with sphere shaping indeed provides significant performance improvements over uniform signaling. In Chapter 5, we provided three algorithms to realize sphere shaping: ESS, LA1, and SM. We then concluded in Sec. 5.4.4 that ESS is the efficient solution for sphere shaping due to the facts that: (1) SM is remarkably more complex than ESS since it requires a huge number of multiplications, although ESS requires larger storage, and (2) LA1 is slightly more complex than ESS due to an extra algorithmic step, although ESS is marginally less energy-efficient. As a matter of fact, in Sec. 5.4.4, we avoid discussing the dynamics of the trade-offs in (1) and (2) for the sake of clearly demonstrating the potential of sphere shaping. We left a thorough discussion on the practical effectiveness of ESS and the explanation of implementation ideas to this chapter.

In Sec. 7.2, we will first investigate the energy-efficiency. We will differentiate between the *sphere shaping set* which consists of all signal points within a sphere, and the *operational shaping set* which is a size-2^k subset that only includes the sequences which are transmitted. Then we will provide a method to compute the amplitude distribution for the operational shaping set of ESS. Using this method, we will show that ESS is marginally less energy-efficient than LA1 and SM for practical purposes. Then we will introduce an optimization routine to modify the backward trellis, such that the energy-efficiency of ESS is improved, if not maximized.

In Sec. 7.3, we will investigate the storage complexity. We will introduce a *bounded precision* (BP) implementation technique for ESS, LA1, and SM where the numbers in their corresponding trellises are represented with a fixed number of bits, resulting in a significant reduction in the required storage to realize these algorithms.[1] We will show that even if they are realized with this BP technique, these algorithms keep working properly and suffer only from a negligible rate loss. Consequently, storage complexity will not be a dominant factor influencing the choice among ESS, LA1, and SM.

In Sec. 7.4, we will investigate computational complexity, latency, and required arithmetic precision. We will introduce sliding window shaping (SWS) which is enabled by the BP precision implementation. With SWS, ESS and LA1 can be implemented with short, fixed-length arithmetic operations, where the part of the input index on which these operations are realized shifts from the most significant bit (MSB) to the least significant bit (LSB) gradually. Consequently, for ESS and LA1, (1) the computational load decreases, (2) shaper/deshaper blocks can start outputting symbols as soon as the shaping/deshaping procedures start, which limits

[1] A similar bounded precision implementation approach was considered for constrained coding in [109].

the latency, and (3) the required arithmetic precision becomes both fixed and independent of the blocklength. We stress that SWS is an extension of the BP implementation, and thus, these improvements can at least partially be attributed to the BP technique.

Finally in Sec. 7.5, we will introduce the *on-the-fly* (OtF) trellis computation technique which further decreases the required storage to realize ESS and LA1 at the expense of increased computational complexity. By storing a single column from the trellis (and a few bits per node for the rest of the trellis in the BP case), we will demonstrate how to compute the remaining columns OtF.

7.2 ESS Optimized for Binary Transmission

A sphere shaping function, i.e., algorithm, creates a one-to-one mapping

$$f : \{0, 1, \cdots, |\mathcal{A}^\bullet| - 1\} \to a^N \in \mathcal{A}^\bullet = \left\{ a^N \middle| e\left(a^N\right) \leq E^\bullet \right\}. \tag{7.1}$$

Consequently, the only objective of f is to create an *ordered* list of $a^N \in \mathcal{A}^\bullet$. As discussed earlier in Sec. 5.4.1, sphere shaping algorithms are used to map k-bit information strings to amplitude sequences in the PAS framework where $k = \lfloor \log_2 |\mathcal{A}^\bullet| \rfloor$. Thus, only the first $2^k = K$ sequences in the corresponding ordered list are subject to transmission, i.e., $\mathcal{E} = f(\{0, 1, \cdots, K-1\}) \subseteq \mathcal{A}^\bullet$. We call this set the *operational shaping set*. Although the complete N-sphere considered by different sphere shaping algorithms is the same, they might have different operational sets due to algorithmic differences, e.g., in the way they order sequences as shown in Table 5.1. Since shaping functions f_{LA1} and f_{SM} of LA1 and SM, respectively, order sequences based on their energy, their operational shaping sets have the minimum possible average sequence energy $E_{\min} \leq E_{\text{av}}$. However, shaping function f_{ESS} of ESS orders sequences lexicographically, and its operational shaping set has average energy $E_{\text{ESS}} \geq E_{\min}$.

In this section, we will propose a method to compute the frequencies of the amplitudes for the K sequences $a^N \in \mathcal{E}$ considered by ESS. This way, the exact channel input distribution can be computed, which is required (1) for an accurate calculation of the transmit power, and (2) for a precise likelihood computation by the demapper at the receiver, see (2.8). A sketch of this method was provided by [110], and it applies directly for LA1 [30, Algorithm 1]. We note that in [111], a method to compute amplitude frequencies was proposed for SM. Next, we will demonstrate that ESS is slightly less energy-efficient than SM and LA1 for ultra-short blocklengths ($N < 20$). However, we note that ESS has smaller complexity than LA1 and SM as we discussed in Sec. 5.4. Then we will introduce a heuristic optimization technique for backward trellises such that the operational energy efficiency of ESS is improved, or maximized. Finally, we will demonstrate using end-to-end decoding results that with this optimization, ESS provides similar performance as SM and LA1 for the AWGN channel for ultra-short blocklengths.

7.2.1 Computing the Operational Amplitude Distribution

The average distribution $p(a)$ for the sequences in the *complete* sphere shaping set $\mathcal{A}^{\bullet}$ can be computed using $T_1^{a^2}$ for $a \in \mathcal{A}$ as discussed in Lemma. 5.1. To compute the distribution for the sequences in the operational shaping set, consider the sequence $z^N = (z_1, z_2, \cdots, z_N) \in \mathcal{A}^{\bullet}$ which has index $f_{\text{ESS}}^{-1}(z^N) = K$. This sequence is the first sequence on the lexicographical list that is not in the operational shaping set $\mathcal{E}$. Then $\mathcal{E} = f_{\text{ESS}}(\{0, 1, \cdots, K-1\})$ can also be defined as

$$\begin{aligned} \mathcal{E} &\triangleq \left\{x^N = (x_1, x_2, \cdots, x_N) : f_{\text{ESS}}^{-1}(x^N) < K\right\} \\ \mathcal{E} &= \bigcup_{n=1}^{N} \bigcup_{a<z_n} \mathcal{E}_n(a) \end{aligned} \tag{7.2}$$

for $a \in \mathcal{A}$ where

$$\mathcal{E}_n(a) \triangleq \{x^N : (x_1, x_2 \cdots, x_{n-1}) = (z_1, z_2 \cdots, z_{n-1}), x_n = a\} \tag{7.3}$$

for $a < z_n$. Intuitively, $\mathcal{E}_n(a)$ is the set of sequences which have their first $n-1$ elements identical to z^N and which are lexicographically smaller than z^N.

Now, we can write the number of times the amplitude $a \in \mathcal{A}$ occurs in the j^{th} position of the sequences in the operational shaping set $\mathcal{E}$ as

$$\#_j(a|\mathcal{E}) = \sum_{n=1}^{N} \sum_{x<z_n} \#_j(a|\mathcal{E}_n(x)) \tag{7.4}$$

$$\begin{aligned} &= \sum_{n=1}^{j-1} \sum_{x<z_n} \#_j(a|\mathcal{E}_n(x)) + \sum_{n=j}^{N} \sum_{x<z_n} \#_j(a|\mathcal{E}_n(x)) \\ &= \sum_{n=1}^{j-1} \sum_{x<z_n} T_{n+1}^{a^2+x^2+\sum_{i=1}^{n-1} z_i^2} \\ &\quad + \sum_{x<z_j} \mathbb{1}[x=a] T_j^{x^2+\sum_{i=1}^{j-1} z_i^2} \\ &\quad + \sum_{n=j+1}^{N} \sum_{x<z_n} \mathbb{1}[z_j=a] T_n^{x^2+\sum_{i=1}^{n-1} z_i^2} \end{aligned} \tag{7.5}$$

where $x \in \mathcal{A}$, (7.4) follows from the partitioning of the operational shaping set in (7.2), and (7.5) follows from Lemma 5.1. Then the distribution $p_n^{\text{op}}(a)$ averaged over the sequences in the operational shaping set $\mathcal{E}$ and over the n^{th} position is

$$p_n^{\text{op}}(a) = \frac{\#_n(a|\mathcal{E})}{K} \tag{7.6}$$

for $a \in \mathcal{A}$ and for $1 \le n \le N$.

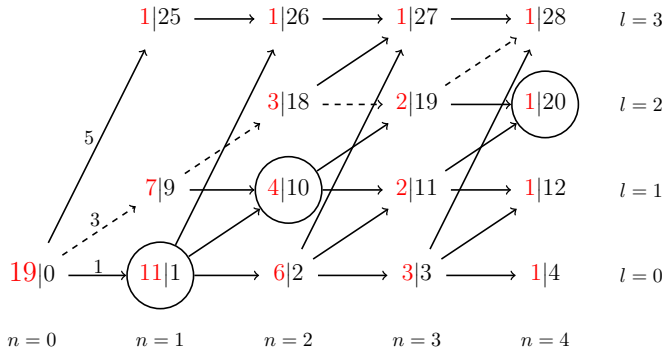

Figure 7.1: Enumerative (backward) trellis for $N = 4$, $\mathcal{A} = \{1, 3, 5, 7\}$, and $E^{\bullet} = 28$.

Example 7.1. Consider the trellis in Fig. 7.1 for which $f_{\text{ESS}}(K) = z^N = (3, 3, 1, 3)$, i.e., the path which is shown with dashed lines, see also $a^N(K)$ in Table 5.1 for $K = 2^4 = 16$. We can write $\mathcal{E} = \mathcal{E}_1(1) \cup \mathcal{E}_2(1) \cup \mathcal{E}_4(1)$ using (7.2). The sequences that are included in $\mathcal{E}_1(1)$, $\mathcal{E}_2(1)$, and $\mathcal{E}_4(1)$ branch off from z^N to nodes $(1, 1)$, $(2, 10)$ and $(4, 20)$, respectively, which are drawn with circles in Fig. 7.1.

We now consider $\#_2(a|\mathcal{E})$, i.e., the frequency of amplitude a for the second position. The contribution of $x^N \in \mathcal{E}_1(1)$ is considered in the first line of (7.5). These sequences branched off to a lower node than that of z^N at position $n = 1 < j = 2$ in the trellis, i.e., all permutations of (x_2, x_3, x_4) can occur after position $n = 1$ satisfying energy constraint $E^{\bullet}$. Thus, their contribution for the position $j = 2 > n = 1$ can be computed using the symmetry discussed in Lemma 5.1. Then for $\mathcal{E}_1(1)$, the frequencies of $a \in \{1, 3, 5\}$ for the position $j = 2$ are given by $T_2^{a^2+1}$ which are $(6, 4, 1)$ as shown in Table 7.1.

The contribution of $x^N \in \mathcal{E}_2(1)$ is considered in the middle line of (7.5). These sequences differ from z^N for the first time at position $n = 2 = j$, i.e., $x_j < z_j$, and thus, they only contribute to the amplitudes $a < z_j$ at position j. Then for $\mathcal{E}_2(1)$, the frequency of $a = 1$ for the position $j = 2$ is given by $T_2^{a^2+9} = T_2^{10}$ which is 4 as shown in Table 7.1.

The contribution of the sequences in $\mathcal{E}_4(1)$ is considered in the third line of (7.5). These sequences differ from z^N for the first time at position $n = 3 > j = 2$, i.e., they have $x_j = z_j$, and thus, they can only contribute to the amplitude $a = z_j$ at the position j. Then for $\mathcal{E}_4(1)$, the frequency of $a = 3$ for the position $j = 2$ is given by $T_4^{1+19} = T_4^{20}$ which is 1 as shown in Table 7.1.

Using (7.5), the frequencies and distributions of the amplitudes are computed and tabulated in Table 7.1. We see that (1) $p^{\text{op}}(a)$ differs from $p(a)$, and (2) $p_j^{\text{op}}(a)$ depends on j unlike $p_j(a)$ of the sphere shaping set, see Lemma 5.1.

Remark 7.1 (Time-variant (time-var.) soft demapping for shaped signaling). To compute LLRs, we assumed in (2.8) that the channel inputs are i.i.d for each channel use. As

Table 7.1: Amplitude Frequencies and Distributions

	$\#_n(1\|\mathcal{E})$	$\#_n(3\|\mathcal{E})$	$\#_n(5\|\mathcal{E})$	$p_n^{\text{op}}(a)$
$n=1$	11	4+1	0	$(11/16, 5/16, 0/16)$
$n=2$	6+4	4+1	1	$(10/16, 5/16, 1/16)$
$n=3$	6+2+1	4+2	1	$(9/16, 6/16, 1/16)$
$n=4$	6+2+1	4+2	1	$(9/16, 6/16, 1/16)$
$p^{\text{op}}(a)$	39/64	22/64	3/64	$p^{\text{op}}(a) = \sum_n p_n^{\text{op}}(a)/N$

an alternative approach, at each time n, the corresponding distribution p_n can be used for demapping. In this case, (2.8) should be modified to

$$L_{j,n} = \log \frac{\sum_{x \in \mathcal{X}_{j,0}} p_n^{\text{op}}(x) p(y_n|x)}{\sum_{x \in \mathcal{X}_{j,1}} p_n^{\text{op}}(x) p(y_n|x)} \tag{7.7}$$

where $p_n^{\text{op}}(x) = p(s) p_n^{\text{op}}(a)$ for uniform $p(s)$.

7.2.2 Comparison

In Fig. 7.2, the average energy E_{ESS} of the sequences in the operational shaping set $\mathcal{E}$ of ESS is shown for $k/N = 1.5$ bit/1-D and $\mathcal{A} = \{1, 3, 5, 7\}$. At each value of N, we choose the minimum $E^\bullet$ for which the corresponding trellis has at least 2^k sequences. In the same figure, the average energy E_{av} of the sequences in the corresponding sphere shaping set $\mathcal{A}^\bullet$, and the minimum average energy $E_{\min}$ that is obtained by f_{LA1} and f_{SM} are also shown.

We see from Fig. 7.2 that the average energy E_{av} of the sphere shaping set can be used as an approximation for the average energy E_{ESS} of the operational shaping set of ESS. This approximation is extremely accurate for $N > 20$. The reason behind this accuracy is that usually, only a small fraction of sequences are left out, i.e., $|\mathcal{E}| \approx |\mathcal{A}^\bullet|$. As an example at $N = 28$, among the $2^{42.0144}$ sequences in the sphere shaping set, 2^{42} sequences are transmitted, i.e., less than 1% of the sequences are unused. Another important observation is that E_{ESS} is never higher than E_{av} for this setting. This hints that the lexicographical ordering rarely creates operational shaping sets where sequences with low energy are left out.

We also computed the KL divergence of $p^{\text{op}}(a)$ from $p(a)$ of ESS found using (7.5) for the set of parameters considered in Fig. 7.2. At all blocklengths, the divergence is smaller than 0.001 bits. Therefore, $p(a)$ is a very accurate approximation for $p^{\text{op}}(a)$ of ESS.

Finally, we see from Fig. 7.2 that the average energy E_{ESS} of ESS is higher than that $E_{\min}$ of LA1 and SM. As an example, this difference is 0.08 dB at $N = 18$. Furthermore, we see that this difference decreases with increasing N as an overall trend. Therefore, we conclude that ESS is slightly less energy-efficient than energy-based ordering methods, especially for ultra-small N. However, ESS has a slightly smaller complexity than LA1 with comparable

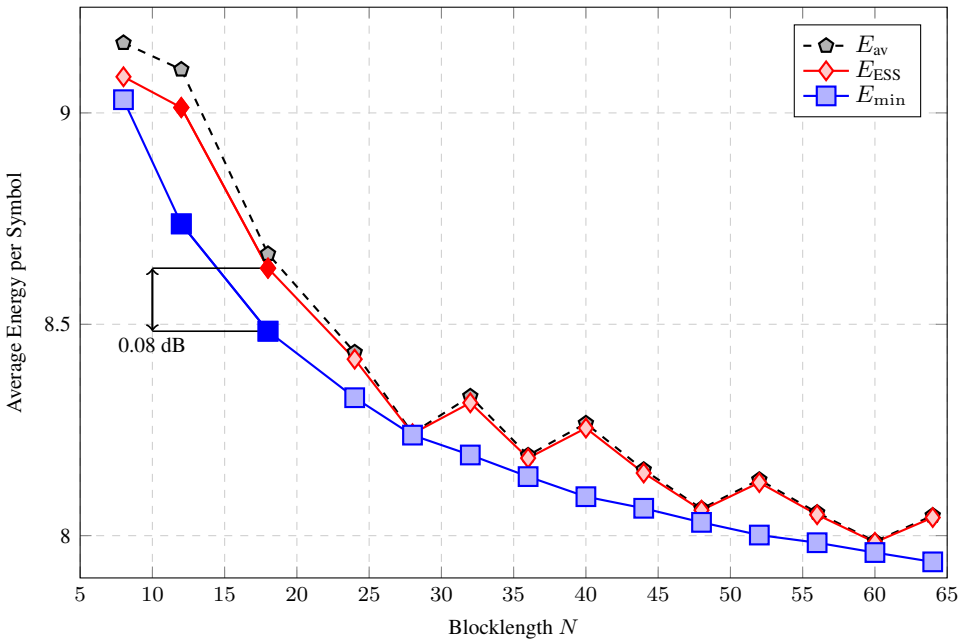

Figure 7.2: Energy efficiency of ESS and energy-based ordering methods. The minimum possible is 7.54, obtained with a sampled Gaussian distribution.

storage requirements, and has much smaller complexity than SM with higher storage requirements as discussed in Sec. 5.4. In the next section, we will close the energy efficiency gap between ESS and these algorithms.

7.2.3 Energy-optimum ESS

In Sec. 7.2.2, we observed that although slightly, ESS is less energy efficient than SM and LA1 for ultra-small N. In this section, we will introduce a method to modify the backward trellis in a rather heuristic manner such that the resulting operational shaping set will have average energy very close to the minimum, i.e., $E_{\text{ESS}} \approx E_{\min}$, if not exactly minimum. A sketch of this method was provided by [110].

To explain this modification method, we consider the trellis computed for $N = 4$, $\mathcal{A} = \{1, 3, 5, 7\}$, and $E^{\bullet} = 60$ as shown in Fig. 7.3 (top). There are $T_0^0 = 82$ sequences represented in this trellis, and consequently, $k = \lfloor \log_2 82 \rfloor = 6$ bits. Average energy of the sphere shaping set (82 sequences) per symbol $E_{\text{av}} = 10.8537$, while the average energy of the operational shaping set ($2^k = 64$ sequences) $E_{\text{ESS}} = 10.1875$. We note that in this setting, the minimum average energy $E_{\min} = 9.6875$ (0.22 dB less than that of ESS) which can be

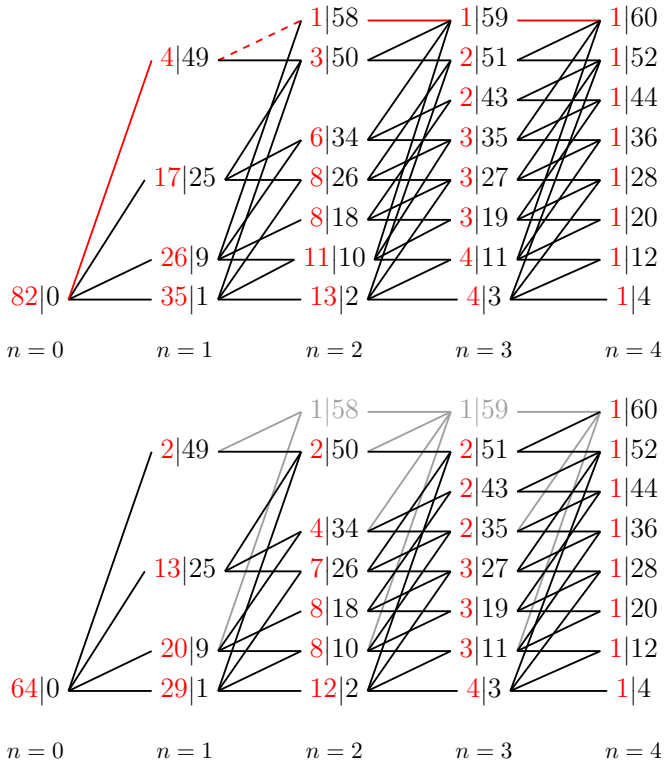

Figure 7.3: For $N = 4$, $\mathcal{A} = \{1, 3, 5, 7\}$, and $E^{\bullet} = 60$: (Top) ESS trellis. (Bottom) An optimized ESS trellis.

7

obtained using an energy-based ordering algorithm such as LA1 or SM.

The key idea here is that the paths in the trellis that correspond to sequences with the highest energy (60 in this example) include at least a single branch that arrives at a node on the topmost level of the trellis. As an example consider the path shown with red in Fig. 7.3 (top) which represents the sequence $(7, 3, 1, 1)$ with energy 60. The transition that ensures that this path will arrive at the node of energy level 60 is the one that connects the node $(1, 49)$ to $(2, 58)$ which is drawn with a dashed line. Removing this branch would effectively mean that the sequence $(7, 3, 1, 1)$ is no longer represented in the trellis, and consequently, cannot be outputted by ESS based on this modified trellis. We call branches that arrive at a node on the topmost level the *bad* branches. We denote the set of bad branches by $\mathcal{B}$ whose cardinality is at most $N|\mathcal{A}|$. To remove the highest-energy sequences from the trellis, some of the bad branches must be removed. We denote this set of removed branches by $\hat{\mathcal{B}}$ and the

resulting trellis by $\hat{T}$. The selection of $\hat{\mathcal{B}}$ must ensure that

$$\hat{T}_0^0 \geq 2^k. \tag{7.8}$$

There exists $2^{N|\mathcal{A}|} - 1$ nonempty subsets $\hat{\mathcal{B}} \subseteq \mathcal{B}$, and we propose the following heuristic to determine $\hat{\mathcal{B}}$:

$$\min_{\hat{\mathcal{B}} \subseteq \mathcal{B}} \hat{T}_0^0 \quad \text{such that} \quad \hat{T}_0^0 \geq 2^k. \tag{7.9}$$

If there are multiple $\hat{\mathcal{B}}$'s that satisfy (7.9), any one of them can be used since they all create trellises, and hence, operational shaping sets, with the same number of sequences and the same average energy. Alternatively, the corresponding operational distribution $p^{\text{op}}(a)$ can be computed for each of them, and the one that is information-theoretically the best can be selected. We note that to obtain the set of sequences with the minimum possible average energy $E_{\min}$ for a given k, (7.8) must be satisfied with equality.

Now we consider Fig. 7.3 (bottom) where we show an optimized version of the initial (complete) trellis. Nine branches from the initial trellis, which are now drawn with light gray, are removed such that the number of sequences dropped from 82 to 64. All 18 removed sequences have energy $E^\bullet = 60$. Therefore, the operational average energy is now $E_{\text{ESS}} = E_{\min} = 9.6875$ which is the minimum possible for $k = 6$.

7.2.4 End-to-end Decoding Results

To evaluate the performance of energy-optimum ESS and the effect of using time-var. demapping (see Remark 7.1), end-to-end decoding is simulated with PAS using $\mathcal{A} = \{1, 3, 5, 7\}$. Both ESS and LA1 are considered. We expect that SM will perform identically to LA1 for the AWGN channel since they both minimize the average energy. As the FEC code, $n_{\text{c}} = 648$-bit systematic LDPC codes from the IEEE 802.11 standard are used [16]. The transmission rate is $R_{\text{t}} = 2$ bit/1-D. The uniform baseline uses the rate-$2/3$ code, while the shaped schemes have $k/N = 1.5$ bit/1-D using the rate-$5/6$ code. An FEC codeword consists of 216 real channel uses. Shaped schemes are simulated with $N = 216$, $N = 18$, and $N = 12$. For $N = 216$, shaping and coding blocklengths are the same. For $N = 18$ and $N = 12$, there are 12 and 18 shaped codewords inside an FEC codeword, resp. For $N = 18$, ESS is simulated with both the complete trellis and with an optimized version of it. For $N = 12$, ESS is simulated in three different settings: (1) with time-invariant demapping (2.8) using the average symbol distribution $p(a)$ of the sphere shaping set, (2) with time-invariant demapping (2.8) using the average symbol distribution $p^{\text{op}}(a)$ of the operational shaping set, and (3) with time-var. demapping (7.7) using the symbol distributions $p_n^{\text{op}}(a)$ of the operational shaping set for $n = 1, 2, \cdots, 12$.

In Fig. 7.4, FER is shown versus SNR. At $N = 216$, PAS with ESS and LA1 perform virtually the same, providing 1.05 dB gain over uniform signaling at an FER of 10^{-3}. This confirms our earlier observation that ESS and LA1 have identical energy efficiency for relatively large blocklengths. However, at $N = 18$, LA1 outperforms ESS by almost 0.1 dB and

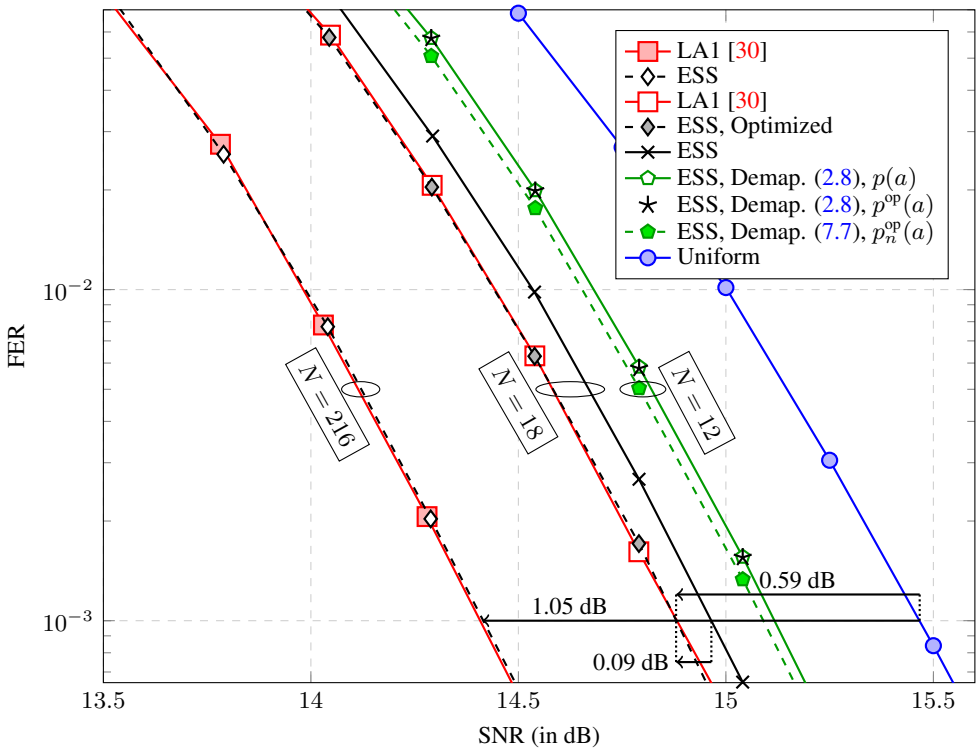

Figure 7.4: 648-bit LDPC-coded FER vs. SNR with 8-ASK at $R_{\mathrm{t}} = 2$ bit/1-D.

7

the uniform reference by 0.59 dB. We note that ESS was shown to be 0.08 dB less energy-efficient than LA1 at this blocklength in Fig. 7.2, which is in rough agreement with the loss in SNR efficiency seen from Fig. 7.4.

Then we simulated an optimized version of the ESS trellis at $N = 18$ such that some sequences with energy $E^{\bullet}$ are removed, and the number of sequences is decreased from $2^{27.3042}$ to $2^{27.0036}$. Here we note that the number of possible subsets $\hat{\mathcal{B}}$ that the minimization in (7.9) should be done over is 2^{72}. To limit the computation time, we refrained from searching for an optimum trellis that satisfies the inequality in (7.8). Instead, we successively removed branches in $\mathcal{B}$ from the trellis starting from the rightmost column until (7.8) is not satisfied. We see from Fig. 7.4 that ESS based on this optimized trellis overcomes the SNR gap to LA1, and it operates virtually the same again.

Finally, time-invariant demapping (2.8) and time-var. demapping (7.7) are compared at $N = 12$ in Fig. 7.4. We first see that for this set of parameters, using the operational average distribution $p^{\mathrm{op}}(a)$ provides no gain over the case where the distribution $p(a)$ from the com-

plete sphere is used. However, the time-var. demapping strategy (7.7) with $p_n^{\mathrm{op}}(a)$ provides improvement over time-invariant demapping strategies, although it is marginal, i.e., less than 0.05 dB. We believe that this is because the largest two amplitudes ($a = 5$ and $a = 7$ in this case) do not occur at the first position for the sequences in the operational shaping set, i.e., $p_1^{\mathrm{op}}(5) = p_1^{\mathrm{op}}(7) = 0$. Since using this information in the demapping effectively reduces the constellation to 16-QAM (instead of 64-QAM) for the first channel use, it provides somewhat observable gains.

7.3 Bounded Precision Implementation

In Sec. 5.1, we motivated the need for constructive sphere shaping algorithms by computing the required storage for a LUT-based implementation, and by demonstrating that it is impractically large. In general, for a fixed constellation and a given target shaping rate R_{s}, the size of the LUT that is needed to realize sphere shaping behaves as $\mathcal{O}(N2^{NR_{\mathrm{s}}})$ as a function of the shaping blocklength N. On the other hand, the storage complexity of ESS is $\mathcal{O}(N^3)$ which is significantly smaller for blocklengths larger than a dozen symbols. However, this is still a large memory demand for practical applications.

In this section, we will introduce a *bounded precision* (BP) sphere shaping implementation for ESS, LA1, and SM. In the BP implementation, numbers in an amplitude trellis will be computed by rounding the result of each arithmetic operation down to n_{m} bits and stored with fixed-length mantissas and exponents. This way, the storage complexity of ESS will be decreased to $\mathcal{O}(N^2 \log N)$, if a negligible rate loss is tolerated. A similar idea was considered for constrained coding in [109].

7.3.1 Approximate Base-2 Number Representation

Any k-bit number c can be rounded down to n_{m} bits and approximated as

$$c \approx \lfloor c \rfloor_{n_{\mathrm{m}}} = m2^p \tag{7.10}$$

where m is the n_{m}-bit mantissa, p is the n_{p}-bit exponent, and

$$n_{\mathrm{p}} = \lceil \log_2(k - n_{\mathrm{m}}) \rceil \tag{7.11}$$

in bits. Here $n_{\mathrm{m}} \geq 1$. Approximation in (7.10) is rounding n_{p} LSBs of c down to zero, while keeping n_{m} MSBs unchanged. Then the number c can approximately be stored in the form (m, p) using $(n_{\mathrm{m}} + n_{\mathrm{p}})$ bits instead of $k+1$. We call numbers stored in this form BP numbers.

7.3.2 Enumerative Sphere Shaping

7.3.2.1 Bounded Precision Backward Trellis

Based on the approximate base-2 representation in (7.10), we modify the backward trellis computation (5.4) to

$$T_n^e \triangleq \left\lfloor \sum_{a\in\mathcal{A}: e+a^2\le E^\bullet} T_{n+1}^{e+a^2} \right\rfloor_{n_\mathrm{m}}. \tag{7.12}$$

The result of (7.12) is then stored by a mantissa-exponent pair (m, p). We call T_n^e computed with (7.12) the BP backward trellis. We note that for $n_\mathrm{m} \ge k+1$, BP trellis becomes identical to the full precision (FP) trellis.

7.3.2.2 Proof of Invertibility

When the backward trellis is computed using (7.12), the invertibility of the shaping function, i.e., the reproducibility of an index through shaping and deshaping, becomes questionable.[2] We will now show that reproducibility based on Algorithms 5.1 and 5.2 is guaranteed if

$$T_n^e \le \sum_{a\in\mathcal{A}: e+a^2\le E^\bullet} T_{n+1}^{e+a^2} \tag{7.13}$$

is satisfied for $n = 0, 1, \cdots, N$ and $e \le E^\bullet$. We note that (7.12) satisfies this condition. The proof will consist of two steps: a lemma and a theorem.

Lemma 7.1. If $0 \le I_n < T_{n-1}^{e(a^{n-1})}$, then Algorithm 5.1 guarantees that $0 \le I_{n+1} < T_n^{e(a^n)}$. This implies that if $0 \le i < T_0^0$, then all I_n for $n = 1, 2, \cdots, N$ satisfy $0 \le I_n < T_{n-1}^{e(a^{n-1})}$.

Proof. Note that[3]

$$0 \le I_n < T_{n-1}^{e(a^{n-1})} \overset{(7.13)}{\le} \sum_{a\in\mathcal{A}} T_n^{e(a^{n-1},a)}. \tag{7.14}$$

Therefore, Algorithm 5.1 will always find an a_n that satisfies (5.12). From (5.12) and (5.13), we then find that

$$I_{n+1} = I_n - \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} \tag{7.15}$$

$$< \sum_{a\le a_n} T_n^{e\left(a^{n-1},a\right)} - \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} \tag{7.16}$$

$$= T_n^{e\left(a^{n-1},a_n\right)}, \tag{7.17}$$

[2] In [112] where arithmetic codes are considered for constrained coding, the reproducibility is called "representability", and it is defined as the dual of the decodability in arithmetic coding.

[3] Notation is simplified by replacing $a \in \mathcal{A} : e + a^2 \le E^\bullet$ into $a \in \mathcal{A}$, and $a \in \mathcal{A} : a \le a_n$ into $a \le a_n$.

and that

$$I_{n+1} = I_n - \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} \tag{7.18}$$

$$\geq \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} - \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} \tag{7.19}$$

$$= 0. \tag{7.20}$$

□

Theorem 7.1. Algorithms 5.1 and 5.2 guarantee that a local index $0 \leq I_n < T_{n-1}^{e\left(a^{n-1}\right)}$ in state $\left(n-1, e\left(a^{n-1}\right)\right)$ for $n = 1, 2, \cdots, N$ results in a sequence $(a_n, a_{n+1}, \cdots, a_N)$ that has local index $J_n = I_n$. Note that we are interested in $n = 1$ in the end.

Proof. The proof is by induction.

- First consider the state $(N-1, e(a^{N-1}))$ at depth $N-1$. The states at depth N to which this state is connected are final states. Observe that there are at least $T_{N-1}^{e(a^{N-1})}$ such final states since (7.13) holds. Note that since $I_N < T_{N-1}^{e(a^{N-1})}$ due to Lemma 7.1, there exist I_N final states below the state that corresponds to a_N that was chosen during shaping. These final states will lead to the local index $J_N = I_N$ during deshaping from (5.14).
- Next focus on the state $(n-1, e(a^{n-1}))$ at depth $n-1$, for $n < N$. During shaping, based on the local index I_n, an a_n was chosen and this resulted in the next local index I_{n+1} from (5.13). The induction hypothesis now tells that in state $(n, e(a^n))$, the corresponding sequence $(a_{n+1}, a_{n+2}, \cdots, a_N)$ will lead to a local index $J_{n+1} = I_{n+1}$. Therefore, the sequence $(a_n, (a_{n+1}, a_{n+2}, \cdots, a_N))$, by (5.14), and then by (5.13), leads to

$$J_n = \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} + J_{n+1} \tag{7.21}$$

$$= \sum_{a<a_n} T_n^{e\left(a^{n-1},a\right)} + I_{n+1} \tag{7.22}$$

$$= I_n. \tag{7.23}$$

□

We have shown now that reproducibility is guaranteed as long as (7.13) holds. Note that summations and subtractions in (5.12), (5.13), and (5.14) are assumed to be exact, more precisely, **not** followed by rounding. However, since the subtrahend in (5.13) and the addend in (5.14) are BP numbers, these exact arithmetic operations are only n_{m}-bit long. We will explain this in more detail as "sliding window shaping" in Sec. 7.4.

Remark 7.2 (BP implementation of LA1). Since the procedure of shaping within the N-shell explained in Sec. 5.7 is the same as ESS, proof of invertibility holds for LA1 as well.

7.3.3 Shell Mapping

7.3.3.1 Bounded Precision Forward Trellis

Based on the approximate base-2 representation in (7.10), we modify the forward trellis computation (5.17) to

$$F_n^e = \left\lfloor \sum_{b \le e} F_{n/2}^b F_{n/2}^{e-b} \right\rfloor_{n_\mathrm{m}} . \tag{7.24}$$

The result of (7.24) is then stored by a mantissa-exponent pair (m, p). We call F_n^e computed with (7.24) the BP forward trellis.

7.3.3.2 Proof of Invertibility

We will now show that the reproducibility based on Algorithms 5.3 and 5.4 is guaranteed if

$$F_n^e \le \sum_{b \le e} F_{n/2}^b F_{n/2}^{e-b} \tag{7.25}$$

is satisfied. We note that (7.24) satisfies this condition. The proof will again consist of two steps: a lemma and a theorem.

Lemma 7.2. If $0 \le I_n(a^n) < F_n^{e(a^n)}$, then Algorithm 5.3 guarantees that $0 \le I_{n/2}(a_i^n) < F_{n/2}^{e(a_i^n)}$ for $i = 1, 2$.[4] This implies that if $0 \le I_N(a^N) < F_N^{e(a^N)}$, then all $I_n(a^n)$ for $n = N/2, N/4, \cdots, 2$ satisfy $0 \le I_n(a^n) < F_n^{e(a^n)}$. Note that there are two $I_{N/2}$, four $I_{N/4}$, etc.

Proof. Note that

$$0 \le I_n\left(a^n\right) < F_n^{e(a^n)} \overset{(7.25)}{\le} \sum_{b \le e(a^n)} F_{n/2}^b F_{n/2}^{e(a^n)-b}. \tag{7.26}$$

Therefore, Algorithm 5.3 will always find an e_1 that satisfies (5.20). From (5.20) and (5.21), we then find that

$$0 \le D_\mathrm{s} < F_{n/2}^{e_1} F_{n/2}^{e(a^n)-e_1}. \tag{7.27}$$

[4] As in Algorithms 5.3 and 5.4, a_1^n denotes the first half of a^n with energy e_1, while a_2^n denotes its second half with energy e_2.

From (5.22a) and (5.22b), using $e_2 = e(a^n) - e_1$, we find that

$$0 \leq I_{n/2}\left(a_1^n\right) < F_{n/2}^{e_1}, \tag{7.28}$$
$$0 \leq I_{n/2}\left(a_2^n\right) < F_{n/2}^{e_2}. \tag{7.29}$$

□

Theorem 7.2. Algorithms 5.3 and 5.4 guarantee that a local offset $0 \leq I_n(a^n) < F_n^{e(a^n)}$ for $n = 2, 4, \cdots, N$, results in a sequence a^n that has a local offset $J_n(a^n) = I_n(a^n)$. We are interested in $n = N$ in the end.

Proof. The proof is by induction.

- First, consider depth 2. Observe that there are at least $F_2^{e(a^2)}$ possible symbol pairs since (7.25) holds. Note that since $I_2(a^2) < F_2^{e(a^2)}$ due to Lemma 7.2, there exists $I_2(a_1, a_2)$ pairs below the (a_1, a_2) which was chosen during shaping. These pairs will lead to a local offset $J_2(a_1, a_2) = I_2(a_1, a_2)$ during deshaping from (5.23b).
- Next focus on depth n for $n > 2$. During shaping, based on the local offset $I_n(a^n)$, an e_1 was chosen and this resulted in the next local offsets $I_{n/2}(a_1^n)$ and $I_{n/2}(a_2^n)$ from (5.22). The induction hypothesis now tells that in depth $n/2$, the corresponding sequences a_1^n and a_2^n will lead to local offsets $J_{n/2}(a_1^n) = I_{n/2}(a_1^n)$ and $J_{n/2}(a_2^n) = I_{n/2}(a_2^n)$. Therefore, the sequence $a^n = (a_1^n, a_2^n)$ by (5.23), and then by (5.21) and (5.22), leads to

$$J_n\left(a^n\right) = \sum_{b<e_1} F_{n/2}^b F_{n/2}^{e(a^n)-b} + D_{\mathrm{d}} \tag{7.30}$$
$$= \sum_{b<e_1} F_{n/2}^b F_{n/2}^{e(a^n)-b} + D_{\mathrm{s}} \tag{7.31}$$
$$= I_n\left(a^n\right). \tag{7.32}$$

□

We have shown now that the reproducibility within a shell is guaranteed as long as (7.25) holds. Along the same lines, we can show that (7.25) eventually implies that $j = i$. Here i is the index from which first the shell is chosen, and then local index $I_N(a^N)$ that enters the shell mapping procedure. Now j is the corresponding output index.

7.3.4 Required Storage

When a trellis is computed with BP, each element of the corresponding shaping matrix is $(n_{\mathrm{m}} + n_{\mathrm{p}})$-bit long, instead of $(k+1)$. Therefore, the required storage is now upper-bounded by $L(N+1)(n_{\mathrm{m}} + n_{\mathrm{p}})$ bits for ESS and LA1 as shown in Table 7.2. Considering Remark 5.1

and (7.11), we see that their storage complexity is $\mathcal{O}(N^2 \log N)$ as a function of N. On the other hand, the required storage is now upper-bounded by $L(\log_2 N + 1)(n_\mathrm{m} + n_\mathrm{p})$ bits for SM. Its storage complexity is $\mathcal{O}(N \log^2 N)$. In Table 7.2, we assumed that the extra step preceding LA1 and SM is realized with no storage as explained in Sec. 5.3.3.1. We will discuss the computational complexity of shaping with BP trellises in Sec. 7.4.1 after we explain sliding window shaping.

Table 7.2: Required Memory to Store BP Trellises

Technique	Memory (bits)
ESS	$L(N+1)(n_\mathrm{m} + n_\mathrm{p})$
LA1	$L(N+1)(n_\mathrm{m} + n_\mathrm{p})$
SM	$L\left(\log_2 N + 1\right)(n_\mathrm{m} + n_\mathrm{p})$

Example 7.2 (Required storage for BP ESS for the IEEE 802.11 Standard). We revisit the set of parameters used in Example 5.8: $N = 96$, $\mathcal{A} = \{1, 3, 5, 7\}$, and $L = 129$. When the backward trellis is computed with $n_\mathrm{m} = 12$ bits instead of FP (with $n_\mathrm{p} = 8$), the shaping rate drops from $R_\mathrm{s} = 1.7503$ to 1.75001 bit/1-D, while the input length stays fixed at $k = 168$ bits. Accordingly, the required storage decreases from 264.34 kB to 31.28 kB, leading to an almost 9-fold decrease.

Example 7.3 (Required storage for BP SM for the IEEE 802.11 standard). We revisit the set of parameters used in Example 5.9: $N = 32$, $\mathcal{A} = \{1, 3, 5, 7\}$, and $L = 48$. When the forward trellis is computed with $n_\mathrm{m} = 6$ bits instead of FP (with $n_\mathrm{p} = 6$), the shaping rate drops from $R_\mathrm{s} = 1.7557$ to 1.7531 bit/1-D, while the input length stays fixed at $k = 56$ bits. Accordingly, the required storage decreases from 2.05 kB to 0.43 kB, leading to an almost 5-fold decrease.

7

7.3.5 Bounded Precision Rate Loss

Numbers in a BP trellis $\tilde{T}_n^e$ are smaller than their FP counterparts T_n^e, which translates to a decrease in shaping rate. We call this decrease the BP rate loss, and we denote it by $R_{\text{loss,BP}}$. To quantify $R_{\text{loss,BP}}$, let $\tilde{c}$ be defined as $\tilde{c} = \lfloor c \rfloor_{n_\mathrm{m}}$. In the worst case, i.e., the case in which the largest possible relative error due to rounding occurs, $\tilde{c}$ can be lower-bounded as $\tilde{c} \geq (1-\delta)c$ where $\delta = 2^{-(n_\mathrm{m}-1)}$. Using this bound, the BP rate loss of a forward or backward trellis can be upper-bounded.

Proposition 7.1. In BP backward trellises, $\tilde{T}_n^e \geq T_n^e (1-\delta)^{(N-n)}$ for $n = 0, 1, \cdots, N$.

Proof. The proof is by induction.

- First consider $n = N$. Since $T_N^e = 1$ for $e \leq E^\bullet$ from (5.5), $\tilde{T}_N^e = T_N^e$.

- Next focus on depth n for $n < N$. The induction hypothesis tells that $\tilde{T}_{n+1}^e \geq T_{n+1}^e(1-\delta)^{N-(n+1)}$. From (7.12), we obtain

$$\tilde{T}_n^e = \left\lfloor \sum_{a\in\mathcal{A}} \tilde{T}_{n+1}^{e+a^2} \right\rfloor_{n_\mathrm{m}} \geq (1-\delta) \sum_{a\in\mathcal{A}} \tilde{T}_{n+1}^{e+a^2} \tag{7.33}$$

$$\geq (1-\delta) \sum_{a\in\mathcal{A}} (1-\delta)^{N-n-1} T_{n+1}^{e+a^2} \tag{7.34}$$

$$= (1-\delta)^{(N-n)} T_n^e. \tag{7.35}$$

□

Then the BP rate loss of a backward trellis can be upper-bounded by

$$R_{\text{loss,BP}} = \frac{1}{N} \log_2 \frac{T_0^0}{\tilde{T}_0^0} \leq -\log_2(1-\delta) = -\log_2\left(1 - 2^{-(n_\mathrm{m}-1)}\right) \tag{7.36}$$

in bit/1-D.

Proposition 7.2. In BP forward trellises, $\tilde{F}_n^e \geq F_n^e(1-\delta)^{(n-1)}$ for $n = 1, 2, 4, \cdots, N$, where $\tilde{F}_n^e$ denotes the trellis computed with BP.

Proof. The proof is by induction.

- First consider $n = 1$. By definition, $F_1^e = 1$ for $e \in \{1, 9, \cdots, (2|\mathcal{A}|-1)^2\}$. Thus, $\tilde{F}_1^e = F_1^e$.
- Next focus on depth n for $n \in \{2, 4, \cdots, N\}$. The induction hypothesis tells that $\tilde{F}_{n/2}^e \geq F_{n/2}^e(1-\delta)^{(n/2-1)}$. Consider from (7.25) that

$$\tilde{F}_n^e \geq (1-\delta) \sum_{b\leq e} \tilde{F}_{n/2}^b \tilde{F}_{n/2}^{e-b} \tag{7.37}$$

$$\geq (1-\delta) \sum_{b\leq e} (1-\delta)^{(n-2)} F_{n/2}^b F_{n/2}^{e-b} \tag{7.38}$$

$$= (1-\delta)^{(n-1)} F_n^e. \tag{7.39}$$

□

Then the BP rate loss of a forward trellis can be upper-bounded by

$$R_{\text{loss,BP}} = \frac{1}{N} \log_2 \frac{\sum_{e\leq E^\bullet} F_N^e}{\sum_{e\leq E^\bullet} \tilde{F}_N^e} \leq -\log_2(1-\delta) = -\log_2\left(1 - 2^{-(n_\mathrm{m}-1)}\right) \tag{7.40}$$

in bit/1-D.

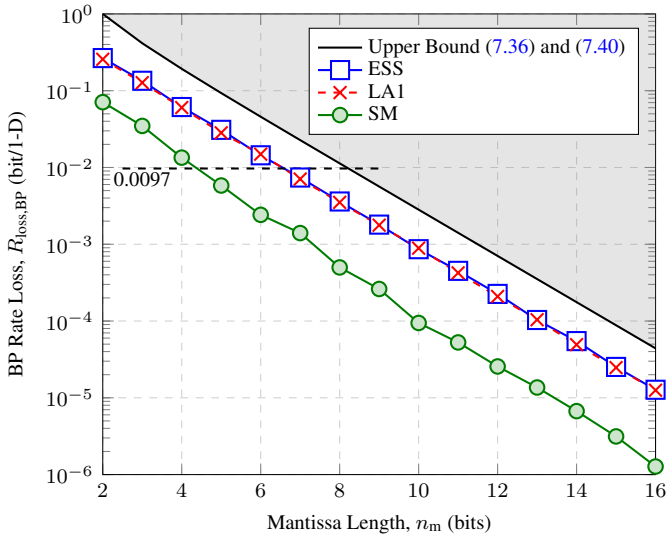

Figure 7.5: BP rate loss for $N = 64$, $L = 59$, and $\mathcal{A} = \{1, 3, 5, 7\}$.

Remark 7.3 (How to select n_{m}?). With FP computation, for a given set of parameters N, $\mathcal{A}$, and L, the rate of the sphere shaping set is R_{s} bit/1-D, and the input length of the corresponding shaping algorithm is $k = k_{\text{target}}$ bits. More precisely, we assume that the parameters are selected such that k satisfies a given target. In such a case, n_{m} should be selected as the smallest value that keeps $k \geq k_{\text{target}}$ to minimize the required storage.

7

In Fig. 7.5, actual BP rate losses of ESS, LA1, and SM are shown as a function of mantissa length n_{m}, along with the upper bound $-\log_2(1-\delta)$ in bit/1-D. For LA1 and SM, we assume that the preceding extra step is realized with no storage as explained in Sec. 5.3.3.1, which causes no additional BP rate loss. Here $N = 64$, $L = 59$, and $\mathcal{A} = \{1, 3, 5, 7\}$, for which the FP shaping rate is $R_{\text{s}} = 1.5097$ bit/1-D. The corresponding input length is $k = 96$ bits. We see from Fig. 7.5 that a small number of bits, e.g., a single byte, can be used to store mantissas instead of $k + 1 = 97$ bits, while the BP rate loss is kept smaller than 10^{-2} bit/1-D. Since SM computes the index of a sequence by concatenating multiple shorter sequences successively, rounding error accumulation during recursion starts later than that of ESS and LA1. Therefore, the BP rate loss of SM is smaller for the same n_{m}. In case the goal is to minimize n_{m} while keeping $k = 96$ as discussed in Remark 7.3, a BP rate loss less than $1.5097 - k/N = 0.0097$ bit/1-D can be tolerated. As shown in Fig. 7.5, $n_{\text{m}} = 7$ bits satisfy this for ESS and LA1, while for SM, $n_{\text{m}} = 5$ bits do. Thus, as a rule of thumb, we say that BP does not incur a noticeable loss in performance for $n_{\text{m}} > 10$ bits, which we designate as typical mantissa lengths.

Algorithm 7.1: BP Backward Amplitude Trellis Computation

Input: Blocklength N, alphabet $\mathcal{A}$ with $n_{\mathrm{a}} = |\mathcal{A}|$, and maximum energy $E^{\bullet}$
Output: Shaping matrix $\mathbf{T}$, i.e., backward trellis T_n^e for $n = 0, 1, \cdots, N$ and $0 \le e \le E^{\bullet}$

1 *Initialization:* $T_N^e = 1$ for $N \le e \le E^{\bullet}$
2 **for** $n = N-1, N-2, \cdots, 0$ **do**
3 **for** $e = n + 8(L-1), n + 8(L-2), \cdots, n$ **do**
4 $j \leftarrow n_{\mathrm{a}}$, $T_n^e \leftarrow 0$
5 **while** $j > 0$ **do**
6 **if** $e + \mathcal{A}(j)^2 \le E^{\bullet} - N + n$ **then**
7 $T_n^e \leftarrow \left\lfloor T_n^e + T_{n+1}^{e+\mathcal{A}(j)^2} \right\rfloor_{n_{\mathrm{m}}}$
8 $j \leftarrow j - 1$
9 **else**
10 $j \leftarrow j - 1$
11 **end**
12 **end**
13 **end**
14 **end**
15 **return T**

7.3.6 A More Realistic Bounded Precision Implementation

In (7.12), rounding is applied after all BP summands are added. BP backward trellis computation can also be realized by applying rounding after each addition. This approach is more suitable for practical implementation, and it is formulated in Algorithm 7.1. We note that in Algorithm 7.1, additions at the 7$^{\text{th}}$ line start with the nodes that correspond to transitions with higher amplitudes, i.e., from top to bottom. Thus, the quantitative effect of rounding is the same as in (7.12). This is because, with this approach, smaller BP numbers are added first (see Property $\mathbf{P_1}$ below), while the larger ones are added later which do not change the bits with lower significance in the outcome of the previous summation. The same idea can be applied to BP forward trellis computation (7.24) and rounding can be applied to the result of each multiplication and addition.

7.4 Sliding Window Shaping

When implemented with FP, ESS procedure in Algorithms 5.1 and 5.2 requires k-bit arithmetic operations where k can be more than 64 bits, e.g., $k = 168$ in Example 7.2, which is highly unfavorable for applications using 32- or 64-bit processors. However, as we hinted

Algorithm 7.2: FP Enumerative Shaping (Extended Version of Algorithm 5.1)

Input: Index $0 \leq I < T_0^0$, trellis $\mathbf{T}$, alphabet $\mathcal{A}$
Output: Sequence $s^N = (s_1, s_2, \cdots, s_N)$

Initialization: $I_1 \leftarrow I$
for $n = 1, 2, \cdots, N$ **do**
  $l \leftarrow 1$
  while $I_n \geq T_n^{\mathcal{A}(l)^2 + \sum_{j=1}^{n-1} s_j^2}$ **do**
    $I_n \leftarrow I_n - T_n^{\mathcal{A}(l)^2 + \sum_{j=1}^{n-1} s_j^2}$
    $l \leftarrow l + 1$
  end
  $I_{n+1} \leftarrow I_n$
  $s_n \leftarrow a_l$
end
return $s^N = (s_1, s_2, \cdots, s_N)$

in [102], the necessity to realize k-bit operations can be removed with the BP implementation by using *sliding window shaping* (SWS). SWS allows operations on the k-bit input index to be carried out locally on its first $n_{\mathrm{m}} + \log_2 n_{\mathrm{a}} = n_{\mathrm{m}} + m_{\mathrm{a}}$ significant binary digits in a sliding window manner where $n_{\mathrm{a}} = |\mathcal{A}|$. We explain SWS based on Algorithm 7.2 which is an extended version of the shaping procedure in Algorithm 5.1 where we now emphasize that comparisons/subtractions start with the nodes that correspond to transitions with lower amplitudes, i.e., from bottom to top.

Before we explain SWS, we first provide two properties of backward trellises that will be useful in the subsequent sections.

P₁ It follows from (5.4) for the FP, and from Algorithm 7.1 for the BP implementation that if $e_1 \leq e_2$, then $T_n^{e_1} \geq T_n^{e_2}$.

P₂ It follows from Property **P₁** and (5.4) for the FP trellis, and from Property **P₁** and Algorithm 7.1 for the BP trellis that

$$T_n^e \leq n_{\mathrm{a}} \max_{a \in \mathcal{A}} T_{n+1}^{e+a^2} = n_{\mathrm{a}} T_{n+1}^{e+1}. \tag{7.41}$$

Thus, T_n^e is at most $\log_2 n_a = m_{\mathrm{a}}$ bits longer than T_{n+1}^{e+1}.

Now consider the enumerative shaping procedure in Algorithm 7.2. At the beginning of

the n^{th} iteration of Algorithm 7.2 (at the 4^{th} line, $l = 1$, $\mathcal{A}(1) = 1$), the following holds:

$$I_n < T_{n-1}^{\sum_{j=1}^{n-1} s_j^2} \tag{7.42}$$

$$\leq \sum_{a \in \mathcal{A}} T_n^{a^2 + \sum_{j=1}^{n-1} s_j^2} \tag{7.43}$$

$$\leq n_{\text{a}} \max_{a \in \mathcal{A}} T_n^{a^2 + \sum_{j=1}^{n-1} s_j^2}$$

$$= n_{\text{a}} T_n^{1 + \sum_{j=1}^{n-1} s_j^2}. \tag{7.44}$$

Here (7.42) follows from Lemma 7.1, (7.43) is by construction of the BP trellis in Algorithm 7.1, and (7.44) is due to Property $\mathbf{P_1}$. Then, I_n can at most be $m_{\text{a}} = \log_2 |\mathcal{A}|$ bits longer than $T_n^{1 + \sum_{j=1}^{n-1} s_j^2}$ which has an n_{m}-bit mantissa. Thus, the first comparison at the 4^{th} line of Algorithm 7.2 ($l = 1$, $a_1 = 1$) concerns only the first $n_{\text{m}} + m_{\text{a}}$ significant binary digits of I_n.

If at the 4^{th} line of Algorithm 7.2, $I_n \geq T_n^{1 + \sum_{j=1}^{n-1} s_j^2}$, the following subtraction occurs at the 5^{th} line:

$$I_n - T_n^{1 + \sum_{j=1}^{n-1} s_j^2} \overset{\text{(d)}}{\leq} \sum_{a \in \mathcal{A}} T_n^{a^2 + \sum_{j=1}^{n-1} s_j^2} - T_n^{1 + \sum_{j=1}^{n-1} s_j^2} \tag{7.45}$$

$$= \sum_{l=2}^{n_{\text{a}}} T_n^{\mathcal{A}(l)^2 + \sum_{j=1}^{n-1} s_j^2}$$

$$\overset{\text{(e)}}{\leq} (n_{\text{a}} - 1) T_n^{9 + \sum_{j=1}^{n-1} s_j^2}. \tag{7.46}$$

Here (7.45) is due to (7.43), and (7.46) follows from Property $\mathbf{P_1}$. Consequently, the result of this subtraction can at most be $\lceil \log_2(n_{\text{a}} - 1) \rceil = m_{\text{a}}$ bits longer than $T_n^{9 + \sum_{j=1}^{n-1} s_j^2}$, which has an n_{m}-bit mantissa. Therefore, the second comparison at the 4^{th} line of Algorithm 7.2 ($l = 2$, $\mathcal{A}(2) = 3$) concerns only the first $n_{\text{m}} + m_{\text{a}}$ significant binary digits of I_n (which was updated by a subtraction at the 5^{th} line). Following this reasoning recursively, we see that each subtraction (or comparison) in Algorithm 7.2 considers only the first $n_{\text{a}} + m_{\text{a}}$ significant binary digits of the local index I_n. Therefore, shaping operates on $(n_{\text{m}} + m_{\text{a}})$-bit portions of the input k-bit index, sliding from the MSB to the least significant one. In the following, we give an example for SWS.

Example 7.4 (Sliding window shaping with a backward BP trellis). We consider the BP trellis computed using $N = 4$, $\mathcal{A} = \{1, 3, 5, 7\}$, and $L = 8$, i.e., $E^{\bullet} = 60$, with $n_{\text{m}} = 3$-bit mantissas. The corresponding shaping array $\mathbf{T}$ is given in (7.47). Here, each entry consists of a *binary* mantissa m with MSB on the left, and a *decimal* exponent p. For this trellis, $T_0^0 = (m, p) = (100, 4)$, which means there are $m2^p = 64$ amplitude sequences represented. Therefore, possible $k = \lfloor \log_2 T_0^0 \rfloor = 6$-bit input indices are $i \in [0, 64)$.

$$\mathbf{T} = \begin{bmatrix} (001,0) & (001,0) & (001,0) & (001,0) & (001,0) \\ (101,0) & (100,0) & (011,0) & (010,0) & (001,0) \\ (101,1) & (111,0) & (100,0) & (010,0) & (001,0) \\ (100,2) & (101,1) & (110,0) & (011,0) & (001,0) \\ (111,2) & (100,2) & (100,1) & (011,0) & (001,0) \\ (101,3) & (101,2) & (100,1) & (011,0) & (001,0) \\ (110,3) & (110,2) & (101,1) & (100,0) & (001,0) \\ (\mathbf{100},\mathbf{4}) & (100,3) & (110,1) & (100,0) & (001,0) \end{bmatrix}. \tag{7.47}$$

Consider $i = 59$, i.e., $I_1 = (111011)$ as shown in Fig. 7.6. Following Algorithm 7.2, shaping starts with comparing I_1 to $T_1^1 = (100,3)$ to check whether $s_1 = 1$ or not. Since the exponent of T_1^1 is $p = 3$, its mantissa is shifted to the left by 3 binary digits. Then this mantissa is compared with the first 3 significant binary digits of I_1. Since $I_1 \geq T_1^1$, we subtract T_1^1 from I_1 to update the local index $I_1 \leftarrow (011011)$. This subtraction is local and concerns the first $n_\mathrm{m} = 3$ significant binary digits of I_1.

Next, the updated local index $I_1 = (011011)$ is compared with $T_1^9 = (110,2)$ to check whether $s_1 = 3$ or not. Since the exponent of T_1^9 is $p = 2$, its mantissa is shifted to the left by 2 binary digits. Then this mantissa is compared with the first 3 significant binary digits of the local index. Since $I_1 \geq T_1^9$, we subtract T_1^9 from I_1 to update the local index $I_1 \leftarrow (000011)$. This subtraction is again local and concerns the first $n_\mathrm{m} = 3$ significant binary digits of I_1.

Finally, the updated local index $I_1 = (000011)$ is compared with $T_1^{25} = (100,2)$ to check whether $s_1 = 5$ or not. Since the exponent of T_1^{25} is $p = 2$, its mantissa is shifted to the left by 2 binary digits. Then this mantissa is compared with the first 3 significant binary digits of the local index. Since $I_1 < T_1^{25}$, we set $I_2 = I_1$, and output $s_1 = 5$. Then the shaping procedure continues with the next step, i.e., $n = 2$, as shown in Fig. 7.6. We observe that each subtraction (or comparison) deals only with the first $n_\mathrm{m} = 3$ significant binary digits of the local index I_n, while the position of the operations gradually shifts towards the LSB.[5]

7.4.1 Computational Complexity

The computational requirement of SWS is at most $(n_\mathrm{a} - 1)$ arithmetic operations per output symbol, each of which is $(n_\mathrm{m} + m_\mathrm{a})$-bit long. Here $n_\mathrm{a} = |\mathcal{A}|$ and $m_\mathrm{a} = \log_2 |\mathcal{A}|$. Therefore, as shown in Table 7.3, at most $(n_\mathrm{a} - 1)(n_\mathrm{m} + m_\mathrm{a})$ bit oper./1-D are necessary to realize BP ESS and LA1. Typically, n_m is smaller than 16 [32, Example 8], and thus, ESS with 8-ASK ($n_\mathrm{a} = 4$) requires only three 16-bit operations per output symbol for most blocklengths.

Example 7.5 (Complexity of BP ESS with SWS for the IEEE 802.11 Standard). We revisit the set of parameters used in Example 7.2: $N = 96$, $\mathcal{A} = \{1,3,5,7\}$, $L = 129$, and $n_\mathrm{m} = 12$ bits. The corresponding sliding window shaping and deshaping algorithms require 36 bit oper./1-D instead of 507 as in Example 5.8, leading to a 14-fold decrease.

[5] We note that although all operations in Example 7.4 are carried out on the first $n_\mathrm{m} = 3$ significant binary digits of the local index I_n, in general, they may affect the first $n_\mathrm{m} + m_\mathrm{a} = 5$ binary digits, e.g., when a borrow is needed.

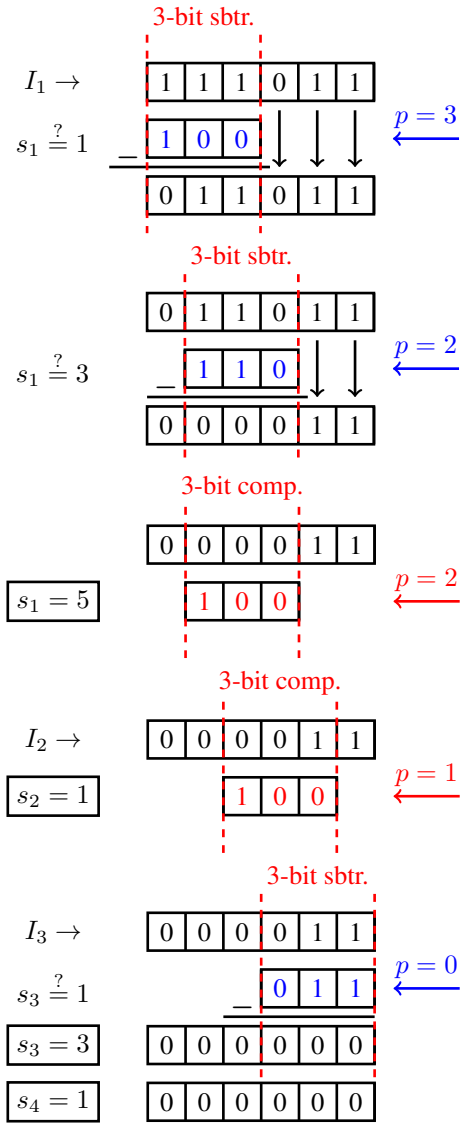

Figure 7.6: Sliding window shaping of $I = (111011)$ based on the shaping array $\mathbf{T}$ in (7.47). The output sequence is $s^N = (5, 1, 3, 1)$. In (7.47), subtracted values are written in blue while compared values are written in red.

Table 7.3: Computational Complexity of BP Sphere Shaping

Technique	Complexity (Bit Oper./1-D)
ESS	$(n_\mathrm{a}-1)(n_\mathrm{m}+m_\mathrm{a})$
LA1	$(n_\mathrm{a}-1)\,(n_\mathrm{m}+m_\mathrm{a})$
SM	Ln_m^2

Remark 7.4 (Complexity of BP SM). When SM is realized with BP (without SWS), at most L multiplications of n_m-bit numbers must be realized. Therefore, as shown in Table 7.3, at most Ln_m^2 bit oper./1-D are necessary to realize BP SM.

Example 7.6 (Complexity of BP SM for the IEEE 802.11 Standard). We revisit the set of parameters used in Example 7.3: $N = 32$, $\mathcal{A} = \{1, 3, 5, 7\}$, $L = 45$, and $n_\mathrm{m} = 6$ bits. The corresponding shaping and deshaping algorithms require 1728 bit oper./1-D instead of 155952 as in Example 5.9, leading to a 90-fold decrease.

7.5 On-the-fly Backward Trellis Computation

Both FP and BP ESS require that the shaping array $\mathbf{T}$ is precomputed and stored in memory. In the FP case, each element of $\mathbf{T}$ can at most be $\lceil \log_2 T_0^0 \rceil = (k+1)$-bit long. In the BP case, each element of $\mathbf{T}$ can at most be $(n_\mathrm{m}+n_\mathrm{p})$-bit long. Thus, the required storage for FP ESS is $L(N+1)(k+1)$ bits, while it is $L(N+1)(n_\mathrm{m}+n_\mathrm{p})$ for BP ESS as shown in Tables 5.3 and 7.2, respectively. As an example, the size of the required memory to realize ESS for the IEEE 802.11 standard with $N = 96$ was computed to be more than 264 kB with FP in Example 5.8, and more than 32 kB with BP in Example 7.2. Therefore, storing the shaping array $\mathbf{T}$ requires a relatively large allocated memory (usually more than 10 kB), even in the BP case. To further reduce the required storage, only the first column of the trellis $\mathbf{t}_0$ can be stored after the initial computation, and the other columns can be computed when they are needed during the shaping procedure. We call this the *on-the-fly* (OtF) backward trellis computation.

In the FP case, OtF computation can be realized straightforwardly since the connections in the trellis are enough to link a column $\mathbf{t}_{n+1}$ to its neighbour $\mathbf{t}_n$ as in Fig. 7.1. However, in the BP case, consecutive columns cannot be linked to each other using only the connections due to the effect of the rounding operation at the 7$^{\text{th}}$ line of Algorithm 7.1. To examine this effect, we consider the isolated part of a BP trellis in Fig. 7.7 (left) where the initial trellis computation is considered for $\mathcal{A} = \{1, 3, 5, 7\}$. We see from Algorithm 7.1 that during the initial computation, T_n^e is calculated as

$$T_n^e = \left\lfloor T_{n+1}^{e+1} + \gamma \right\rfloor_{n_\mathrm{m}} \tag{7.48}$$

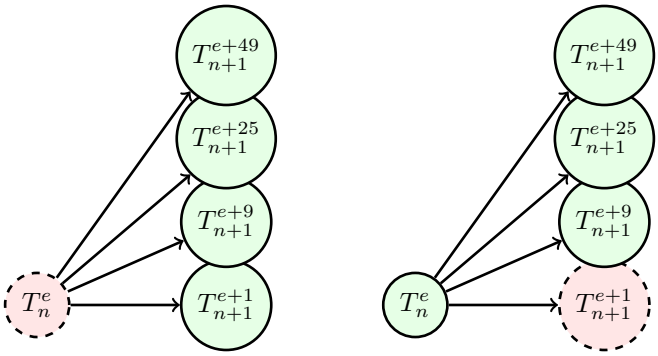

Figure 7.7: (Left) An instance of the initial trellis computation starting from $n = N$ to $n = 0$, i.e., right-to-left. (Right) An instance of the on-the-fly trellis computation starting from $n = 0$ to $n = N$, i.e., left-to-right. Here $n_{\mathrm{a}} = |\mathcal{A}| = 4$. Dashed nodes filled with red are to be computed using already-known solid nodes which are filled with green.

where

$$\gamma = \left\lfloor T_{n+1}^{e+9} + \left\lfloor T_{n+1}^{e+25} + T_{n+1}^{e+49} \right\rfloor_{n_{\mathrm{m}}} \right\rfloor_{n_{\mathrm{m}}}. \tag{7.49}$$

Due to Property $\mathbf{P_2}$, at most $m_{\mathrm{a}} = \log_2 |\mathcal{A}|$ bits are affected from rounding in (7.48). We call these m_{a} bits the *remainders* and denote them by r_n^e. More specifically, (7.48) can be rewritten as

$$T_n^e = \left\lfloor T_{n+1}^{e+1} + \gamma \right\rfloor_{n_{\mathrm{m}}} = T_{n+1}^{e+1} + \gamma - r_n^e. \tag{7.50}$$

Now consider Fig. 7.7 (right) where OtF trellis computation is considered, i.e., given $\mathbf{t}_n$, $\mathbf{t}_{n+1}$ is to be computed. We assume that nodes of higher energy are computed before nodes of lower energy. Consequently, when the node $(n+1, e+1)$ is to be calculated, nodes (n, e), $(n+1, e+9)$, $(n+1, e+25)$ and $(n+1, e+49)$ have already been computed, and so does γ in (7.49). Then the node $(n+1, e+1)$ can be computed from (7.50) as

$$T_{n+1}^{e+1} = T_n^e - \gamma + r_n^e, \tag{7.51}$$

which requires that r_n^e was stored during the initial computation. Provided that r_n^e is stored for $n = 1, 2, \cdots, N$ and $e \leq E^{\bullet}$ along with the first column $\mathbf{t}_0$, BP OtF trellis computation in (7.51) can be generalized as in Algorithm 7.3.

7.5.1 Required Storage and Computational Complexity

For BP OtF trellis computation, $L(n_{\mathrm{m}} + n_{\mathrm{p}})$ bits are necessary to store $\mathbf{t}_0$. In addition, LNm_{a} bits should be allocated to store the remainders r_n^e for $n = 0, 1, \cdots, N-1$ and for $e \leq E^{\bullet}$. Thus, the storage requirement is $L(n_{\mathrm{m}} + n_{\mathrm{p}} + Nm_{\mathrm{a}})$ bits as shown in Table 7.4.

Algorithm 7.3: On-the-fly BP Backward Amplitude Trellis Computation

Input: n, $\mathbf{t}_{n-1}$, $\mathcal{A}$, and r_{n-1}^{e} for $e \leq E^{\bullet}$
Output: $\mathbf{t}_n$
for $l = L-1, L-2, \cdots, 0$ **do**
  $j \leftarrow n_{\mathrm{a}}$, $\gamma \leftarrow 0$, $e \leftarrow n + 8l$
  while $j > 2$ **do**
    if $e + \mathcal{A}(j)^2 - 1 \leq n + 8(L-1)$ **then**
      $\gamma \leftarrow \left\lfloor \gamma + T_n^{e+\mathcal{A}(j)^2-1} \right\rfloor_{n_{\mathrm{m}}}$
    else
      $\gamma \leftarrow \gamma$
    end
    $j \leftarrow j - 1$
  end
  $T_n^e \leftarrow T_{n-1}^{e-1} - \gamma + r_{n-1}^{e-1}$
end
return $\mathbf{t}_n$

Table 7.4: Required Storage and Computational Complexity: Backward Trellis

	Technique	Storage (bits)	Complexity (Bit Oper./1-D)
Table 5.3	FP Trellis	$L(N+1)(k+1)$	0
	OtF FP Trellis	$L(k+1)$	$L(n_{\mathrm{a}}-1)(k+1)$
Table 7.2	BP Trellis	$L(N+1)(n_{\mathrm{m}}+n_{\mathrm{p}})$	0
	OtF BP Trellis	$L\left(n_{\mathrm{m}}+n_{\mathrm{p}}+Nm_{\mathrm{a}}\right)$	$L(n_{\mathrm{a}}-1)n_{\mathrm{m}}$

During BP OtF trellis computation, L numbers must be computed per column, i.e., per output symbol.[6] Each of these computations requires at most $n_{\mathrm{a}} - 2$ additions and 1 subtraction of n_{m}-bit numbers, and a single m_{a}-bit addition. Therefore, neglecting the m_{a}-bit addition at the 11^{th} step of Algorithm 7.3, the computational requirement is at most $L(n_{\mathrm{a}}-1)n_{\mathrm{m}}$ bit oper./1-D as shown in Table 7.4.

Example 7.7 (Complexity of OtF BP ESS for the IEEE 802.11 Standard). We revisit the set of parameters used in Example 7.2: $N = 96$, $\mathcal{A} = \{1, 3, 5, 7\}$, $L = 129$, and $n_{\mathrm{m}} = 12$ bits. With OtF computation, the required storage further decreases from $L(N+1)(n_{\mathrm{m}}+n_{\mathrm{p}}) = 32.28$ kB to $L(n_{\mathrm{m}}+n_{\mathrm{p}}+Nm_{\mathrm{a}}) = 3.42$ kB, leading to an almost 10-fold decrease. The corresponding shaping and deshaping algorithms still require 36 bit oper./1-D. However now, additional computational complexity is introduced due to OtF computation.

[6] Here we neglect the fact that in practice, some of the lower energy nodes might not need to be computed after a certain point since lower rows of the trellis become irrelevant for the shaping procedure. Therefore, it may be enough to compute less than L numbers.

To compute the trellis OtF, at most $L(n_\mathrm{a}-1)(n_\mathrm{m}+n_\mathrm{p}) = 7740$ bit oper./1-D are necessary. Compared to BP SM at $N = 32$ in Example 7.3, BP OtF ESS requires slightly more storage (3.42 kB instead of 0.43) with a much smaller computational requirement ($7740+36 = 7776$ bit oper./1-D instead of 111952).

7.6 Conclusion

In this chapter, we searched for an answer to the following research question.

RQ-5 Can we further improve the energy-efficiency of ESS for practical scenarios? How can ESS be implemented with low storage complexity, minimal computational requirements, and limited latency?

We computed the symbol distribution for the operational shaping set of ESS, and we showed that ESS is slightly less energy-efficient than LA1 and SM. We proposed a straightforward heuristic routine to optimize the ESS trellis such that its energy-efficiency is improved. We then introduced a bounded precision computation and storage technique for ESS, LA1, and SM which results only in a negligible rate loss. With this technique, the required storage to realize these algorithms is significantly reduced. Then we devised a sliding-window shaping (SWS) procedure for ESS and LA1 which only requires fixed- and short-length arithmetic operations. This way, the computational complexity of shaping and the required arithmetic precision are decreased. Furthermore, unlike regular enumerative shaping, SWS can start outputting symbols as soon as the procedure starts which leads to reduced and limited latency. Finally, we introduced the on-the-fly trellis computation technique for ESS which further decreases the storage complexity at the expense of increased computational load. All techniques considered, ESS can be implemented for the IEEE 802.11 standard [16] at $N = 96$ either with (1) slightly more than 30 kB storage and 36 bit oper./1-D, or with (2) slightly more than 3 kB storage and 7740 bit/oper./1-D.

CHAPTER 8

Partial Enumerative Sphere Shaping

Parts of this chapter are published in:

Y. C. Gültekin, W. J. van Houtum, A. G. C. Koppelaar, and F. M. J. Willems, "Partial enumerative sphere shaping," in *Proc. IEEE Veh. Technol. Conf. (VTC-Fall),* Honolulu, HI, USA, Sep. 2019.

Y. C. Gültekin, F. M. J. Willems, W. J. van Houtum, and S. Şerbetli, "K-bit enumerative sphere shaping of multidimensional constellations," U.S. Patent 10 523 480 B1, Dec., 31, 2019.

8.1 Introduction

In this chapter, we will investigate the research question **RQ-6** which arises naturally while considering constellation shaping: How shaped the channel input distribution should be to close most of the shaping gap? In the context of the direct method, i.e., DM, this question can be reformulated as "How sensitive the MI (or the corresponding AIR) is to the changes in $p(x)$?" On the other hand for the indirect method, i.e., sphere shaping, it is "How close the signal structure should be to an N-sphere?" Our motivation to investigate these questions stems from the following fact. If some type of a *partially* shaped input distribution is enough to obtain most of the maximum shaping gain, it may be possible to decrease the storage space and computational resources reserved for the shaping operation.

Here, we first define an approximation to the MB distribution. We demonstrate that amplitudes with these approximate distributions have some of their amplitude bit-levels uniform and independent of the others. Then we evaluate gap-to-capacity (3.97) for these approximate distributions. We show that when only some amplitude bit-levels are shaped, it is still possible to reap most of the maximum shaping gain, especially for large constellations. Subsequently, we propose an amplitude shaping architecture which we call *partial ESS (P-ESS)* to produce channel inputs that have these approximate distributions. We evaluate the performance of P-ESS in terms of rate loss (4.11) and end-to-end decoding results. Finally, we demonstrate the reduction in required storage and computational complexity provided by P-ESS.

8.2 Effect of "Gaussianity" on Gap-to-capacity

To investigate how well the amplitude distribution $p(a)$ has to resemble an MB distribution to close the most of the shaping gap, we define a particular type of approximation which we call *partial MB distribution*. We will later relate these approximate amplitude-level distributions to bit-level distributions. The basic idea here is to realize MB distributions over amplitude pairs, quartets, etc., instead of individual amplitudes. These approximations can be considered as quantized versions of the MB distribution. We now explain this with an example.

Example 8.1 (Partially MB-distributed 16-ASK). We consecutively gather amplitudes of 16-ASK $\mathcal{A} = \{1, 3, 5, 7, 9, 11, 13, 15\}$ into groups of two, i.e., $\mathcal{A}_1 = \{1, 3\}$, $\mathcal{A}_2 = \{5, 7\}$, $\mathcal{A}_3 = \{9, 11\}$, and $\mathcal{A}_4 = \{13, 15\}$. Then we define the MB distribution (2.24) over these pairs as

$$\Pr\{a \in \mathcal{A}_i\} = K(\lambda) \exp\left(-\lambda \mathbb{E}\left[|\mathcal{A}_i|^2\right]\right) \tag{8.1}$$

where $\mathbb{E}[|\mathcal{A}_i|^2]$ is the average energy of $a \in \mathcal{A}_i$ assuming they are equiprobable, i.e.,

$$\mathbb{E}\left[|\mathcal{A}_i|^2\right] = \frac{1}{2} \sum_{a \in \mathcal{A}_i} a^2, \quad \text{for } i \in \{1, 2, 3, 4\}. \tag{8.2}$$

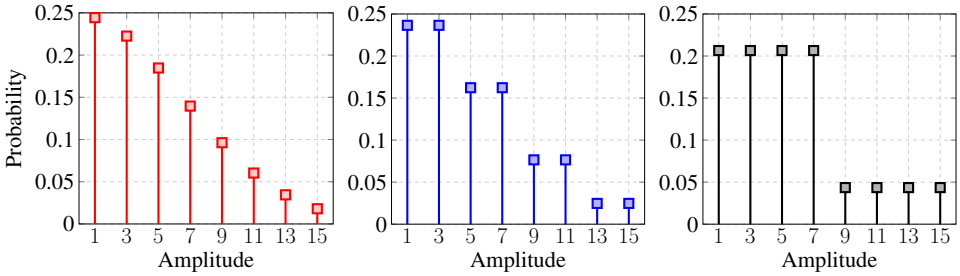

Figure 8.1: (Left) Fully, and (middle) 2-bit and (right) 1-bit partially MB-distributed $p(a)$ for 16-ASK.

This approximation can be realized over symbol quartets $\{1, 3, 5, 7\}$ and $\{9, 11, 13, 15\}$, and so on. As discussed in Sec. 2.6.1, λ governs the variance of the distribution in (8.1) while $K(\lambda)$ is a normalization factor.

Table 8.1: Fully and Partially MB-Distributed $p(a)$ for 16-ASK

$p(1)$	$p(3)$	$p(5)$	$p(7)$	$p(9)$	$p(11)$	$p(13)$	$p(15)$	E_{av}	G_{s} (in dB)
0.2443	0.2225	0.1847	0.1396	0.0962	0.0603	0.0345	0.0180	38.66	1.40
0.2365	0.2365	0.1623	0.1623	0.0765	0.0765	0.0247	0.0247	39.57	1.30
0.2065	0.2065	0.2065	0.2065	0.0435	0.0435	0.0435	0.0435	43.27	0.92

In Table 8.1, a numeric approximate MB distribution example is tabulated for 16-ASK. Here, the first row is the exact MB distribution, and the following are the approximations over 2- and 4-symbol groups, respectively, all rounded to the nearest 4 decimal digits. The entropy of the distribution $\mathbb{H}(A) = 2.667$ bits in all three cases. The average energy $E_{\text{av}} = \mathbb{E}[A^2]$ and the shaping gain G_{s} with respect to uniform signaling (4.4) are also provided in Table 8.1 assuming shaping rate of the distribution $R_{\text{s}} = \mathbb{H}(A)$. We see that as we apply the MB distribution over symbols, pairs, and quartets for a fixed entropy, E_{av} increases which indicates that energy efficiency is decreasing. This can also be verified by observing the decreasing shaping gain. The distributions in Table 8.1 are also shown in Fig. 8.1.

When the amplitudes of the channel inputs have these approximate MB distributions and BRGCs are used for labeling, some amplitude bit-levels become uniform and independent of the others. Consider Example 8.1 and the amplitude bits of the BRGC for 16-ASK given in Table 8.2. Pairing the amplitudes of 16-ASK and transmitting the elements of a group equiprobably means that the bit-level B_4 is now uniform and independent of the other two amplitude bits. Similarly, assigning the same probability to each amplitude in a consecutive group of four implies that the bit-levels B_3 and B_4 are uniform and independent of the others. We call these distributions *s-bit shaped* where $s < m - 1$ is the number of shaped amplitude bit-levels. If $s = m - 1$, i.e., all amplitude bit-levels are shaped, we call the corresponding

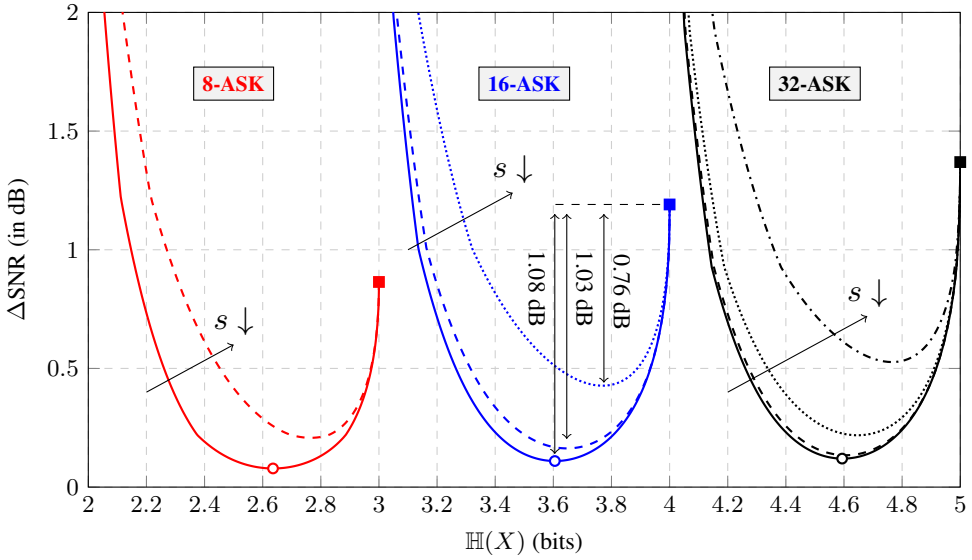

Figure 8.2: Constellation entropy $H(X)$ vs. gap-to-capacity ΔSNR for 2^m-ASK at rate $R_{\mathrm{t}} = m-1$ bit/1-D for $m = 3, 4, 5$ and $s = 1, 2, \cdots, m-1$: (Solid) $s = m-1$, (dashed) $s = m-2$, (dotted) $s = m-3$, and (dash-dotted) $s = m-4$.

distribution *fully-shaped*.

Table 8.2: The Amplitude Bits of the BRGC for 16-ASK

A	1	3	5	7	9	11	13	15
B_2	0	0	0	0	1	1	1	1
B_3	0	0	1	1	1	1	0	0
B_4	0	1	1	0	0	1	1	0

In Fig. 8.2, the gap-to-capacity (3.97) is plotted for 8-, 16-, and 32-ASK, and for $s = 1, 2 \cdots, m-1$. The target transmission rate is $R_{\mathrm{t}} = m-1$ bit/1-D. Square markers indicate the gap-to-capacity of uniform 2^m-ASK at rate $m-1$ bit/1-D. Circle markers denote $H(X)$ for which the corresponding MB distribution minimizes the gap-to-capacity at rate $m-1$ bit/1-D. We call the vertical difference between the square and circle markers the *maximum shaping gain*.

The important observation from Fig. 8.2 is that it is possible to obtain a large portion of the maximum shaping gain even when only a couple of amplitude bits are shaped. As an example, the maximum gain for 16-ASK drops from 1.08 dB to 1.03, and then to 0.76

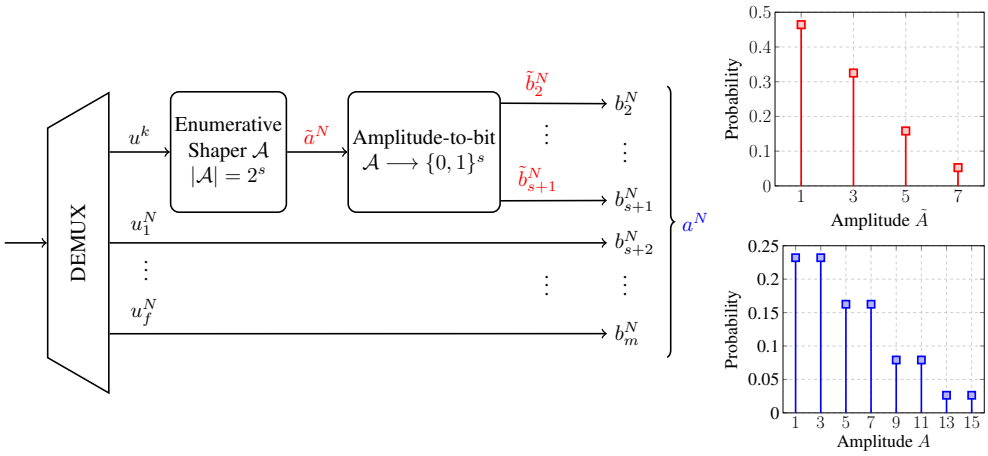

Figure 8.3: (Left) Block diagram of partial enumerative sphere shaper with rate $R_s = \frac{k}{N} + f$ bit/1-D. (Right) Output distributions of the enumerative shaper (red) and the P-ESS block (blue) for $s = 2$-bit shaped 16-ASK, i.e., $f = 1$, with $k/N = 1.667$ at $N = 216$.

when 2 and 1 bits are shaped instead of 3, respectively. Motivated by this, next we will build an amplitude shaping block based on ESS to realize output distributions resembling the approximate, i.e., quantized, MB distributions (8.1). We note that a similar gap-to-capacity analysis is provided in [81], only for product distributions.

8.3 Partial Enumerative Sphere Shaping

Our goal is to produce channel inputs with a distribution resembling the approximate MB distributions defined in Example 8.1. This is equivalent to keeping some amplitude bits uniform and independent of the others. The number of shaped and uniform amplitude bit-levels are denoted by s and f, respectively, where $m - 1 = s + f$.

We propose to use an enumerative shaper that operates based on the 2^{s+1}-ASK amplitude alphabet as shown in Fig. 8.3. This shaper maps k-bit message indices u^k to shaped amplitude sequences $\tilde{a}^N$. The distribution of $\tilde{a}^N$ is Gaussian-like over $\{1, 3, \cdots, 2^s - 1\}$. Accordingly, corresponding binary amplitude labels $\tilde{b}_2\tilde{b}_3 \cdots \tilde{b}_{s+1}$ are also shaped. Therefore, we now have s shaped bit-levels which are highlighted by red in Fig. 8.3. We will later show in Sec. 8.4.2 that a standard enumerative shaper can be used without any modification in P-ESS to obtain partial MB distributions over ASK alphabets.

Then f additional N-bit data sequences $u_1^N, u_2^N, \cdots, u_f^N$ are used as the uniform amplitude bit-levels for 2^m-ASK, and they are combined with the s shaped levels. The way uniform and shaped amplitude bit-levels are combined depends on the employed binary la-

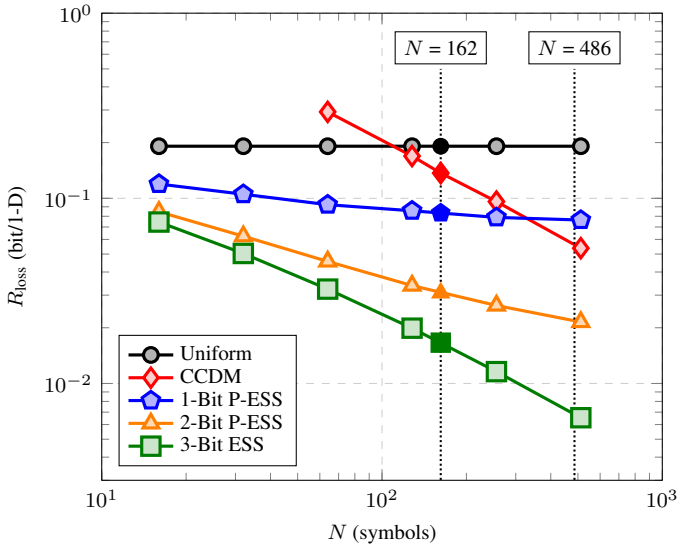

Figure 8.4: R_{loss} vs. blocklength with 16-ASK for various shaping schemes.

beling strategies at the output of the shaper and in the symbol mapper. In this work, we consider BRGCs. Thus, we connect the extra uniform data sequences to the last f bit-levels of 2^m-ASK. Shaped bit levels of 2^{s+1}-ASK are connected to the bit-levels of 2^m-ASK with the same index as shown in Fig. 8.3. We give the following example to clarify this construction.

Example 8.2 (2-bit sphere shaped 16-ASK). Consider a transmission scheme based on 16-ASK, i.e., $m = 4$. To have an $s = 2$-bit shaped output distribution, an enumerative sphere shaper employing the $2^{s+1} = 8$-ASK amplitude alphabet is used. Outputs $\tilde{a}^N$ of this shaper are then labeled with $(\tilde{b}_2, \tilde{b}_3)$ using the mapping

$$1 \to (0,0), \qquad 3 \to (0,1), \qquad 5 \to (1,1), \qquad 7 \to (1,0). \tag{8.3}$$

Then these bit levels $(\tilde{b}_2, \tilde{b}_3)$ are used as the first and second amplitude bit levels (b_2, b_3) of 16-ASK. Next, each label is concatenated with a uniform data bit (b_4) and the result is outputted as the label of an amplitude a from the 16-ASK alphabet as in Table 8.2. Note that the shaped bits are (B_2, B_3) and the uniform bit is B_4 in this setting. The distributions of $\tilde{A}$ and A at the outputs of the enumerative shaper and the overall P-ESS block are shown in Fig. 8.3 for $k/N = 1.667$ bit/1-D at $N = 216$

Figure 8.4 shows the rate loss R_{loss} (4.11) versus shaping blocklength N for 1- and 2-bit P-ESS, 3-bit ESS, and CCDM [28] for 16-ASK. For comparison, the same is plotted also for

uniform signaling. The target shaping rate is $R_{\mathrm{s}} = 2.6667$ bit/1-D with 16-ASK. As some of the amplitude bits are kept uniform and independent of the others, R_{loss} converges to a nonzero value for 1- and 2-bit P-ESS, i.e., 0.071 and 0.015 bit/1-D, respectively, unlike the fully shaped schemes. Thus, P-ESS is not an asymptotically optimum shaping architecture according to Definition 4.1. However for this example, although asymptotically optimum, CCDM requires roughly $N > 300$ to surpass 1-bit P-ESS. This shows that shaping some amplitude bits using ESS provides a better rate loss performance than CCDM in the short blocklength regime.

Remark 8.1 (**Alternative partial shaping techniques).** PDM [81] and BL-DM [113] can also be used to shape a subset of the amplitude bit-levels. In [81], the bit-level distributions are optimized such that $\mathbb{E}[X^2]$ is minimized.

8.4 Implementation Aspects and Complexity

8.4.1 Required Storage and Computational Complexity

A straightforward advantage of P-ESS is the decrease in required storage and computational complexity of amplitude shaping. Consider a PAS-based transmission strategy where the rate of the information that is carried by the amplitudes of the channel inputs is $R_{\mathrm{t}} - \gamma$ bit/1-D. If ESS is employed as the amplitude shaping strategy, the rate of the shaper is $k/N = R_{\mathrm{t}} - \gamma$. If on the other hand P-ESS is employed such that f amplitude bits are kept uniform, the rate of the shaper is $k/N = R_{\mathrm{t}} - \gamma - f$. Due to Remark 5.1, the relation between the number of energy levels L_{ess} and $L_{\text{p-ess}}$ that are considered by ESS and P-ESS, resp., can be written as

$$L_{\text{p-ess}} \approx \frac{L_{\mathrm{ess}}}{2^{2f}}. \tag{8.4}$$

Consequently, observing the linear dependence of storage on L as shown in Tables 5.3 and 7.2, a decrease in required storage roughly by a factor of 2^{2f} can be expected when P-ESS is used instead of ESS. Furthermore, in the BP case, the number of bits n_{m} required to store the mantissas may decrease with decreasing $|\mathcal{A}|$ (of the shaper) and L, leading to a further decrease in required storage, and an additional decrease in computational complexity from Table 7.3. In Sec. 8.5, we will provide a numeric example for these effects.

8.4.2 Compatibility of ESS and P-ESS Trellises

Consider the set of energies of the amplitudes from an $M/2$-ASK alphabet which is denoted by $\mathcal{E} = \{e_1, e_2, \cdots, e_{M/4}\}$ where $e_i = (2i-1)^2$. Based on the first-level approximation proposed for M-ASK in Example 8.1, we define the set of average energies of the symbol pairs as $\mathcal{E}_1 = \{e_{1,1}, e_{2,1}, \cdots, e_{M/4,1}\}$ where $e_{j,1} = (1/2) \cdot \{(4j-3)^2 + (4j-1)^2\}$ noting that $e_{j,1} = \mathbb{E}[|\mathcal{A}_j|^2]$. It is then by definition that $e_{l,1} = 4 \cdot e_l + 1$ for $l = 1, 2, \cdots, M/4$. This observation has two consequences:

- The bounded-energy ESS trellises constructed based on $\mathcal{E}$ and $\mathcal{E}_1$ have the same structure, i.e., the connections relating two consecutive columns are identical.
- The MB distribution over $\sqrt{\mathcal{E}}$ for $i = 1, 2, \cdots, M/4$ and the MB distribution over $\sqrt{\mathcal{E}_1}$ for $j = 1, 2, \cdots, M/4$ are the same given that they have the same entropy, where $\sqrt{\mathcal{E}}$ indicates the set of square roots of the elements in $\mathcal{E}$.

Thus, the shaper that is implemented to realize ESS can directly be reused for P-ESS.

8.4.3 PAS with Lower FEC Code Rates

In the PAS scheme, there is a lower bound on the FEC code rate that is $R_\mathrm{c} \geq (m-1)/m$ [13, Sec. IV]. This is because by prescribing the amplitudes at the output of the shaper, $m-1$ bit/1-D are already fixed before FEC coding as shown in Fig. 2.8. Thus, the FEC encoder can at most add 1 bit redundancy per symbol, making the smallest possible code rate $(m-1)/m$. However, when P-ESS is used, only $s < m-1$ of the amplitude bits are fixed by the shaping process. Then, instead of using information bits for the remaining f bit-levels as in Sec. 8.3, we can use the parity added by the encoder. Thus, we can relax the code rate constraint to $R_\mathrm{c} > s/m$. We note that this can also be achieved with bit-level DM [81, 113].

8.5 End-to-end Decoding Performance

In this section, we evaluate the performance of P-ESS in the PAS framework by Monte Carlo simulations. For comparison, uniform signaling, CCDM, and ESS are also simulated. As the channel input constellation, 16-ASK is considered. As in Chapter 5, before transmission over the communication channel, two ASK symbols are combined to a single QAM symbol. The BRGC which is given in Table 8.2 is used for labeling the symbols. As the FEC code, rate-R_c systematic LDPC codes of length $n_\mathrm{c} \in \{648, 1944\}$ bits are used from the IEEE 802.11 standard [16]. Each LDPC codeword corresponds to $N = n_\mathrm{c}/m \in \{162, 486\}$ real symbols where $m = \log_2 16 = 4$. Both 2- and 1-bit P-ESS are considered. The target transmission rate is $R_\mathrm{t} = 3$ bit/1-D. Shaping techniques are coupled with the rate-$R_\mathrm{c} = 5/6$ FEC code leading to $\gamma = 1/3$ where the uniform transmission is with the rate-$R_\mathrm{c} = 3/4$ code. The rate of the amplitude shaping block is $k/N = R_\mathrm{t} - \gamma = 2.667$. For CCDM, the most energy-efficient composition that has at least 2^k sequences is selected. Corresponding shaping parameters are tabulated for $N = 162$ and $N = 486$ in Tables 8.3 and 8.4, respectively.

In Figures 8.5 and 8.6, FER is plotted versus SNR for PAS and uniform signaling at $N = 162$ and $N = 486$, respectively. At $N = 162$, we observe that at an FER of 10^{-3}, ESS performs 1.35 dB more power-efficiently than uniform signaling. Here 2- and 1-bit P-ESS provide 1.27 and 0.95 dB improvement, respectively, while the gain is 0.45 dB for CCDM. At $N = 486$, we observe that at an FER of 10^{-3}, ESS performs 1.25 dB more power-efficiently than uniform signaling. Here 2- and 1-bit P-ESS provide 1.20 and 0.9 dB improvement, respectively, while the gain is 0.97 dB for CCDM. Firstly, all these values roughly match the

Table 8.3: Shaping Parameters for $m = 4$, $N = 162$, $\gamma = 1/3$ and $R_\mathrm{t} = 3$ bit/1-D

Method	s	$E^\bullet$ or C	k/N	E_av	G_s (in dB)
ESS	3	6514	2.667	39.69	1.29
P-ESS	2	1626	1.667	40.73	1.18
P-ESS	1	402	0.667	44.44	0.81
CCDM	3	(34, 32, 28, 23, 18, 13, 9, 5)	2.667	48.31	0.44

Table 8.4: Shaping Parameters for $m = 4$, $N = 486$, $\gamma = 1/3$ and $R_\mathrm{t} = 3$ bit/1-D

Method	s	$E^\bullet$ or C	k/N	E_av	G_s (in dB)
ESS	3	19086	2.6667	39.10	1.36
P-ESS	2	4758	1.6667	40.01	1.26
P-ESS	1	1182	0.6667	43.84	0.86
CCDM	3	(112, 103, 88, 69, 50, 33, 20, 11)	2.6667	42.22	1.02

corresponding shaping gains G_s given in Tables 8.3 and 8.4. Secondly, as claimed following the discussion in Sec. 8.2, 2-bit P-ESS operates very close to the 3-bit ESS, i.e., in its 0.1 dB vicinity. This provides operational evidence for our claim that not all amplitude bits have to be shaped to close most of the shaping gap. We note that the relative performance of all techniques is as predicted by their rate losses in Fig. 8.4.

Finally, the required storage and computational complexity of the ESS-based schemes in Fig. 8.5, i.e., $N = 162$, are tabulated in Table 8.5. Here, the BP ESS implementation explained in Sec. 7.3.2 is employed. We see that by shaping 2 amplitude bits instead of 3, the required storage and computational complexity of shaping can be decreased by factors of 6 and 4, respectively. This reduction is accomplished in the expense of less than 0.1 dB in decoding performance as shown in Figures 8.5 and 8.6. We note that although shaping just 1 amplitude bit provides limited gains, it can be implemented with less than 10 kB storage and with 9 bit oper./1-D. Thus, we conclude that P-ESS provides design flexibility enabling a trade-off between the shaping gain and shaping complexity.

Table 8.5: Required Storage and Computational Complexity

Shaping Technique	Storage $L(N+1)(n_\mathrm{m}+n_\mathrm{p})$	Computation $(\lvert\mathcal{A}\rvert-1)(n_\mathrm{m}+m_\mathrm{a})$
3-bit ESS ($n_\mathrm{m} = 17$, $n_\mathrm{p} = 9$)	421.15 kB	140 bit oper./1-D
2-bit P-ESS ($n_\mathrm{m} = 10$, $n_\mathrm{p} = 9$)	71.23 kB	36 bit oper./1-D
1-bit P-ESS ($n_\mathrm{m} = 8$, $n_\mathrm{p} = 7$)	9.47 kB	9 bit oper./1-D

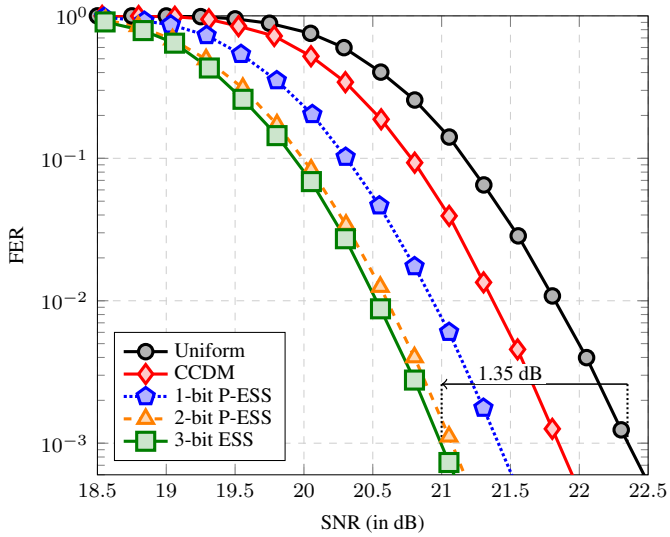

Figure 8.5: 648-bit LDPC-coded FER vs. SNR with 16-ASK at $R_\mathrm{t} = 3$ bit/1-D. The corresponding shaping blocklength $N = 162$.

8.6 Conclusion

In this chapter, we searched for an answer to the following research question.

> **RQ-6** How much do we need to shape the channel input to reap most of the possible shaping gain? Is it possible to obtain a reduction in required storage and computational complexity by realizing a "rough" shaping strategy? How can this rough shaping be realized based on ESS?

We considered a family of approximate MB distributions for shaping the amplitudes, which corresponds to keeping some amplitude bits uniform and independent of the others. Based on these distributions, we demonstrated through a gap-to-capacity analysis that by shaping a couple of amplitude bits of a constellation (as a rule of thumb, two amplitude bits), most of the shaping gap can be closed. Then we proposed partial ESS, a technique to generate amplitude sequences, to shape only a subset of the amplitude bits. We demonstrated using end-to-end decoding simulations that partial ESS performs very close to ESS. This way, the required storage and computational complexity of shaping are significantly reduced.

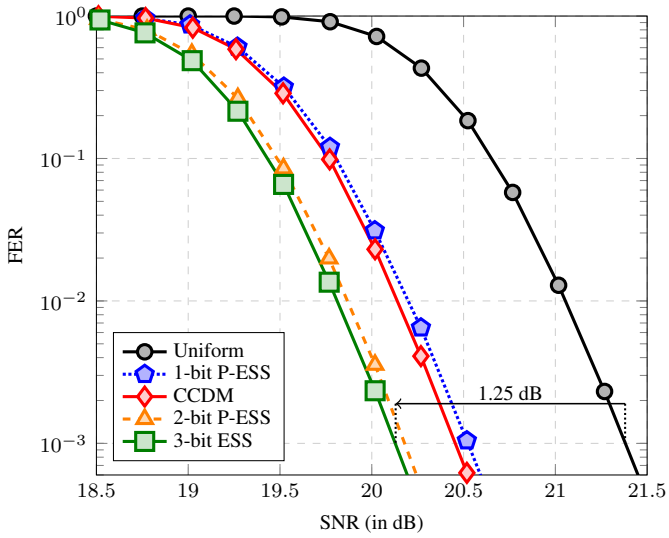

Figure 8.6: 1944-bit LDPC-coded FER vs. SNR with 16-ASK at $R_{\mathrm{t}} = 3$ bit/1-D. The corresponding shaping blocklength $N = 486$.

CHAPTER 9

Summary and Conclusion

In this dissertation, we first studied the achievable information rates (AIRs) of the probabilistic amplitude shaping (PAS) framework. Then we examined sphere shaping in PAS for communication with short blocklengths. In particular, we investigated the enumerative sphere shaping (ESS) technique, we compared ESS with other prominent sphere shaping algorithms, and we proposed low-complexity implementation methods for ESS.

In **Chapter 3**, we addressed the following research question.

> **RQ-1** What are the AIRs of PAS for symbol-metric decoding (SMD) and bit-metric decoding (BMD)? Is it possible to achieve the capacity of memoryless channels with PAS? What are the optimum shaping and coding rates in PAS that maximize AIR gains?

We introduced random sign-coding arguments based on a modified version of weak typicality (β-typicality) that enabled us to compute AIRs of PAS more simply than that followed in the existing literature. In our random sign-coding experiment, the objective was to provide alternative proofs of achievability in which the codes are generated as constructively as possible. Thus, only a fraction γ of the signs of the channel inputs are drawn from a code at random, while their amplitudes and the remaining signs are produced constructively. Unlike most proofs of Shannon's channel coding theorem, how should the code be constructed is (at least partially) self-evident from our proofs. Besides, random sign-coding provided a unified framework in which achievability results can be obtained for all possible PAS settings, i.e., for SMD or BMD, and for basic PAS ($\gamma = 0$) or generalized PAS ($0 < \gamma < 1$). We showed that when SMD is used, the mutual information between the input and output of a memoryless channel is achievable for PAS with uniform signs. This demonstrated that PAS achieves the capacity of memoryless channels if the capacity-achieving distribution is symmetric. When BMD is used, our AIR expression coincides with what was proposed in the literature which is an instance of the so-called LM rate. Finally, we showed that achievability can also be obtained with binary linear codes for both SMD and BMD. The main conclusions of this chapter are as follows:

- Random sign-coding is a unified and simple framework to compute AIRs for PAS.
- As demonstrated earlier by Böcherer [23], PAS is a capacity-achieving coded modulation strategy for memoryless channels.
- AIRs of PAS can be achieved with binary linear codes.

In **Chapter 4**, we searched for an answer to the following research question.

> **RQ-2** What is the "best" amplitude shaping strategy for finite values of the blocklength N? What are the metrics to be used to assess the "goodness" of different amplitude shaping approaches?

We compared constant composition distribution matching (CCDM) with sphere shaping for short blocklengths. We first showed that both techniques are average-energy-optimum for asymptotically large blocklengths. This also implies that sphere shaping induces the Maxwell-Boltzmann distribution over discrete constellations. We then demonstrated that for finite blocklengths, sphere shaping has the minimum rate loss possible, or equivalently, has the maximum shaping gain. Furthermore, the gap between CCDM and sphere shaping in rate loss and in shaping gain becomes significant for short blocklengths. The main conclusion of this chapter is as follows:

- Sphere shaping provides the smallest rate loss and the largest shaping gain at any blocklength.

In **Chapter 5**, we investigated the following research question.

RQ-3 How can sphere shaping be realized algorithmically? Which algorithm provides high performance with low complexity? What is the end-to-end decoding performance of PAS using sphere shaping over the AWGN and frequency selective channels?

We introduced ESS, and we demonstrated its efficiency. We first compared ESS with two different sphere shaping techniques, namely an algorithm by Laroia *et al.* [30, Algorithm 1] (LA1) and shell mapping (SM). We showed that ESS has a significantly smaller complexity than SM due to the use of additions/subtractions instead of multiplications/divisions. ESS also has a slightly smaller complexity than LA1 which requires an extra step to determine the specific N-shell that the amplitude sequence is located on. Then we demonstrated via Monte Carlo simulation that PAS with ESS provides more than 1 dB gain in power-efficiency over uniform signaling for the AWGN channel for a large range of transmission rates and shaping blocklengths. Finally, we showed that PAS with ESS also improves the power-efficiency for frequency selective fading channels, if shaping redundancy is kept relatively small. The main conclusions of this chapter are as follows:

- ESS is an effective amplitude shaping technique for short blocklengths.
- Shaping improves the bandwidth-efficiency of digital communication systems not only for the linear AWGN channel but also for frequency-selective channels.

In **Chapter 6**, we studied the following research question.

RQ-4 Can PAS be incorporated into existing communication systems that are based on the IEEE 802.11 standard? Can PAS be combined with the nonsystematic convolutional codes used in 802.11 [16] which are a mandatory part of the standard?

We demonstrated how to combine amplitude shaping with the nonsystematic convolutional codes used in the IEEE 802.11 standard [16]. We proposed an input selection layer between the shaping and coding layers such that the temporal structure of the amplitude sequences is preserved through nonsystematic encoding. This layer uses the finite state machine model of the following channel code to shuffle amplitude bits in a specific way. Simulation results are then used to show that PAS with ESS and nonsytematic channel codes from 802.11 also provides more than 1 dB gain in power-efficiency for the AWGN channel and frequency selective channels. The main conclusion of this chapter is as follows:

- The nonsystematic convolutional codes from the IEEE 802.11 can be used in the PAS framework to transmit probabilistically shaped channel inputs.

In **Chapter 7**, we examined the following research question.

RQ-5 Can we further improve the energy-efficiency of ESS for practical scenarios? How can ESS be implemented with low storage complexity, minimal computational requirements, and limited latency?

We first proposed an algorithm to compute the amplitude distribution for the operational shaping set of ESS which is a size-2^k subset of the complete sphere for integer k. Then we used this algorithm to show that the difference in the average energy of the operational shaping set is negligible for ESS, LA1, and SM at moderate to long blocklengths. Finally, for short blocklengths where this difference is somewhat significant, we proposed a heuristic method to optimize the ESS trellis. In this way, ESS constructs roughly the most energy-efficient operational shaping set similar to LA1 and SM. We then studied low-complexity implementations of sphere shaping. We first showed that ESS, LA1, and SM can be realized by storing the numbers in their corresponding shaping trellises approximately, i.e., with bounded precision (BP). In this way, the required storage for shaping is decreased. Next, we described sliding-window shaping for BP ESS where only a small part of the input index is required to be considered at any step of the shaping operation. In this way, both the computational complexity of shaping is reduced, and the required arithmetic precision is made fixed and independent of the blocklength. Furthermore, instead of outputting symbols after all input symbols are processed, the shaper/deshaper can now output symbols as soon as the shaping/deshaping operation starts. Finally, we demonstrated that the ESS trellis can be computed on-the-fly during the shaping operation, instead of precomputing and storing it completely, which further decreases the required storage at the expense of increased computational complexity. For this purpose, in the full precision scenario, only the first column needs to be stored, while in the BP case, a few bits per node must additionally be stored for the rest of the trellis. The main conclusions of this chapter are as follows:

- ESS is slightly less energy inefficient with respect to LA1 and SM for short blocklengths. This inefficiency can be removed using a straightforward optimization routine for the ESS trellis.
- Sphere shaping implementation can be tailored to specific constraints imposed by the available hardware resources and/or quality-of-service requirements such as limited memory, restricted computational power, finite arithmetic precision, bounded serialism, and minimal latency.

Finally in **Chapter 8**, we investigated the following research question.

RQ-6 How much do we need to shape the channel input to reap most of the possible shaping gain? Is it possible to obtain a reduction in required storage and computational complexity by realizing a "rough" shaping strategy? How can this rough shaping be realized based on ESS?

We studied the effect of imperfect shaping in terms of AIRs. We showed for large constellations that shaping a subset of the amplitude bits while keeping the remaining bits uniform and independent of the others does not cause a significant penalty on the achievable rates. Then we introduced partial ESS (P-ESS) where ESS is used to shape one or two most significant amplitude bits in the binary labels of the channel input symbols. We showed that for very short blocklengths, even 1-bit P-ESS achieves smaller rate losses than CCDM. We demonstrated via Monte Carlo simulation that the end-to-end decoding performance of PAS with P-ESS is virtually the same as PAS with ESS. The main conclusions of this chapter are as follows:

- As a rule of thumb, shaping more than two amplitude bits of a constellation provides diminishing returns in AIR-sense. Thus, the complexity of shaping can be kept independent of the constellation size, which is especially important for applications with very large constellations such as future digital subscriber line standards [21].
- The ESS algorithm can be used to shape some amplitude bits (P-ESS) without requiring any modification, and thus, single shaping hardware is compatible with both ESS and P-ESS.

References

[1] *Cisco Visual Networking Index: Forecast and Trends, 2017-2022*. Cisco Systems Inc., 2019.

[2] C. E. Shannon, "A mathematical theory of communication," *Bell System Tech. J.*, vol. 27, pp. 379–423, 623–656, July, Oct. 1948.

[3] 3GPP, "3GPP TS 38.104 V.15.2.0: 5G New Radio: Base Station radio transmission and reception," Tech. Spec., July 2018.

[4] T. Kleine-Ostmann and T. Nagatsuma, "A review on terahertz communications research," *J. Infrared. Milli. Terahz. Waves*, vol. 32, pp. 143–171, Jan. 2011.

[5] G. D. Forney, "The viterbi algorithm: A personal history," 2005. [Online]. Available: http://arxiv.org/abs/cs/0504020

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. on Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070 vol.2.

[7] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, Sep. 1960.

[8] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, Mar. 1997.

[9] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. on Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[10] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable signaling on the Gaussian channel," *IEEE Trans. Inf. Theory*, vol. 36, no. 4, pp. 726–740, July 1990.

[11] G. D. Forney, L. Brown, M. V. Eyuboglu, and J. L. Moran, "The V.34 High Speed Modem Standard," *IEEE Commun. Magazine*, vol. 34, no. 12, pp. 28–33, Dec. 1996.

[12] G. Böcherer, "Probabilistic signal shaping for bit-metric decoding," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, June 2014, pp. 431–435.

[13] G. Böcherer, F. Steiner, and P. Schulte, "Bandwidth efficient and rate-matched low-density parity-check coded modulation," *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 4651–4665, Dec. 2015.

[14] W. G. Bliss, "Circuitry for performing error correction calculations on baseband encoded data to eliminate error propagation," *IBM Technol. Disclosure Bulletin*, vol. 23, pp. 4633–4634, Mar. 1981.

[15] J. L. Fan and J. M. Cioffi, "Constrained coding techniques for soft iterative decoders," in *Proc. IEEE Global Commun. Conf.*, Rio de Janeiro, Brazil, Dec. 1999, pp. 723–727.

[16] *IEEE Standard for Inform. Technol.-Telecommun. and Inform. Exchange Between Syst. Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2016 (Revision of IEEE Standard 802.11-2012), Dec. 2016.

[17] T. Fehenberger, A. Alvarado, G. Böcherer, and N. Hanik, "On probabilistic shaping of quadrature amplitude modulation for the nonlinear fiber channel," *J. Lightw. Technol.*, vol. 34, no. 21, pp. 5063–5073, Nov. 2016.

[18] F. Buchali, F. Steiner, G. Böcherer, L. Schmalen, P. Schulte, and W. Idler, "Rate adaptation and reach increase by probabilistically shaped 64-qam: An experimental demonstration," *J. Lightw. Technol.*, vol. 34, no. 7, pp. 1599–1609, Apr. 2016.

[19] W. Idler, F. Buchali, L. Schmalen, E. Lach, R. Braun, G. Böcherer, P. Schulte, and F. Steiner, "Field trial of a 1 tb/s super-channel network using probabilistically shaped constellations," *J. Lightw. Technol.*, vol. 35, no. 8, pp. 1399–1406, Apr. 2017.

[20] Huawei, "3GPP TSG RAN no. 88 R1-1705061: Signal shaping for QAM constellations," Tech. Report, Apr. 2017.

[21] P. Iannone, Y. Lefevre, W. Coomans, D. van Veen, and J. Cho, "Increasing cable bandwidth through probabilistic constellation shaping," *Proc. of the Society of Cable Telecommun. Eng. Int. Soc. of Broadband Experts Technical Forum*, Oct. 2018.

[22] G. Böcherer, "Achievable rates for probabilistic shaping," May 2018. [Online]. Available: http://arxiv.org/abs/1707.01134v5

[23] ——, "Principles of coded modulation," Habilitation thesis, Dept. of Electr. and Comput. Eng., Tech. Uni. of Munich, Germany, 2018. [Online]. Available: http://www.georg-boecherer.de/bocherer2018principles.pdf

[24] R. A. Amjad, "Information rates and error exponents for probabilistic amplitude shaping," in *Proc. IEEE Inf. Theory Workshop*, Guangzhou, China, Nov. 2018.

[25] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY, USA: John Wiley & Sons, 1968.

[26] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, 2006.

[27] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: theoretical concepts and practical design rules," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1361–1391, July 1999.

[28] P. Schulte and G. Böcherer, "Constant composition distribution matching," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 430–434, Jan. 2016.

[29] F. Willems and J. Wuijts, "A pragmatic approach to shaped coded modulation," in *Proc. Symp. on Commun. and Veh. Technol. in the Benelux*, Oct. 1993.

[30] R. Laroia, N. Farvardin, and S. A. Tretter, "On optimal shaping of multidimensional constellations," *IEEE Trans. Inf. Theory*, vol. 40, no. 4, pp. 1044–1056, July 1994.

[31] A. Goldsmith, *Wireless Communications*. Cambridge, UK: Cambridge University Press, 2005.

[32] Y. C. Gültekin, W. J. van Houtum, A. Koppelaar, and F. M. J. Willems, "Enumerative sphere shaping for wireless communications with short packets," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1098–1112, Feb. 2020.

[33] D. J. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, UK: Cambridge University Press, 2003.

[34] Y. C. Gültekin, A. Alvarado, and F. M. J. Willems, "Achievable information rates for probabilistic amplitude shaping: An alternative approach via random sign-coding arguments," *Entropy*, vol. 22, no. 7: 762, July 2020.

[35] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York, NY, USA: John Wiley & Sons, 1965.

[36] Y. C. Gültekin, W. J. van Houtum, and F. M. J. Willems, "On constellation shaping for short block lengths," in *Proc. Symp. on Inf. Theory and Signal Process. in the Benelux (SITB)*, Enschede, The Netherlands, June 2018, pp. 86–96.

[37] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. on Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[38] L. Szczecinski and A. Alvarado, *Bit-Interleaved Coded Modulation: Fundamentals, Analysis, and Design*. Chichester, UK: John Wiley & Sons, 2015.

[39] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 371–377, May 1977.

[40] G. Ungerböck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inf. Theory*, vol. 28, no. 1, pp. 55–67, Jan. 1982.

[41] E. Zehavi, "8-psk trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, May 1992.

[42] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. on Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[43] A. Guillén i Fàbregas, A. Martinez, and G. Caire, "Bit-interleaved coded modulation," *Found. Trends Commun. Inf. Theory*, vol. 5, no. 1-2, pp. 1–153, Nov. 2008.

[44] A. Martinez, Guillén i Fàbregas, G. Caire, and F. M. J. Willems, "Bit-interleaved coded modulation revisited: A mismatched decoding perspective," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2756–2765, June 2009.

[45] R. Fischer, *Precoding and Signal Shaping for Digital Transmission*. New York, NY, USA: John Wiley & Sons, 2002.

[46] G. Forney, R. Gallager, G. Lang, F. Longstaff, and S. Qureshi, "Efficient modulation for band-limited channels," *IEEE J. Sel. Areas Commun.*, vol. 2, no. 5, pp. 632–647, Sep. 1984.

[47] 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.

[48] Feng-Wen Sun and H. C. A. van Tilborg, "Approaching capacity by equiprobable signaling on the Gaussian channel," *IEEE Trans. Inf. Theory*, vol. 39, no. 5, pp. 1714–1716, Sep. 1993.

[49] N. S. Loghin, J. Zöllner, B. Mouhouche, D. Ansorregui, J. Kim, and S. Park, "Non-uniform constellations for ATSC 3.0," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 197–203, Mar. 2016.

[50] Z. Qu and I. B. Djordjevic, "Geometrically shaped 16QAM outperforming probabilistically shaped 16QAM," in *Proc. Eur. Conf. Opt. Commun.*, Gothenburg, Sweden, Sep. 2017.

[51] F. Steiner and G. Böcherer, "Comparison of geometric and probabilistic shaping with application to ATSC 3.0," in *Proc. ITG Conf. on Syst., Commun. and Coding*, Feb. 2017.

[52] J. J. Boutros, U. Erez, J. V. Wonterghem, G. I. Shamir, and G. Zèmorl, "Geometric shaping: low-density coding of Gaussian-like constellations," in *Proc. IEEE Inf. Theory Workshop*, Nov. 2018.

[53] B. Chen, C. Okonkwo, H. Hafermann, and A. Alvarado, "Increasing achievable information rates via geometric shaping," in *Proc. Eur. Conf. Opt. Commun.*, Roma, Italy, Sep. 2018.

[54] B. Chen, C. Okonkwo, D. Lavery, and A. Alvarado, "Geometrically-shaped 64-point constellations via achievable information rates," in *Proc. Int. Conf. on Transparent Opt. Netw.*, Bucharest, Romania, July 2018.

[55] B. Chen, Y. Lei, D. Lavery, C. Okonkwo, and A. Alvarado, "Rate-adaptive coded modulation with geometrically-shaped constellations," in *Proc. Asia Commun. and Photon. Conf.*, Hangzhou, China, Oct. 2018.

[56] G. D. Forney, "Trellis shaping," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 281–300, Mar. 1992.

[57] G. Böcherer and R. Mathar, "Matching Dyadic Distributions to Channels," in *Proc. Data Compression Conf.*, Snowbird, UT, USA, Mar. 2011, pp. 23–32.

[58] H. G. Batshon, M. V. Mazurczyk, J. Cai, O. V. Sinkin, M. Paskov, C. R. Davidson, D. Wang, M. Bolshtyansky, and D. Foursa, "Coded modulation based on 56APSK with hybrid shaping for high spectral efficiency transmission," in *Proc. Eur. Conf. Opt. Commun.*, Gothenburg, Sweden, Sep. 2017.

[59] J.-X. Cai, H. G. Batshon, M. V. Mazurczyk, O. V. Sinkin, D. Wang, M. Paskov, W. W. Patterson, C. R. Davidson, P. C. Corbett, G. M. Wolter *et al.*, "70.46 Tb/s Over 7,600 km and 71.65 Tb/s Over 6,970 km Transmission in C+L Band Using Coded Modulation With Hybrid Constellation Shaping and Nonlinearity Compensation," *J. Lightw. Technol.*, vol. 36, pp. 114–121, Jan. 2018.

[60] J. Cai, H. G. Batshon, M. V. Mazurczyk, O. V. Sinkin, D. Wang, M. Paskov, C. R. Davidson, W. W. Patterson, A. Turukhin, M. A. Bolshtyansky, and D. G. Foursa, "51.5 Tb/s Capacity over 17,107 km in C+L Bandwidth Using Single-Mode Fibers and Nonlinearity Compensation," *J. Lightw. Technol.*, vol. 36, no. 11, pp. 2135–2141, June 2018.

[61] D. Sommer and G. P. Fettweis, "Signal shaping by non-uniform QAM for AWGN channels and applications using turbo coding," in *Proc. ITG Conf. on Source and Channel Coding*, Munich, Germany, Jan. 2000.

[62] S. Y. Le Goff, "Signal constellations for bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 307–313, Jan. 2003.

[63] M. F. Barsoum, C. Jones, and M. Fitz, "Constellation design via capacity maximization," in *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, June 2007, pp. 1821–1825.

[64] S. Y. Le Goff, B. S. Sharif, and S. A. Jimaa, "A new bit-interleaved coded modulation scheme using shaping coding," in *Proc. IEEE Global Commun. Conf.*, Dallas, TX, USA, Nov.-Dec. 2004.

[65] D. Raphaeli and A. Gurevitz, "Constellation shaping for pragmatic turbo-coded modulation with high spectral efficiency," *IEEE Trans. Commun.*, vol. 52, no. 3, pp. 341–345, Mar. 2004.

[66] S. Y. Le Goff, B. S. Sharif, and S. A. Jimaa, "Bit-interleaved turbo-coded modulation using shaping coding," *IEEE Commun. Lett.*, vol. 9, no. 3, pp. 246–248, Mar. 2005.

[67] M. C. Valenti and X. Xiang, "Constellation shaping for bit-interleaved LDPC coded APSK," *IEEE Trans. Commun.*, vol. 60, no. 10, pp. 2960–2970, Oct. 2012.

[68] 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, pp. 2223–2233, June 2007.

[69] A. Guillén i Fàbregas and A. Martinez, "Bit-interleaved coded modulation with shaping," in *Proc. IEEE Inf. Theory Workshop*, Dublin, Ireland, Aug.-Sep. 2010.

[70] A. Alvarado, F. Brännström, and E. Agrell, "High SNR bounds for the BICM capacity," in *Proc. IEEE Inf. Theory Workshop*, Paraty, Brazil, Oct. 2011, pp. 360–364.

[71] G. Böcherer, F. Altenbach, A. Alvarado, S. Corroy, and R. Mathar, "An efficient algorithm to calculate BICM capacity," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, July 2012, pp. 309–313.

[72] L. Peng, A. Guillén i Fàbregas, and A. Martinez, "Mismatched shaping schemes for bit-interleaved coded modulation," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, July 2012.

[73] L. Peng, "Fundamentals of bit-interleaved coded modulation and reliable source transmission," Ph.D. dissertation, University of Cambridge, Cambridge, UK, Dec. 2012.

[74] E. Agrell and A. Alvarado, "Signal shaping for BICM at low SNR," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2396–2410, Apr. 2013.

[75] B. S. Bouazza and A. Djebari, "Bit-interleaved coded modulation with iterative decoding using constellation shaping over Rayleigh fading channels," *AEÜ Int. J. of Electron. and Commun.*, vol. 61, no. 6, pp. 405–410, June 2007.

[76] X. Xiang and M. C. Valenti, "Improving DVB-S2 performance through constellation shaping and iterative demapping," in *Proc. Military Commun. Conf.*, Baltimore, MD, USA, Nov. 2011, pp. 549–554.

[77] *Digital Video Broadcasting (DVB); 2nd Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications (DVB-S2)*, European Telecommun. Standards Inst. (ETSI) Standard EN 302 307, Rev. 1.2.1, 2009.

[78] T. Prinz, P. Yuan, G. Böcherer, F. Steiner, O. İşcan, R. Böhnke, and W. Xu, "Polar coded probabilistic amplitude shaping for short packets," in *Proc. Int. Workshop on Signal Process. Advances in Wireless Commun.*, July 2017.

[79] T. Fehenberger, D. Lavery, R. Maher, A. Alvarado, P. Bayvel, and N. Hanik, "Sensitivity gains by mismatched probabilistic shaping for optical communication systems," *IEEE Photon. Technol. Lett.*, vol. 28, no. 7, pp. 786–789, Apr. 2016.

[80] A. Amari, S. Goossens, Y. C. Gültekin, O. Vassilieva, I. Kim, T. Ikeuchi, C. Okonkwo, F. M. J. Willems, and A. Alvarado, "Introducing enumerative sphere shaping for optical communication systems with short blocklengths," *J. Lightw. Technol.*, vol. 37, no. 23, pp. 5926–5936, Dec. 2019.

[81] F. Steiner, P. Schulte, and G. Böcherer, "Approaching waterfilling capacity of parallel channels by higher order modulation and probabilistic amplitude shaping," in *Proc. Conf. on Inf. Syst. and Sci.*, Princeton, NJ, USA, Mar. 2018.

[82] Y. C. Gültekin, W. J. van Houtum, S. Şerbetli, and F. M. J. Willems, "Constellation shaping for IEEE 802.11," in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Commun.*, Montreal, QC, Canada, Oct. 2017.

[83] Y. C. Gültekin, W. J. van Houtum, A. Koppelaar, and F. M. J. Willems, "Partial enumerative sphere shaping," in *Proc. IEEE Veh. Technol. Conf. (Fall)*, Honolulu, HI, USA, Sep. 2019.

[84] G. Kaplan and S. Shamai (Shitz), "Information rates and error exponents of compound channels with application to antipodal signaling in a fading environment," *AËU*, vol. 47, no. 4, pp. 228–239, 1993.

[85] N. Merhav, G. Kaplan, A. Lapidoth, and S. Shamai (Shitz), "On information rates for mismatched decoders," *IEEE Trans. on Inf. Theory*, vol. 40, no. 6, pp. 1953–1967, Nov. 1994.

[86] G. Böcherer, "Probabilistic signal shaping for bit-metric decoding," Apr. 2014. [Online]. Available: http://arxiv.org/abs/1401.6190

[87] G. Kramer, "Topics in multi-user information theory," *Found. Trends Commun. Inf. Theory*, vol. 4, no. 4-5, pp. 265–444, June 2008.

[88] G. Böcherer, "Achievable rates for shaped bit-metric decoding," May 2016. [Online]. Available: http://arxiv.org/abs/1410.8075v6

[89] R. A. Amjad, "Information rates and error exponents for probabilistic amplitude shaping," June 2018. [Online]. Available: https://arxiv.org/abs/1802.05973

[90] S. M. Moser. (2020). Information theory (Lecture Notes, version 6.7). ETH Zürich, Switzerland.

[91] N. Shulman and M. Feder, "Random coding techniques for nonrandom codes," *IEEE Trans. on Inf. Theory*, vol. 45, no. 6, pp. 2101–2104, Sep. 1999.

[92] R. Yeung, *Information Theory and Network Coding*. Boston, MA, USA: Springer, 2008.

[93] T. Fehenberger, D. S. Millar, T. Koike-Akino, K. Kojima, and K. Parsons, "Multiset-partition distribution matching," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 1885–1893, Mar. 2019.

[94] R. F. H. Fischer, J. B. Huber, and U. Wachsmann, "Multilevel coding: aspects from information theory," in *Proc. IEEE Global Commun. Conf.*, London, UK, Nov. 1996.

[95] J. Cho, "Prefix-free code distribution matching for probabilistic constellation shaping," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 670–682, Feb. 2020.

[96] G. Böcherer and B. C. Geiger, "Optimal Quantization for Distribution Synthesis," *IEEE Trans. on Inf. Theory*, vol. 62, no. 11, pp. 6162–6172, Nov. 2016.

[97] P. Schulte and F. Steiner, "Divergence-optimal fixed-to-fixed length distribution matching with shell mapping," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 620–623, Apr. 2019.

[98] P. Schulte, "Algorithms for distribution matching," Ph.D. dissertation, Tech. Uni. of Munich, Germany, Apr. 2020.

[99] T. Cover, "Enumerative source encoding," *IEEE Trans. on Inf. Theory*, vol. 19, no. 1, pp. 73–77, Jan. 1973.

[100] G. R. Lang and F. M. Longstaff, "A leech lattice modem," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 6, pp. 968–973, Aug. 1989.

[101] S. A. Tretter, *Constellation Shaping, Nonlinear Precoding, and Trellis Coding for Voiceband Telephone Channel Modems: with Emphasis on ITU-T Recommendation V.34*. New York, NY, USA: Springer, 2002.

[102] Y. C. Gültekin, F. M. J. Willems, W. J. van Houtum, and S. Şerbetli, "Approximate enumerative sphere shaping," in *Proc. IEEE Int. Symp. Inf. Theory*, Vail, CO, USA, June 2018, pp. 676–680.

[103] O. Vassilieva, I. Kim, and T. Ikeuchi, "On the Fairness of the Performance Evaluation of Probabilistically Shaped QAM," in *Proc. Eur. Conf. Opt. Commun.*, Dublin, Ireland, Sep. 2019, (to appear).

[104] J. Medbo and P. Schramm, "Channel models for HIPERLAN/2," ETSI/BRAN doc. no. 3ERI085B, 1998.

[105] D. S. Millar, T. Fehenberger, T. Koike-Akino, K. Kojima, and K. Parsons, "Distribution matching for high spectral efficiency optical communication with multiset partitions," *J. Lightw. Technol.*, vol. 37, no. 2, pp. 517–523, Jan. 2019.

[106] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*. New York, NY, USA: Wiley, 2015.

[107] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2004.

[108] M. Bossert, *Channel Coding for Telecommunications*. New York, NY, USA: Wiley, 1999.

[109] K. A. S. Immink, "A practical method for approaching the channel capacity of constrained channels," *IEEE Trans. on Inf. Theory*, vol. 43, no. 5, pp. 1389–1399, Sep. 1997.

[110] Y. C. Gültekin and F. M. J. Willems, "Building the optimum enumerative shaping trellis," in *Proc. Symp. on Inf. Theory and Signal Process. in the Benelux (SITB)*, Gent, Belgium, May 2019, p. 34, (abstract).

[111] R. F. H. Fischer, "Calculation of shell frequency distributions obtained with shell-mapping schemes," *IEEE Trans. on Inf. Theory*, vol. 45, no. 5, pp. 1631–1639, July 1999.

[112] G. N. N. Martin, G. G. Langdon, and S. J. P. Todd, "Arithmetic codes for constrained channels," *IBM J. of Research and Develop.*, vol. 27, no. 2, pp. 94–106, Mar. 1983.

[113] M. Pikus and W. Xu, "Bit-level probabilistically shaped coded modulation," *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 1929–1932, Sep. 2017.

About the Author

Yunus Can Gültekin was born on July 5, 1991, in İzmir, Turkey. He received the B.Sc. and M.Sc. degrees in Electrical and Electronics Engineering from the Middle East Technical University, Ankara, Turkey, in 2013 and 2015, respectively. He completed the M.Sc. program in the Telecommunications group with the thesis entitled "OFDMA Based Device-to-device Communication Protocols".

In January 2016, Yunus started working towards a Ph.D. degree in the Signal Processing Systems group at the Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands, under the supervision of Frans Willems. His research mainly focused on constellation shaping architectures for communication systems and supported by NXP Semiconductors. This thesis includes some of the main results of this Ph.D. research.

Since 2020, Yunus is working as a researcher in the Information and Communication Theory Lab at the TU/e. His research interests include the design of coded modulation systems and constellation shaping techniques.

List of Publications

Journal Articles

- **Y. C. Gültekin**, W. J. van Houtum, A. G. C. Koppelaar, and F. M. J. Willems, "Comparison and optimization of enumerative coding techniques for amplitude shaping," Nov. 2020. (submitted to IEEE Commun. Lett.)
- **Y. C. Gültekin**, W. J. van Houtum, A. G. C. Koppelaar, and F. M. J. Willems, "Low-complexity enumerative coding techniques with applications to amplitude shaping," *IEEE Commun. Lett.*, Sep. 2020.
- **Y. C. Gültekin**, A. Alvarado, and F. M. J. Willems, "Achievable information rates for probabilistic amplitude shaping: An alternative approach via random sign-coding arguments," *Entropy*, vol. 22, no. 7: 762, July 2020.
- **Y. C. Gültekin**, T. Fehenberger, A. Alvarado, and F. M. J. Willems, "Probabilistic shaping for finite blocklengths: Distribution matching and sphere shaping," *Entropy*, vol. 19, no. 5: 581, May 2020.
- **Y. C. Gültekin**, W. J. van Houtum, A. G. C. Koppelaar, and F. M. J. Willems, "Enumerative sphere shaping for wireless communications with Short Packets," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1098-1112, Feb. 2020.

Conference Articles

- **Y. C. Gültekin**, A. Alvarado, and F. M. J. Willems, "Achievable information rates of probabilistic amplitude shaping: An alternative approach via random sign-coding arguments," *in Proc. Int. Zurich Seminar on Inf. and Commun. (IZS)*, Zurich, Switzerland, Feb. 2020. (Abstract & poster presentation)

- **Y. C. Gültekin**, W. J. van Houtum, A. G. C. Koppelaar, and F.M.J. Willems, "Partial Enumerative Sphere Shaping," *in Proc. IEEE Veh. Technol. Conf. (VTC-Fall)*, Honolulu, HI, USA, Sep. 2019.
- **Y. C. Gültekin** and F. M. J. Willems, "Building the optimum enumerative shaping trellis," *in Proc. Symp. on Inf. Theory and Signal Process. in the Benelux (SITB)*, Ghent, Belgium, May 2019. (Abstract & poster presentation)
- **Y. C. Gültekin**, F. M. J. Willems, W. J. van Houtum, and S. Şerbetli, "Approximate Enumerative Sphere Shaping," *in Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, June 2018.
- **Y. C. Gültekin**, W. J. van Houtum, and F. M. J. Willems, "On Constellation Shaping for Short Block Lengths," *in Proc. Symp. on Inf. Theory and Signal Process. in the Benelux (SITB)*, Enschede, The Netherlands, June 2018.[1]
- **Y. C. Gültekin**, W. J. van Houtum, S. Şerbetli, and F. M. J. Willems, "Constellation shaping for IEEE 802.11," *in Proc. IEEE Int. Symp. Personal, Indoor and Mobile Commun. (PIMRC)*, Montreal, QC, Canada, Oct. 2017.
- **Y. C. Gültekin** and F. M. J. Willems, "Comparison of enumerative and probabilistic shaping for short block lengths," *Eur. School of Inf. Theory (ESIT)*, Madrid, Spain, May 2017. (Abstract submission & poster presentation)

Patents

- **Y. C. Gültekin**, F. M. J. Willems, W. J. van Houtum, and S. Şerbetli, "Encoder input selector," U.S. Patent 10 530 630 B2, Jan., 7, 2020.
- **Y. C. Gültekin**, F. M. J. Willems, W. J. van Houtum and S. Şerbetli, "Approximate enumerative sphere shaping," U.S. Patent 10 523 474 B1, Dec., 31, 2019.
- **Y. C. Gültekin**, F. M. J. Willems, W. J. van Houtum, and S. Şerbetli, "K-bit enumerative sphere shaping of multidimensional constellations," U.S. Patent 10 523 480 B1, Dec., 31, 2019.

Not Included in This Thesis

- A. Amari, L. Lampe, S. K. O. Soman, **Y. C. Gültekin**, and A. Alvarado, "Comparison of short blocklength sphere shaping and nonlinearity compensation in WDM systems," *IEEE Photon. Technol. Lett.*, vol. 32, no. 22, pp. 1435-1438, Nov. 2020.

[1] Best young PhD researcher paper award

- A. Amari, S. Goossens, **Y. C. Gültekin**, O. Vassilieva, I. Kim, T. Ikeuchi, C. Okonkwo, F. M. J. Willems, and A. Alvarado, "Introducing enumerative sphere shaping for optical communication systems with short blocklengths," *J. Lightw. Technol.*, vol. 37, no. 23, pp. 5926–5936, Dec. 2019.

- A. Amari, S. Goossens, **Y. C. Gültekin**, O. Vassilieva, I. Kim, T. Ikeuchi, C. Okonkwo, F. M. J. Willems, and A. Alvarado, "Enumerative sphere shaping for rate adaptation and reach increase in WDM transmission systems," *in Proc. Eur. Conf. Opt. Commun. (ECOC)*, Dublin, Ireland, Sep. 2019.

- S. Goossens, S. van der Heide, M. van den Hout, A. Amari, **Y. C. Gültekin**, O. Vassilieva, I. Kim, T. Ikeuchi, F. M. J. Willems, A. Alvarado, and C. Okonkwo, "First experimental demonstration of probabilistic enumerative sphere shaping in optical fiber communications," *in Proc. Opto-Electron. and Commun. Conf. and Int. Conf. on Photon. in Switch. and Comput. OECC/PSC)*, Fukuoka, Japan, Jul. 2019. (Postdeadline paper)