GRAND-assisted Optimal Modulation

by

Basak Ozaydin

B.S., Bilkent University (2020)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author . . . .
Department of Electrical Engineering and Computer Science
May 13, 2022

Certified by . . . .
Muriel Médard
Cecil H. Green Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by . . . .
Ken R. Duffy
Director, Hamilton Institute
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

GRAND-assisted Optimal Modulation

by

Basak Ozaydin

Submitted to the Department of Electrical Engineering and Computer Science
on May 13, 2022, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

For Gaussian channels with peak and average power constraints the optimal modulation (OM) schemes are known to have nonuniform probability distributions over the signal points. An established way to obtain these distributions is assigning different number of bits to different constellation points. However, this method leads to challenges in demodulation as if a symbol is identified falsely, due to the different bit lengths of symbols, bit insertions or deletions may occur which may in return cause error propagation. Hence, the difficulty of realizing the channel optimal distributions on constellation signals impeded OM from becoming widely utilized in communication systems. In this thesis, we propose a practical system for OM that uses only a simple padding scheme instead of the complex mechanisms in the current literature. A guess-based error correction demodulator lies at the core of the proposed system. Together with the padding scheme of our choice, our novel light-weight variant of Guessing Random Additive Noise Decoding (GRAND) demodulator protects the system against insertions and deletions. We display that with our approach an overall gain of up to 2 dB in energy per bit over noise spectral density (E_b/N_0) is achievable compared to Quadrature Amplitude Modulation (QAM) with the same number of points.

Thesis Supervisor: Muriel Médard
Title: Cecil H. Green Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Ken R. Duffy
Title: Director, Hamilton Institute

Acknowledgments

I wish to thank both of my supervisors; Prof. Muriel Medard and Prof. Ken Duffy for their guidance and their support throughout this work. Their passion for their work was really both amazing and inspiring.

My mother is my greatest support as in every stage of my life and therefore, I want to thank her for her being the shoulder that I lean on whenever I feel like I need help.

I am deeply grateful to the EECS department, Irwin Mark Jacobs and Joan Klein Jacobs for providing me with my fellowship and DARPA for funding my research afterwards.

Finally, I want to thank the members of past and current members of NCRC, including but not limited to Kathleen Yang, Homa Esfahanizadeh, Amit Solomon, Benoit Pit-Claudel, Alexander Mariona, Rafael G. L. D'Oliveira, as they were with me on this journey.

Contents

1 Introduction 14

1.1 Notation and Channel Model 19

1.2 Background Information 21

1.2.1 Source Encoding 21

1.2.2 Guess-Based Decoding 23

1.2.3 Modulation and Constellation Shaping 25

2 Constellation Design 32

2.1 Approximation to the Cutting Plane Algorithm 33

2.2 Greedy Quantization 36

3 Modulator Design 40

3.1 Power Normalization 41

3.2 Error Aware Mapping 43

3.3 Padding 45

4 Demodulator Design and Length-Based Correction 49

4.1 Decision Regions 50

4.2 Length-Based Correction 53

5 Overall System Performance **58**

6 Conclusion **65**

A Figures **68**

List of Figures

1-1 The constellation on the left is a 64-QAM constellation whereas the constellation on the right is an optimal non-uniform 64-point constellation with the same average signal power. 15

1-2 A Toy Huffman Tree 29

1-3 An illustration for Ungerboeck's padding scheme. The S&P field stores the synchronization bit and the pointer to the beginning of the information frame assigned to the particular symbol frame. 30

2-1 An example 128-point constellation design. 37

3-1 The theoretical performance of the constellations after reverse Huffman mapping 42

3-2 A toy constellation design 46

4-1 The decision regions for the constellation given in the right panel of Fig. 1-1. 52

5-1 SER and BER performances of the overall system in comparison with QAM 59

5-2 Illustration of the change in the attainable mutual information between the channel output and the channel input distribution found during the iterations of the modified cutting plane algorithm and the modified steepest ascend cutting plane algorithm with respect to the iteration number. 60

5-3 BER performance comparison of the 128-point OM design with uncoded 128 QAM and 128-QAM with an additional LDPC code for binary error correction. Due to the varying-length overhead in the proposed scheme, our 128-point design has different rates at different SNRs. The coded 128-QAM results at each SNR are obtained by using LDPC codes of the same average rate as the design at that particular SNR 62

A-1 256-point constellation design 68

A-2 The decision regions for the 128-point constellation design in Fig. 2-1. 69

A-3 The decision regions for the 256-point constellation design in Fig. A-1. 70

List of Tables

4.1 Some Message Lengths for 128-point Constellation Design 55

Chapter 1

Introduction

In 1971, Smith proved that the optimal channel input of a scalar Gaussian channel under peak and average power constraints is a set of discrete points and that the optimal distribution over them can be determined through convex optimization [25]. Analogous results have since been established for other channels [27, 22, 13, 21, 5, 19, 17]. In the complex additive white Gaussian (CAWGN) channel with maximum and average power restrictions, it has been proved that the constellation points in the optimal channel input are discrete in amplitude and continuous in phase (DACP), i.e. they form concentric circles around the origin [22]. Later, it was found that these continuous sets of points can be discretized with negligible loss in performance [14].

Modulation schemes are divided into two classes depending on symbol locations and the probability distribution of their use: uniform and non-uniform. For uniform constellations if binary input data is uniform, each symbol is used equally likely. Moreover, constellation points are symmetric when they are represented on the I-Q plane. In non-uniform constellations, non-uniformity can arise in the frequency of use of the signal points or in asymmetries in the locations of the constellation points

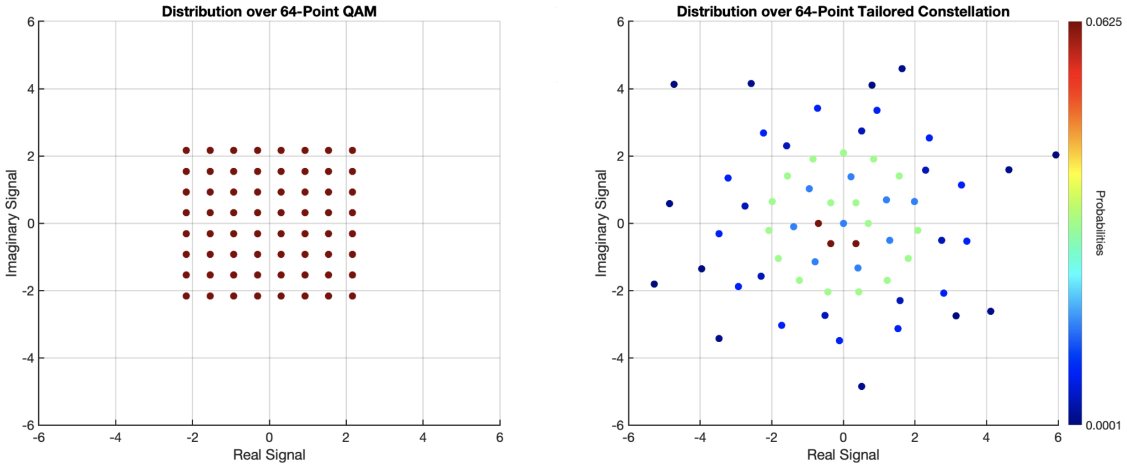

Figure 1-1: The constellation on the left is a 64-QAM constellation whereas the constellation on the right is an optimal non-uniform 64-point constellation with the same average signal power.

on the I-Q plane. Some of the most common schemes are uniform, including QAM and Phase Shift Keying (PSK), as used in applications including 5G, LTE, and IEEE 802.11 [24]. Fig. 1-1 compares these two different constellation structures. Each of these constellations consist of 64 points and the average energies of the signal points in both of the constellations are the same. The probability of each signal point of the nonuniform constellation on the right panel, which is obtained with the methodology described in this thesis, is displayed with a heatmap, whereas the probability of every signal in the 64-QAM constellation on the left panel is 1/64. With the optimal channel input distributions of the CAWGN being nonuniform, studying the methods to identify nonuniform constellations is a worthwhile endeavour. As such, the aim of constellation shaping literature is to obtain enhanced power efficiency, which is achieved by nonuniform constellations.

Methods for identifying improved constellations include dividing constellation points into sublattices and using a binary error-correcting code on top of them

[6, 10, 11], or through varying the lengths of the bit sequences assigned to symbols to approximate target symbol-use distributions [18, 28, 4, 3, 2]. When the latter methods are used they are typically paired with an error correction code [28]. For creating a variable length bit mapping, Huffman shaping is one technique that yields dyadic approximations to the desired target symbol-use probability distribution. In [18], it is shown that this method can get very close to the maximum possible shaping gain of 1.53 dB for unbounded Gaussian channels. Despite the core ideas underlying the design of optimal nonuniform constellations for peak-power constrained channels being known, they are not widely deployed. This is due to practical problems introduced by nonuniform distributions and the methods to identify them, which we discuss next.

In contrast to QAM or PSK, where every symbol is mapped to a fixed number of bits, the main challenge in using bit-symbol mappings of different lengths is that it makes the demodulation vulnerable to insertion and deletion errors. Namely, when using such varying length mappings if a symbol is demodulated as another symbol which is mapped to less or more number of bits, the bit sequence corresponding to the symbols that follows this false demodulation will experience a shift, resulting in error propagation. Hence, some sort of protection in the form of an error correction code needs to be employed in these systems. This results in a large computational overhead. As for the nonuniform constellation design approach with sublattices, one needs to design a suitable code that would yield the channel optimal distribution. In the system that we are proposing with this thesis, we draw inspiration from the recently introduced Guessing Random Additive Noise Decoding (GRAND) algorithm to design a simple insertion-deletion detection and correction system with a considerably small overhead.

By reflecting on the above-discussed previous works, we identify three main challenges for practical OM systems:

A. Designing channel optimal constellations.

B. Constructing a transmitter that shape the probabilities of the transmitted signals according to the optimal channel input design.

C. Devising a receiver that can handle the non-uniformly distributed received signals in such a way that the SER gains that can be obtained via constellation shaping are successfully translated to BER gains.

In this thesis, our contributions to address these challenges and to realize OM are as follows:

I. We propose a new approximation for the cutting plane algorithm introduced in [15] to facilitate the constellation design in channels with high signal-to-noise (SNR) power ratio.

II. We introduce a greedy algorithm for quantizing the continuous energy levels of the optimal channel input.

III. We build a padding scheme based on Huffman shaping to lower the total overhead and system complexity.

IV. We introduce a new light-weight GRAND variant that corrects the length of the transmitted message if an insertion or deletion event occurs.

The items I and II concern challenge A, padding scheme in item III is the change we propose to the existing systems for challenge B and works as a facilitator for item IV. In addition to this, the proposed padding scheme is not a function of the data, resulting in faster and less complex operations than standard binary error correction coding schemes. Most importantly, item IV is the novel approach we bring to challenge C and it converts the insertion and deletion errors of the varying-length constellation shaping methods into a mean of error correction. Translating the SER gains to BER is the primary motivation and the contribution of this paper, as BER is heavily affected from insertions and deletions. The novel scheme we introduce with III and IV mainly focuses on this goal.

The rest of this thesis is organized as follows: In the remaining part of this section, basic underlying concepts are discussed to provide background information in conjunction with how these concepts are utilized. As for the layout of the rest of the sections, we will follow the flow of a traditional communication system. First, in chapter 2, we will elaborate on our approach to optimal constellation design and discuss the design choices for items I and II in sections 2.1 and 2.2, respectively.

The transmitter side of the proposed communication system will be provided in chapter 3. The modulator we propose builds upon the Huffman shaping approach but follows different design choices in three main areas. To begin with, since Huffman shaping approximates a probability distribution that satisfies channel input constraints, there are some optimal distributions where the dyadic approximation obtained via Huffman shaping ends up violating this set of constraints. We discuss how we address such violations in section 3.1. Commonly, the bit mappings for fixed-length mapping constellations are chosen to reduce the possible number of bit errors should a symbol be falsely demodulated. We extend this idea to the variable-length mapping setting through a greedy algorithm in section 3.2. Last but not least, the padding scheme that facilitates GRAND-assisted length correction is introduced in section 3.3.

The novel receiver side that performs guess-based length corrections is presented in chapter 4. Chapter 5 illustrates the gains that the proposed system can achieve as well as provide insight into some interesting future work directions. Finally, we give an overview of the proposed OM scheme and draw a roadmap for the future of this study in chapter 6.

1.1 Notation and Channel Model

Before proceeding with an overview of the prerequisite topics for this thesis, providing the notation that will be used throughout is essential. In line with the convention, random variables are represented by uppercase letters, X; vectors are represented by boldface lowercase letters, $\mathbf{x}$; matrices are represented by boldface uppercase letters, $\mathbf{X}$; and sets are represented by italic uppercase letters, $\mathcal{X}$.

The channel of interest in our constellation and system design is the complex Gaussian channel with average and peak power constraints on the input. Let X and Y be the complex channel input and output respectively and N be a complex Gaussian noise, i.e. $N \sim \mathcal{N}_{\mathbf{C}}(0, N_0)$. Then the channel is modeled as

$$Y = X + N, \tag{1.1}$$

with $E_{f_X(x)}[X^2] \leq \sigma_P^2$ and $|X| \leq M$, where M is the peak power constraint and σ_P^2 is the average power constraint. We will be denoting the set of all possible input distributions of a channel that satisfies the stated average and peak power constraints with $\mathcal{M}$, i.e.,

$$\mathcal{M} = \{f_X(x) : E_{f_X(x)}[X^2] \leq \sigma_P^2,\ |X| \leq M\}. \tag{1.2}$$

N is independent from X and, distributed independent and identically with independent in-phase and quadrature components. The capacity of this channel given by:

$$\begin{aligned} C &= \max_{f_X(x)\in\mathcal{M}} I(X;Y) \quad \text{subject to} \\ & E[X^2] \leq \sigma_P^2, \quad |X| \leq M \end{aligned} \tag{1.3}$$

The complex Gaussian channel is the special case of the Rician channel whose channel equation is given by $Y = (m + A)X + N$ where A and N are independent and identically distributed complex Gaussian variables with $E[|A|^2] = \gamma^2$ and $E[|N|^2] = N_0$. According to [13], the conditional distribution of the channel output given the channel input of a Rician channel is given by

$$f_{Y|X}(y|x) = \frac{1}{\pi\left(\gamma^2|x|^2 + N_0\right)} \exp\left(-\frac{|y - mx|^2}{(\gamma^2|x|^2 + N_0)}\right). \tag{1.4}$$

When A is a deterministic variable, i.e. $\gamma = 0$, and m=1, we obtain the complex Gaussian channel. Then, the conditional distribution of the channel output given the channel input is:

$$f_{Y|X}(y|x) = \frac{1}{\pi N_0} \exp\left(-\frac{|y - x|^2}{N_0}\right). \tag{1.5}$$

Let $X = A_X e^{j\theta}$ and $Y = A_Y e^{j\eta}$. From (1.5), one can readily obtain the conditional distribution of A_Y given A_X [22]:

$$f_{A_Y|A_X}(a_Y|a_X) = \exp\left(-\frac{(a_Y^2 + a_X^2)}{N_0}\right) I_0\left(\frac{2a_Y a_X}{N_0}\right), \tag{1.6}$$

where $I_0(x)$ is the modified Bessel function of the first kind and of 0th order. When the order of the modified Bessel function is an integer, as the case in (1.6), it is expressed as:

$$I_n(z) = \frac{1}{\pi} \int_0^{\pi} e^{z\cos\theta} \cos(n\theta) d\theta \text{ [1]}. \tag{1.7}$$

As a result of this, $I_n(z)$ is a monotonously increasing function and the following lemma provides how fast it increases asymptotically [1].

Lemma 1.1.1. *For large $|z|$ and $\alpha = 4n^2$,*

$$I_n(z) = \frac{e^z}{\sqrt{2\pi z}} \left(1 - \frac{\alpha - 1}{8z} + O\left(z^{-2}\right)\right). \tag{1.8}$$

This asymptotic behaviour is useful in making numerical approximations and performing some simplifications when we are designing close-to-optimal constellations at SNR regimes of engineering interest, which will be detailed in section 2.1.

1.2 Background Information

A conventional communication system consists of three blocks; transmitter, channel and the receiver. The transmitter usually contain three blocks; source encoder, channel encoder, modulator. The receiver, which is the counterpart of the transmitter has demodulator, channel decoder and source decoder blocks, respectively. In this section, we will provide an overview of these conventional blocks of communication systems together with the related assumptions and relations used in this work to provide a background for the reader.

1.2.1 Source Encoding

The output of an information source is a random variable, S. If S is discrete random variable, then the source is referred to as discrete. On the other hand, if S is continuous, then the source is an analog one. In discrete sources S is chosen randomly from an alphabet $\mathcal{S} = \{s_1, \ldots, s_K\}$ according to a certain probability mass function (pmf):

$$P(S = s_k) = p_k \;\; 1 \leq k \leq K, \tag{1.9}$$

where p_k is a valid pmf. A source code is a mapping from $\mathcal{S}$ to a set of finite symbol strings $\mathcal{A}$ coming from a A-ary alphabet [7]. The length of symbols strings in $\mathcal{A}$ may vary. From here on, we will considering the binary alphabet, i.e. $A = 2$. One of the goals of source encoding is compressing the sequence generated by the source to increase transmission efficiency. The following theorem states how much a string of symbols can be compressed without considerable loss of information:

Theorem 1.2.1. *(Strong law of large numbers for incompressible sequences [7]) If a string $x_1 x_2 \ldots$ is incompressible, it satisfies the law of large numbers in the sense that*

$$\frac{1}{n}\sum_{i=1}^{n} x_i \rightarrow \frac{1}{2} \tag{1.10}$$

Hence the proportion of 0s and 1s in any incompressible string are almost equal.

According to Theorem 1.2.1, uniform bit sequences are incompressible. As a result, source encoding results in an almost uniform distribution. Therefore, assuming that the output of the source encoder is a uniformly distributed bit sequence is a common assumption in the existing literature, particularly in works related to Huffman shaping [18, 4]. Hence, while designing the modulator we will assume that the input bit sequence of the modulator is uniformly distributed and each bit independent from other bits, in line with the convention.

1.2.2 Guess-Based Decoding

In this section we will focus on the recently introduced maximum-likelihood decoding scheme GRAND, as the length correction mechanism we introduce in this thesis follows a guess-based approach and is inspired by GRAND, hence it is vital to have an insight on GRAND for the demodulator proposed in this system.

Channel encoder divides the output of the source encoder into sequences of k bit each and maps these k bits to n-bit sequences, i.e. $f : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^n$ where $\mathbb{F}_2$ is the binary field. Each n-bit sequence is referred to as a codeword and the collection of all the codewords is called codebook. In his seminal work, Shannon envisioned using a random channel code and a maximum likelihood decoder and proved that random codes are capacity achieving [23]. However, in traditional communication systems random codes cannot be used as there had been no universal maximum likelihood decoder due to its complexity. Rather than using random codes and maximum likelihood decoders, in practice, codes with certain structures and decoders that are adjusted for the particular choice of code are used.

The recently introduced decoding scheme GRAND is a practical universal maximum likelihood decoder and it can be used with any kind of code as long as it is provided with a method to check whether a bit sequence is in the codebook or not. The main idea behind GRAND and its variants is guessing the additive noise that the channel applies on the transmitted sequence. First, we will discuss GRAND and then we will briefly elaborate on its ORBGRAND variant, which was our inspiration in the design of the demodulator presented in section 4. To perform the guessing operation, GRAND orders possible noise effect sequences from the most likely to the least likely. Starting from the most likely one, it removes the noise effect sequences from the received bit sequence and checks whether the remaining binary string is a member of the codebook. The first codeword that one reaches with this method is the maximum likelihood decoding.

Algorithm 1: ORBGRAND [8]

```
Input: C: code membership function
        T: query threshold
        $y^n$: demodulated bits
        $r^n$:order of bit reliabilities
Result: $c^{*,n}$: decoded codeword
        d: flag to indicate that a decoding is found
        q: number of iterations until a decoding is found
Initialization: q←0, d ← 0;
while $q < T$ do
    $z^n \leftarrow$
     Next most likely noise sequence assuming that greater bit position = greater reliability
    $z_r^n \leftarrow$ Permutation of $z^n$ according to r
    $q \leftarrow q + 1$
    if $C(y^n \ominus z_r^n) == 1$ then
        $c^{*,n} \leftarrow y^n \ominus z_r^n$
        $d \leftarrow 1$
        return
    end
end
```

ORBGRAND is a GRAND variant which uses soft information by calculating the reliability of each received bit. The noise effect sequences are generated by assuming that the first bit is the least reliable bit position and as the bit position increases, the reliabilities the bits increase. Starting from the first generated noise effect sequence, the queried noise effect sequence is permuted according to the bit reliability order. The permuted bit sequence is removed from the hard demodulated signal. Using the codebook membership function of the code, whether the resulting bit sequence is in the codebook or not is checked. The first codeword found this way is the output of the decoder. Alg. 1 gives an outline of ORBGRAND.

In this thesis, we will be introducing a light-weight GRAND variant that considers noise effects on symbol-level and corrects one symbol error in an attempt to correct length changes that occur due to insertions and deletions, while doing so, the demodulator utilizes soft information through calculating reliabilities of the symbols.

1.2.3 Modulation and Constellation Shaping

Modulation is the process of mapping bit sequences to a set of waveforms, $\mathcal{S} = \{s_m(t), t = 1, 2, \ldots, M\}$, that can be transmitted across the channel [20]. $s_m(t)$ for $m = 1, 2, \ldots, M$ can possibly be a set of scalar signals or a set of complex signals. When $\mathcal{S}$ is a set of complex signals, the signals within this set can be easily represented in a vector form by using

$$s_i(t) = \sqrt{\frac{2}{\varepsilon_g}} g(t) cos(2\pi f_c t) \text{ and } s_q(t) = \sqrt{\frac{2}{\varepsilon_g}} g(t) sin(2\pi f_c t), \tag{1.11}$$

as the basis of the signal space spanned by $\mathcal{S}$ where ε_g is the energy of the signal g(t). In such a case, for all $m = 1, 2, \ldots, M$, $s_m(t)$ corresponds to vector with 2 components; $\mathbf{s_m} = \begin{bmatrix} s_m^i \\ s_m^q \end{bmatrix}$. s_m^i and s_m^q are referred to as the in-phase and the quadrature components. Every $\mathbf{s_m}$, therefore represents a point on the I-Q plane whose coordinates are (s_m^i, s_m^q). The set of points, $\{\mathbf{s_m} | m = 1, 2, \ldots, M\}$ is called the modulation constellation.

As previously mentioned, the nonuniformity within a constellation can be achieved through the probability distribution over the constellation points and/or the locations of constellation points. Correspondingly, there are two approaches in constellation shaping; geometric shaping strives for nonuniform signal intervals, i.e. by modifying points of the constellation, $\mathbf{s_m}$, whereas probabilistic shaping matches the distribution of the constellation to a particular nonuniform distribution, i.e. it modifies the frequency of use of constellation points $\mathbf{s_m}$.

In this thesis, we perform both geometric shaping and probabilistic shaping; the locations of the channel optimal modulation schemes turn out to be non-symmetric with respect to the axis due the quantization step that the optimal DACP distributions go through. In addition to this, the quantization of the DACP distributions yield a nonuniform probability distribution over the constellation points, hence probabilistic shaping is also required to realize these distributions over the geometrically shaped constellations. We will focus on Huffman shaping mentioned in [18, 28], for shaping constellations probabilistically.

The premise of Huffman shaping is simple and it relies on reversing Huffman coding. Huffman coding is a source coding algorithm that produces an optimal code according to the following procedure [12]: Let $\mathcal{S}$ be the alphabet that will be encoded and consider each element of $\mathcal{S}$ as the leaf nodes of a binary code tree. Huffman algorithm arranges each symbol from the most likely to the least likely, then chooses the last two in this list and creates a new internal node whose probability is the sum of the probabilities of these two least probable nodes. These three nodes are connected to each other such that the new one is the parent of the other two. The two old nodes are removed from the nodes list and the new node is added to it. In the next iteration, again the nodes with the smallest probabilities are chosen from the list and connected to a parent node as described. This processes is repeated until only one node remains in the list, and this last-standing node becomes the root of the binary code tree. By assigning the branches "0" and "1", one can obtain the optimal code for the given discrete distribution. Alg. 2 describes this procedure, and Fig. 1-2 provides a code tree built using the Huffman algorithm.

Huffman coding assembles the Huffman tree according to the above-given pro-

Algorithm 2: Huffman Tree Generation

```
Input: X: alphabet of possible symbols
       P: vector of probabilities corresponding to the symbols in X
Result: Huffman tree: a binary tree in which each branch represents a bit
Initialization: Create nodes for each symbol with their corresponding
  probabilities
                Enumerate each node and place the nodes into an array
repeat
    i) Find the 2 nodes with smallest probability and remove them from the
       list
    ii) Create a new parent node with probability equal to the sum of the 2
        chosen nodes
    iii) Add the new node to the list, set this new node as the parent of the 2
         nodes
until Until a single node remains;
```

cedure and then it matches symbols with bit sequences by reading the bits on the branches. For instance, after building the tree in Fig. 1-2, symbol number 3 is mapped to the bit sequence "10", and symbol number 5 is mapped to "0010". This method yields a 1-1 mapping between the symbols and the bit sequences. Huffman shaping reverses this mapping, namely instead of assigning bit sequences to symbols, it assigns signals to bit sequences entering the modulator. After having grown the binary code tree, the binary input of the modulator is followed through the branches of the Huffman tree to decide what symbols correspond to the received input bit sequence. Tracking the bits along branches leads to a leaf node. When a leaf node is reached, the symbol corresponding to that particular node is transmitted, and then to map the remaining bits, the procedure returns back to the root of the tree and again follows bits until reaching a leaf. This process terminates when there is no longer any bits that need to be mapped to symbols.

In this work, we chose to model the bit sequence coming to the modulator as a fixed number, i.e. the message lengths are fixed. Consider the tree in Fig. 1-2 and suppose that from the channel encoder, the bit sequence “10011101” has arrived at the modulator. Starting from the root node, node number 31, we follow the bits on the branches. The first received bit is 1, advancing along the branch that represents 1, we arrive at node 29. Carrying on in this way, we see that after the first 2 bits, we end up at symbol 3. Hence, the first symbol that the transmitter sends is symbol 3. The first bit after we arrive at the leaf corresponding to the third symbol is 0. This time starting from node 31, we move along branch 0 to arrive at node 30. The substring “0111” leads us to symbol 1. So, after 3, we transmit 1. The remaining substring after attaining two symbols is “01”. From Fig. 1-2, one can see that this last bit sequence does not reach any of the leaves. Thus, the algorithm to map bit sequences to symbols needs one more adjustment. This topic will be covered with the padding scheme that will be presented in section 3.3.

This method yields the dyadic approximation of the optimal distribution. We assume that the message bits are independent and identically distributed and each bit is 0 with probability 1/2 and 1 with probability 1/2, as we previously discussed in the source encoding section. So, when moving along the branches towards the leaves, each branch adds a factor of 1/2 to the probability of the arrived leaf node. As an example, consider Fig. 1-2; in the final distribution $p_X(2) = 1/4$ and $p_X(16) = 1/128$, as the distances of the leaf nodes corresponding to these symbols from the root are 2 and 7 respectively.

One drawback of this method for obtaining nonuniform constellations is the fact that there may be insertions and deletions and hence, bit errors may propagate after a false demodulation. Some padding schemes were previously studied such as the padding scheme proposed by Ungerboeck that relies on synchronization and pointers [28] and is as illustrated as in Fig.1-3. However, we observed that with an

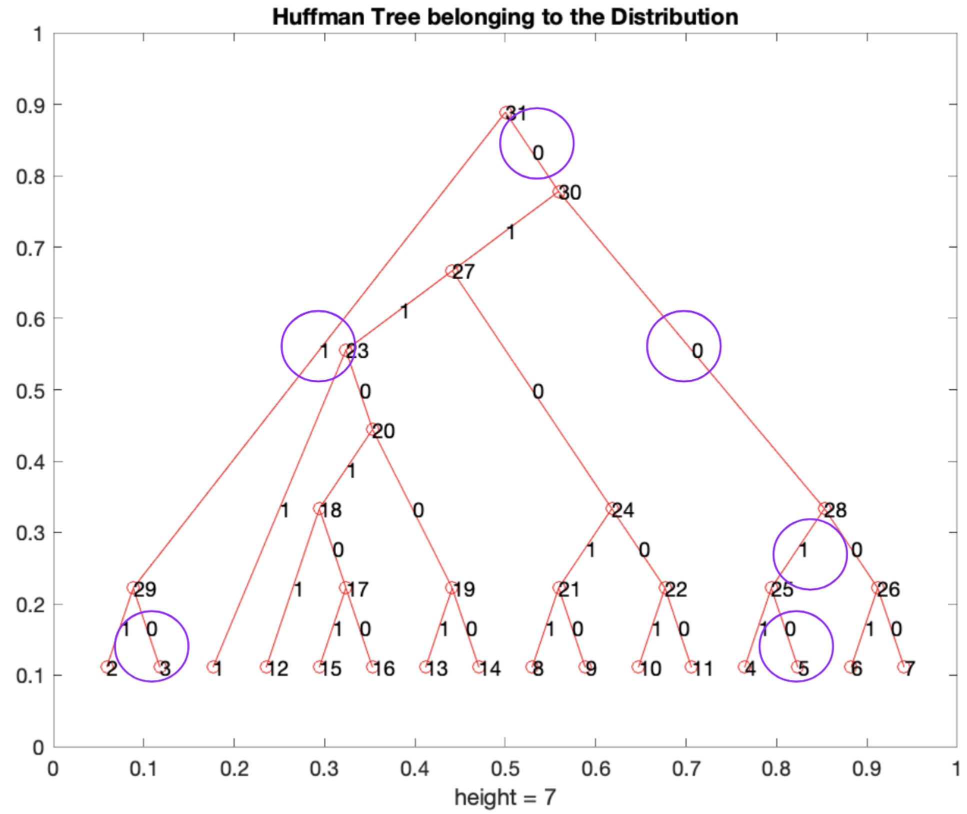

Figure 1-2: A Toy Huffman Tree

average of 3.5 overhead bits, the padding scheme that we propose in this thesis is significantly more efficient. In Ungerboeck's padding scheme symbols are transmitted in groups of N_s symbols and each N_s-symbol frame is assigned to a N_b-bit information frame. At the beginning of each symbol frame, there are overhead bits reserved to synchronization and a pointer. Frames may contain bits belonging to the information frames of the previous symbol frames due to the variable-length mappings of Huffman shaping. Thus, pointer field of a symbol frame points to the location of the beginning of the bit frame associated with that particular symbol frame. With one bit being reserved to synchronization, for Ungerboeck's padding scheme to be more efficient

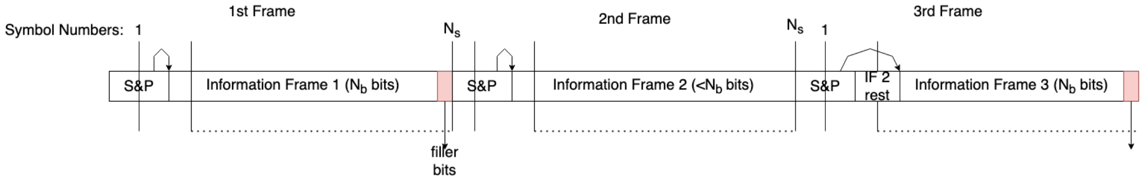
Symbol Numbers: 1
1st Frame
Ns
2nd Frame
Ns
1
3rd Frame
S&P
Information Frame 1 (Nb bits)
S&P
Information Frame 2 (<Nb bits)
S&P
IF 2 rest
Information Frame 3 (Nb bits)
filler bits

Figure 1-3: An illustration for Ungerboeck's padding scheme. The S&P field stores the synchronization bit and the pointer to the beginning of the information frame assigned to the particular symbol frame.

than the one we propose, the pointer field should consist of at most 2 bits. A 2-bit pointer field can only point to positions that are at most 4 bits away from the pointer field. However, as an example, in the optimal 128-point constellation that we obtained, the average length of bit mappings of constellation symbols is $\bar{L} = 6.3264$. This means that the choices of pointer field sizes that will make Ungerboeck's padding scheme more efficient than ours cannot even correct single symbol shifts between symbol frames.

Chapter 2

Constellation Design

There are two separate optimization problems when constructing the close-to-optimal constellations. The first problem, which we consider in section 2.1 is finding the probability distribution of the amplitudes. A key result of [15] is that an optimal channel input distribution, $f_X(x)$, is a distribution where the values A_X or, equivalently, the energy levels, are in a discrete set, $\mathcal{A}$, while the phases are uniformly distributed. Solving the first problem yields a DACP distribution. The second problem, which we consider in section 2.2, is determining the number and the phases of the quantized points that should be used to represent each of the amplitudes in the first problem in order to yield a discrete constellation. For the first task, we will use a modified version of the cutting plane algorithm introduced in [15]. Our modification arises from a numerical instability we observed when the algorithm is used in high SNRs and relies on an approximation of the channel sensitivity function of the CAWGN channel, which is defined in the upcoming section. For the second task, we use a method in [14] to determine the number of quantization points that we should use for each energy level. In order to determine the phases of the constellation points on each energy level, we introduce a greedy optimization algorithm that aims increasing the smallest distance between two points in consecutive energy levels.

2.1 Approximation to the Cutting Plane Algorithm

Algorithm 3: Cutting Plane Algorithm for Determining Optimal Channel Input Distribution [15]

```
Input: X: Alphabet of possible symbols
       $\mathcal{M}$: Set of possible channel input distributions
Result: Optimal channel input distribution over X
Initialization: Take an arbitrary input distribution, $f_{X_0} \in \mathcal{M}$;
repeat
    i) Find the piecewise-linear approximation of the mutual information:
        $I_n(f_X(x), f_{X_i}(x)) = \min_{0 \leq 1 < n} E_{f_X(x)}\left[g\left(f_X(x), f_{X_i}(x)\right)\right], \; f_X(x) \in \mathcal{M}.$   (2.1)
    ii) Find the next distribution:
        $f_{X_n} = argmax\{I_n(\mu)\}$   (2.2)
until the input distribution converges;
```

A cutting plane algorithm, which is as presented in Alg. 3, to optimize the probability distribution over a fixed set of energy levels, $\mathcal{A}$, was previously proposed [15] and shown to converge considerably more quickly than the Blahut-Arimoto algorithm. We provide a brief overview of that algorithm as applied to our setting. Reference [15] solves the maximization for channel capacity by a cutting plane algorithm that, instead of seeking directly the mutual information maximizing distribution $f_X(x)$, uses a sequence of increasingly tight relaxations, using approximations of the mutual information.

We define the channel sensitivity function $g\left(f_X(x), f_{X_i}(x)\right)$ as follows

$$g\left(f_X(x), f_{X_i}(x)\right) = D\left(f_{Y_{f_X(x)}}(y)||f_{Y_{f_{X_i}(x)}}(y)\right), \tag{2.3}$$

where $D(.||.)$ is the notation for the Kullback-Leibler divergence and $f_{Y_{f_X(x)}}(y)$ is the distribution of the output Y when the input distribution is $f_X(x)$. We begin from an initializing input distribution $f_{X_0}(x) \in \mathcal{M}$. At each iteration n of the algorithm we solve the following approximation to $I(X;Y)$

$$\begin{aligned} & I_n(f_X(x), f_{X_i}(x)) \\ = & \min_{0\leq 1<n} E_{f_X(x)}\left[g\left(f_X(x), f_{X_i}(x)\right)\right], f_X(x) \in \mathcal{M}. \end{aligned} \tag{2.4}$$

One can readily verify that $I_n(f_X(x), f_{X_i}(x)) \geq I(X;Y)$. Each iteration of the algorithm can be expressed as

$$\begin{aligned} & \max c \\ s.t. \quad & E_{f_x(x)}\left[I_n(f_X(x), f_{X_i}(x))\right] \geq c, \\ & 0 \leq i < n, \\ & f_X(x) \in \mathcal{M}. \end{aligned} \tag{2.5}$$

A key feature of the above optimization, proven in [15], is that we can restrict ourselves to a discrete subset of distributions in $\mathcal{M}$ so that the above optimization is finite-dimensional The procedure to use this algorithm with CAWGN channel is explained in [15].

We apply an essential modification to the procedure of [15]. As N_0 decreases, the exponential term in $f_{A_Y|A_X}(a_Y|a_X)$, plummets whereas the modified Bessel function of the first kind increases steeply in (1.6), resulting in numerical issues for high SNR designs. To resolve this issue, we propose using the approximation in 1.1.1. Using this lemma, $I_0(z)$ is replaced with $e^z/\sqrt{2\pi z}$. This provides simplifications when solving (2.5), and removes a numerical indeterminacy in the calculation of $g\left(f_X(x), f_{X_i}(x)\right)$ and the steps for this approximation are provided below:

$$\begin{aligned}
\log\left(\frac{e^{\frac{-|y-x|^2}{N0}}}{\sum_{i=1}^{|M|} p_{\mathcal{A}}(a_i) e^{\frac{-|y|^2-a_i^2}{N_0}} I_0\left(\frac{2|y|a_i}{N_0}\right)}\right) &\approx \log\left(\frac{e^{\frac{-|y-x|^2}{N0}}}{\sum_{i=1}^{|M|} p_{\mathcal{A}}(a_i) e^{\frac{-|y|^2-a_i^2}{N_0}} e^{\frac{2|y|a_i}{N_0}} \frac{1}{\sqrt{4\pi|y|a_i/N_0}}}\right) \\
&= \log\left(\frac{1}{\sum_{i=1}^{|M|} b_i(y,N_0) e^{\frac{-|y|^2-a_i^2}{N_0}+\frac{|y-x|^2}{N0}+\frac{2|y|a_i)}{N_0}}}\right) \\
&= -\log\left(\sum_{i=1}^{|M|} b_i(y,N_0) e^{\frac{-|y|^2-a_i^2}{N_0}+\frac{|y-x|^2}{N0}+\frac{2|y|a_i}{N_0}}\right) \\
&= -\log\left(\sum_{i=1}^{|M|} e^{\ln(b_i(y,N_0))} e^{\frac{-|y|^2-a_i^2}{N_0}+\frac{|y-x|^2}{N0}+\frac{2|y|a_i}{N_0}}\right) \\
&= -\log\left(\sum_{i=1}^{|M|} e^{\left(\ln(b_i(y,N_0))+\frac{-|y|^2-a_i^2}{N_0}+\frac{|y-x|^2}{N0}+\frac{2|y|a_i}{N_0}\right)}\right) \\
&= -\max_k\left\{v_k(y,N_0)\right\} - \log\left(\sum_{i=1}^{|M|} e^{v_i(y,N_0)-\max_k\{v_k(y,N_0)\}}\right) \\
&\approx -\max_k\left\{v_k(y,N_0)\right\},
\end{aligned} \tag{2.6}$$

with a_i being the ith element of the set of energy levels, $\mathcal{A}$; $b_i(y, N_0) = \frac{p_{\mathcal{A}}(a_i)}{\sqrt{4\pi|y|a_i/N_0}}$; $v_k(y, N_0) = \ln(b_k(y, N_0)) + \frac{-|y|^2-a_k^2}{N_0} + \frac{|y-x|^2}{N0} + \frac{2|y|a_k}{N_0}$; and $p_{\mathcal{A}}(a_i) = \int_{|x|=a_i} f_X(x)$. This final result is utilized when calculating the logarithmic expression coming from the Kullback-Leibler divergence in the channel sensitivity function given in (2.3).

2.2 Greedy Quantization

The cutting plane algorithm in the previous section yields a continuous input distribution, which is not a realizable as a channel input distribution due to its continuous nature, hence these continuous energy levels need to be quantized to obtain applicable constellations. It is known that when quantizing these continuous energy levels, the probabilities of points on the same energy level are the same, since the phase information is not preserved over the channel transmission and moreover the arcs between consecutive constellation points on the same energy level are equal in length [14]. Reference [14] proposes three different rules to elect the number of points each ring need to have for the loss of quantization to be small. Among these three rules, the best performing quantization methods suggests using at least k_a number of points for energy level $a \in \mathcal{A}$, with k_a being equal to:

$$k_a = \left\lfloor \frac{\sqrt[3]{a^2 p_{\mathcal{A}}(a)}}{\sum_{\bar{a}\in\mathcal{A}} \sqrt[3]{\bar{a}^2 p_{\mathcal{A}}(\bar{a})}} K \right\rfloor , \tag{2.7}$$

where K is the total number of points we want to have in the constellation, and $p_{\mathcal{A}}(a) = \int_{|x|=a} f_X(x)dx$. Due to the flooring operation, some constellation sizes cannot be achieved for some distributions. In such a case, one can consider the constellation with the smallest number of points that has more than K points in total. The probability of every point on the energy level a is $\frac{p_{\mathcal{A}}(a)}{k_a}$. Since constellation points divide an energy level into arcs of same length, the degree that sees one such arc on energy level a is given by:

$$\Delta\theta_a = \frac{2\pi}{k_a}. \tag{2.8}$$

To specify a point on the coordinate plane, not only its distance from the origin but also the angle it makes with the x-axis is needed. Hence, the exact phases of constellation points need to be identified to describe a constellation. For this purpose we devised a greedy optimization algorithm. This algorithm works as follows: the points on the first energy level are placed starting from the x-axis with intervals of degree $\Delta\theta_a$. The points on the second energy level are initially placed starting from the x-axis just as the innermost level but than rotated in such a way that the smallest distance between a point in the first level and a point in the second level is maximized. After fixing the angle of rotation for the second energy level, the third energy level is placed starting from the x-axis and then rotated to increase the minimum distance between the second level and the third. This process goes on until all the energy levels are placed and possibly rotated.

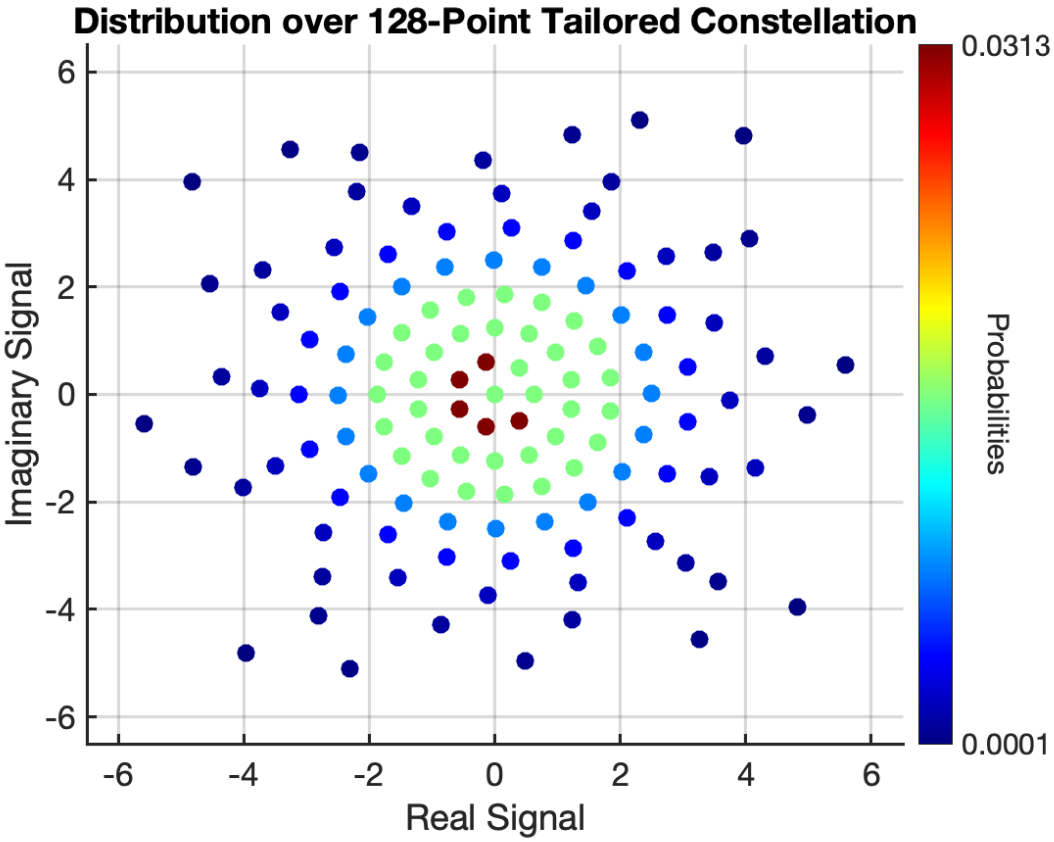

Figure 2-1: An example 128-point constellation design.

An example constellation design using the described method is presented in Fig. 2-1. In order to obtain this constellation, the following parameters are used with this modified cutting plane algorithm,

$$\mathcal{A} = \{x|\ x = 0.6\,k,\ k = 0, \ldots, 10\} \tag{2.9}$$

The constellation design is made for $N_0 = 0.01$. In the design procedure, the average channel input power constraint is set as $\sigma_P^2 = 4$. The choice of $\sigma_P^2 = 4$ is the same as the parameter used in [15] but the overall design SNR is chosen to be higher than the constellations presented there by taking $N_0 = 0.01$. The reason for this higher SNR choice is to target the SNR values of engineering interest where the bit error rate of 128-QAM is around 10^{-4} in the design. In the results presented in the upcoming sections, we show that the proposed constellation design technique is robust and the constellations designed for $N_0 = 0.01$ perform well at lower SNRs. The 64-point constellation design is presented on the right panel of Fig.1-1. In Appendix A, 256-point constellation design together with the parameters to obtain the 64-point and 256-point constellations are presented.

Chapter 3

Modulator Design

Once the OM constellations are designed, the output bits of the channel encoder need to be mapped to the constellation points such that each symbol occurs according to the probabilities in the constellation design. The mapping function should take the channel modulator input distribution and convert it to the distribution of the constellation points. To perform this mapping, we chose using the Huffman shaping method, which is described in the introduction. However directly using Huffman shaping on its own may end up violating the average power constraint we had when designing the constellation since it changes the probabilities of the constellation points to their dyadic approximation. Moreover, fixed length mappings can utilize Gray code to decrease the number of bit errors as it reduces the Hamming distance between the bit mappings of two signals close to one another in the I-Q plane. Last but not least as discussed in the introduction, when using Huffman shaping with fixed message lengths, which is a design choice we adopt in our system, one may end up at a non-leaf node of the binary Huffman code tree. In this section, we provide our solutions to these challenges.

3.1 Power Normalization

The channel model of interest had two constraints on the input and thus the valid channel input distributions are expressed by (1.2). Discretization and assigning the approximate dyadic distribution does not violate the peak power constraint as the energy levels remain unchanged from this operation. However, the average power constraint may be violated due to the changes in the probabilities of the signal points. Define p_i as the probability of symbol i, $\mathbf{s}_i$, after applying the Huffman shaping on the designed constellation. The average power of a constellation of M symbols after the Huffman shaping is equal to:

$$\bar{\sigma} = \sqrt{\sum_{i=1}^{M} p_i ||\mathbf{s}_i||^2} \tag{3.1}$$

To ensure that the average power constraint is satisfied, a normalization is applied to radius of the energy levels. If after the discretization, the average energy constraint is not violated, then we don't have to perform this normalization, in fact normalizing energy levels in such a case may violate the peak power energy constraint. On the other hand, if the average energy constraint is violated this means that the mean energy of the signal points exceed the average energy constraint. In these cases, the factor

$$\frac{\sigma_P}{\bar{\sigma}}, \tag{3.2}$$

is less than 1, with σ_P being the average power constraint in the constellation design, as defined in section 1.1. Hence, normalizing the x and y coordinates of the constellation points with this factor cannot violate the peak power constraint and the resulting constellation becomes a valid input distribution according to the condi-

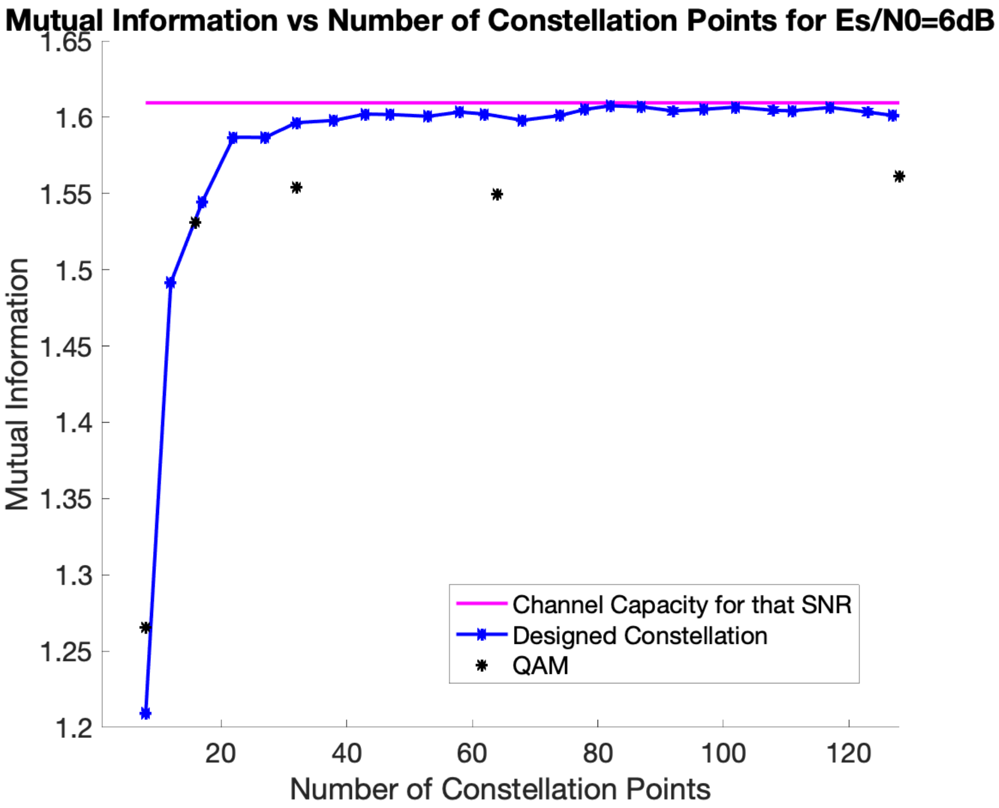

Figure 3-1: The theoretical performance of the constellations after reverse Huffman mapping

tions enforced in the constellation design. In Fig. 3-1, one can observe the theoretical performance of the designed constellations with respect to channel capacity for the channel statistics used in the design. The constellation design parameters used with the cutting plane algorithm presented in the previous chapter to obtain this theoretical performance is the same the parameters used in [15] in order to observe the effect of the quantization and normalization and they can be summarized as:

$$\mathcal{A} = \{0, 1, 2, 3, 4, 5\}, \quad \sigma_P^2 = 4, \quad N_0 = 1. \tag{3.3}$$

The y axis stands for the mutual information between channel input and channel output whereas x axis represents the number of quantization points used to discretize the DACP distribution. The magenta line represents the channel capacity for the complex Gaussian channel that satisfies $N_0 = 1$. The blue line represents the performance of the designed constellations with respect to number of quantization points used to discretize the constellation. Finally, the black dots represent the theoretical performance of 8, 16, 32, 64 and 128 QAM. One can readily see that after 20 quantization points, the designed constellation points exceed the maximum mutual information that QAM can achieve and around 40 quantization points, the performance gets very close to the channel capacity. As a result of our non-linear method of mapping probabilities some small-amplitude ripples occur in the mutual information, yet these do not change the fact that designed constellations can get closer to the channel capacity compared the QAM constellations of the same sizes.

3.2 Error Aware Mapping

Due to the reliance on the varying-length bit strings, the question of how to perform the bit-mapping arises in the system we propose with this thesis. In common communications systems, where the employed modulation scheme performs constant-length bit mappings, Gray code is utilized. We therefore develop a different bit mapping algorithm, suited to OM.

We propose rearranging the bit sequences of the symbols that have the same probability after Huffman shaping such that the two closest constellation points differ in only one bit if they have the same probability. The algorithm sweeps through the constellation points and checks whether the closest constellation point of the currently queried point is mapped to a bit sequence of the same length. If the lengths are the same, then the proposed algorithm checks the Hamming distance between the bit mappings of these sequences. If the Hamming distance is only one, then it moves onto the next constellation point. However, if the Hamming distance is greater than one, it inspects other bit sequences of the same length until an exchangeable bit sequence is detected or all the same-length bit sequences are exhausted. In this codebook fixing process an attempt to fix the bit mappings of nearest neighbour symbols which are mapped to different number of bits is not made as this type of error will trigger the error correction mechanism that will be discussed in the next chapter, and hence dealt separately from the errors that do not cause insertions or deletions.

Let the list of the bit sequences that have the same length as the current constellation point be denoted by $\mathcal{L}$. Suppose that the closest neighbor of the symbol $\mathbf{s}$, call it $\mathbf{s}_j$, is more than one Hamming distances away from $\mathbf{s}$ and the algorithm found out that bit sequence mapped to $\mathbf{s}_j$, denote it by l_j, is in $\mathcal{L}$. Assume that l_i is the bit sequence of symbol $\mathbf{s}_i$ such that $l_i \in \mathcal{L}$. The algorithm first checks l_i's Hamming distance from the mapping of $\mathbf{s}$. If Hamming distance is more than one, then it moves onto the next bit mapping in $\mathcal{L}$. If the distance is one bit, it checks whether l_i is exchangeable with l_j. The bit sequence l_i is deemed as exchangeable if it satisfies one of the following criteria:

- The Hamming distance between the bit mappings of $\mathbf{s}_i$ and its closest neighbor is more than one.
- The Hamming distance between the bit mappings of $\mathbf{s}_i$ and its closest neighbor is one, but the Hamming distance between the mapping of $\mathbf{s}_j$ and the closest neighbor of $\mathbf{s}_i$ is also one.

If one of the above conditions is met, the algorithm swaps the bit sequences corresponding to $\mathbf{s}_j$ and $\mathbf{s}_i$. If no exchangeable symbol is encountered despite having its nearest neighbor more than one bit away, the algorithm does not change anything and moves onto the next point.

3.3 Padding

As previously stated, in the system model we propose, the length of the bit sequence that arrives at the modulator is modelled as fixed. With a fixed number of bits to be transmitted, the last tree traversal in the Huffman shaping method may stop before reaching a leaf. In such cases, the following simple padding scheme, which GRAND will avail of for length correction, is proposed. If the mapping process stops at a non-leaf node, first the branch that is labelled with a "1" is followed and then the 0-labelled branches are followed until a leaf is reached. Reading backwards from the end of the symbol to the first 1 reveals how many bits were used to pad. This information is used in the GRAND-assisted demodulation described in the next section. If there is no bit in the original bit string that isn't mapped to any symbol, padding starts at the root of the tree.

An illustration of this process is presented in Fig. 3-2. If the bit sequence that arrived at the modulator is "1110111", then the symbols 7 and 4 are transmitted. The last bit does not reach a leaf and stops at node 37 of the tree. The proposed padding scheme dictates following the "1" branch to node 33 and the "0" branches are followed from there, leading to the transmission of symbol 15.

The proposed padding scheme is simple and effective when combined with the GRAND-based demodulator. The location of the last 1 indicates the end of the original message, hence the padding scheme provides the length of the original message. If the last received symbol is demodulated in error the location of the last 1 may

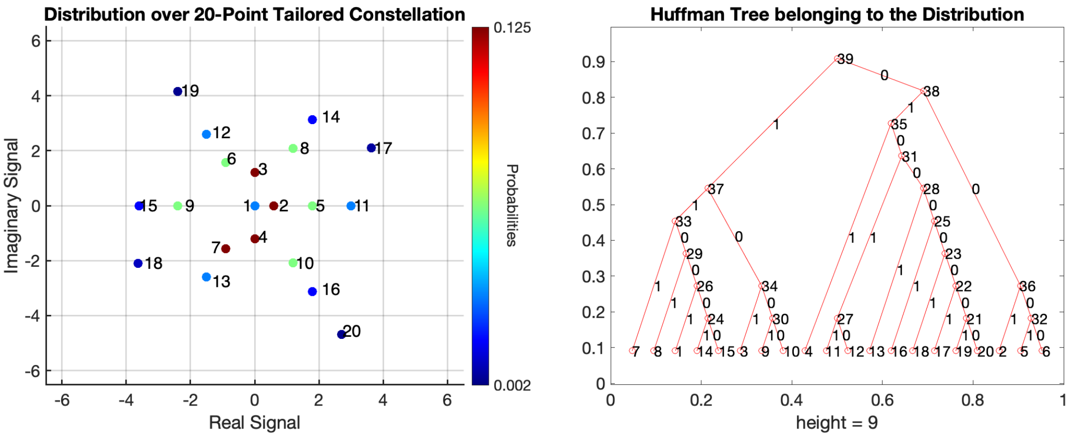

Figure 3-2: A toy constellation design

change, but we establish that the padding frequency can be designed so that this happens sufficiently infrequently so as not to harm performance.

One small detail to note occurs when the message bit sequence is mapped to a symbol sequence without any leftover bits. In that case, we pad starting from the root of the Huffman tree. In the example in fig. 3-2, the bit sequence that will be used for padding if there is no unmapped bit, is "1000". This means that the last transmitted symbol in such cases is symbol 10. The reason behind this choice will be explained in more detail in the next chapter of this thesis but to give a rough idea, the position of the last 1 in the bit sequence of the last symbol is important in determining the length of the received bit sequence and consequently it should be ensured that the last 1 in the final symbol comes from padding.

This scheme leads to variable length padding. In the worst case scenario, the number of overhead bits for a message is equal to the depth of the Huffman tree and the number of symbols used for overhead is at most one, yet, this is rarely the case. For instance, for the 128-point OM constellation design, average number of overhead bits turn out to be 3.5, which corresponds to almost the half of the average bit sequence length of this constellation.

Chapter 4

Demodulator Design and Length-Based Correction

There are two main problems that the demodulator needs to answer. First, the demodulator needs to form its decision regions according to the prior probabilities of the signals in the constellation. By using the Bayes' Theorem, determining decision regions is simple and we give a brief overview of the obtained decision regions using this method in section 4.1. The second problem, which can be considered as the most vital challenge when using varying-length constellation shaping methods, is preventing or correcting the possible insertions or deletions in the transmission. To solve this latter problem, in section 4.2 we propose a novel guess-based length correction mechanism that detects and corrects insertion and deletion errors when such errors are caused by a single symbol error. We choose transmission lengths used in the proposed systems such that the probability of having more than one symbol errors that result in insertion and deletion is sufficiently low and hence the length correction mechanism can successfully identify and correct these tricky situations.

Section 4.2 also contains the full discussion on how to choose transmission lengths in the proposed system.

4.1 Decision Regions

Let's revisit the channel model. As mentioned in section 1.1, Y is the channel output, X is the channel input and N is the zero-mean, circularly symmetric complex Gaussian that satisfies $E[|N|^2] = N_0$. Also let $\mathbf{s}_i$ be the ith constellation symbol. Unlike the case in the uniform constellations, with the constellation design described in the previous sections, X no longer comes from a uniform distribution but it is distributed according to the “Huffman approximation distribution".

The demodulation process needs to incorporate the non-uniform prior distribution of the symbols in the constellations suggested in this paper. To achieve this, we can use Bayes' Theorem. As the priors in the uniform constellations are uniform, maximum likelihood (ML) and maximum a posteriori (MAP) demodulations yield the same results. Yet, in the non-uniform constellation, the a priori probabilities differ which results in a difference between ML and MAP estimations. In the MAP estimation, for a channel output to be labeled as symbol $\mathbf{s}_i$, the following inequality must be satisfied:

$$\frac{f_{Y|X}(\mathbf{y}|\mathbf{s}_i)P(X=\mathbf{s}_i)}{f_{Y|X}(\mathbf{y}|\mathbf{s}_j)P(X=\mathbf{s}_j)} \geq 1 \quad \forall j \neq i \tag{4.1}$$

Or equivalently,

$$\frac{f_N(\mathbf{y}-\mathbf{s}_i)P(X=\mathbf{s}_i)}{f_N(\mathbf{y}-\mathbf{s}_j)P(X=\mathbf{s}_j)} \geq 1 \quad \forall j \neq i, \tag{4.2}$$

where $f_N(\mathbf{n})$ is the transition function of a channel and in our setting it is as follows:

$f_N(\mathbf{n}) = \frac{1}{\pi N_0} exp(-\frac{1}{N_0}|\mathbf{n}|^2)$. Then, equation (16) becomes,

$$\frac{exp\left(-\frac{1}{N_0}|\mathbf{y}-\mathbf{s}_i|^2\right) P(X=\mathbf{s}_i)}{exp\left(-\frac{1}{N_0}|\mathbf{y}-\mathbf{s}_j|^2\right) P(X=\mathbf{s}_j)} \geq 1 \quad \forall j \neq i \tag{4.3}$$

Let's study the boundary between symbol i and any other symbol k for convenience. Using the monotonicity of the logarithm,

$$\begin{aligned} &log\left(\frac{exp\left(-\frac{1}{N_0}|\mathbf{y}-\mathbf{s}_i|^2\right) P(X=\mathbf{s}_i)}{exp\left(-\frac{1}{N_0}|\mathbf{y}-\mathbf{s}_k|^2\right) P(X=\mathbf{s}_k)}\right) \\ &\qquad = log\left(\frac{P(X=\mathbf{s}_i)}{P(X=\mathbf{s}_k)}\right) + \left(-\frac{1}{N_0}|\mathbf{y}-\mathbf{s}_i|^2\right) - \left(-\frac{1}{N_0}|\mathbf{y}-\mathbf{s}_k|^2\right) \geq 0 \end{aligned} \tag{4.4}$$

After cancellations and reorderings of the terms, we obtain the following inequality:

$$log\left(\frac{P(X=\mathbf{s}_i)}{P(X=\mathbf{s}_k)}\right) - \frac{1}{N_0}(a_i^2 - a_k^2 + b_i^2 - b_k^2) \geq a\frac{-2a_i+2a_k}{N_0} + b\frac{-2b_i+2b_k}{N_0} \tag{4.5}$$

where $\mathbf{y} = (a,b)$, $\mathbf{s}_i = (a_i, b_i)$ and $\mathbf{s}_k = (a_k, b_k)$. The boundary between the decision regions of symbol i and symbol j is expressed as the equality version of the inequality in (4.5). The decision region of symbol i is expressed with all the inequalities of the form (4.5) where $k \neq i$ and $\mathbf{s}_k$ is a symbol in the constellation of interest. Applying this to the constellation on figure 1-1, we obtain the regions in figure 4-1. Each decision region is colored with a different shade of cyan and magenta dots are the constellation points. By observing figure 4-1, one can verify that the decision boundaries do not split the line between two constellation points equally but rather the distances between them and the constellation points depend on the prior probabilities of the points.

The demodulator calculates $f_{Y|X}(\mathbf{y}|\mathbf{s}_i)P(X = \mathbf{s}_i)$ for each of the received symbols and decides in which decision region the received signal lies depending on for which i $f_{Y|X}(\mathbf{y}|\mathbf{s}_i)P(X = \mathbf{s}_i)$ is maximized. This step is independent of the bit sequences and, therefore, after this initial demodulation the bit sequences we obtain may differ from the transmitted bit sequence in terms of length if some signal lies in the decision region of another symbol with a bit sequence of different length. The next section provides a methodology to undertake this situation. The decision regions for the 128-point constellation in Fig. 2-1 and the 256-point constellation design is provided in Appendix A.

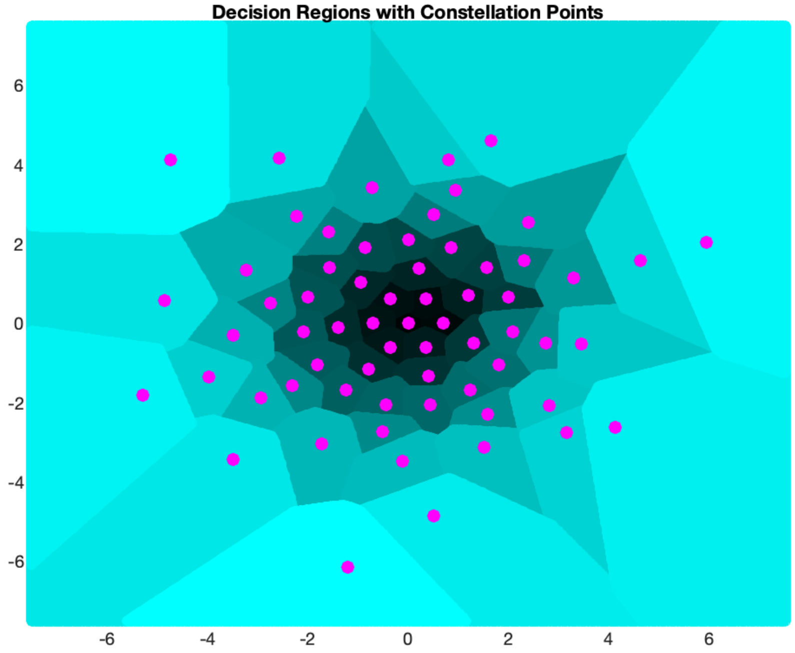

Figure 4-1: The decision regions for the constellation given in the right panel of Fig. 1-1.

4.2 Length-Based Correction

As a result of the varying bit lengths of symbols, insertions and deletions can result from erroneous demodulation. By adapting GRAND to use the padding as a form of an error correcting code, one can readily correct many of these length errors. The proposed procedure is presented in Alg. 4.

```
Algorithm 4: Demodulation and Length Correction
Input: $\mathbf{y}$: channel output
       C: constellation points and their probabilities
       f: symbol to bit mapping
       $N_0$: noise power
       n: number of transmitted bits
Result: $\hat{\mathbf{x}}$: demodulated bit sequence
$\hat{\mathbf{y}} \leftarrow$ MAP demodulation
$\hat{\mathbf{x}} \leftarrow$ bit sequence corresponding to $\hat{y}$
if $\hat{x}.length \neq n$ then
    $i \leftarrow 0$, r $\leftarrow$ likelihood order of symbols in $\hat{\mathbf{y}}$
    while $i < \hat{\mathbf{y}}.length$ do
        $\mathcal{P} \leftarrow$ set of constellation points on the same ring and the neighboring
          rings
        p $\leftarrow$ likelihood order of elements in $\mathcal{P}$
        for $j$= 1 to $\mathcal{P}.length$ do
            $\hat{\mathbf{x}} \leftarrow$ bit sequence of $\hat{\mathbf{y}}[r(i)] = \mathcal{P}(p(j))$
            if $\hat{\mathbf{x}}.length == n$ then
                return $\hat{\mathbf{x}}$
        end
        $i \leftarrow i + 1$
    end
end
return bit sequence corresponding to $\hat{y}$
```

We use the term “message" to indicate the information bit sequence and “transmitted bit sequence" to include the padding bits. In the proposed system, the message length is constant and known by both the transmitter and receiver. However, the number of bits in the transmitted sequence changes based on the number of padding bits. Having removed the padding bits at the receiver, if the length of the remaining bits is not equal to the agreed message length, then at least one demodulated symbol resulted in an insertion or deletion.

Our goal in setting message lengths is to ensure that the likelihood of having more than one insertion or deletion error between padded symbols be made negligible. For a given constellation and E_S/N_0, the probability of an insertion or deletion in demodulation, $p_{\text{indel}}(E_S/N_0)$, can be evaluated with theory or through simulation. As symbol errors resulting in length changes occur independently, the number of transmitted symbols until an error with length change is geometrically distributed with probability $p_{\text{indel}}(E_S/N_0)$. Namely, let L_n be the random variable corresponding to the number of symbols errors with length change in a message block of length n. Then, the probability of having more than one symbol error that causes length change is expressed by

$$P(L_n > 1) = 1 - \binom{n}{0}(1 - p_{\text{indel}}(E_S/N_0))^n - \binom{n}{1} p_{\text{indel}}(E_S/N_0)^1 (1 - p_{\text{indel}}(E_S/N_0))^{n-1}. \tag{4.6}$$

Using this model, we set the message length such that $P(L_n > 1)$ is bounded above by $a(E_S/N_0)\, p_{\text{indel}}(E_S/N_0)$ where $a(E_S/N_0)$ is a small tunable parameter of the order 10^{-2}. The value of the $a(E_S/N_0)$ is determined with a search to minimize the BER of the final binary system. All the results that will be presented in the next section are obtained after fixing the $a(E_S/N_0)$ values. Message lengths for the reported simulations are presented in Table 1 for the 128-point constellation design in Fig. 2-1. The message bit lengths are obtained by multiplying the number of symbols found via the above model with the weighted average length of the bit mappings.

In this design, message lengths increase with E_s/N_0, so that the rate of the overhead decreases with increasing SNR. As SNR goes to infinity, the overhead becomes infinitesimally small compared to the transmitted block length and the rate of the proposed scheme converges to one. In Table 1, the last column illustrates this change in rate for the 128 point constellation. When an insertion or deletion does occur, it is most likely that there is a single erroneous symbol and that results in shifting to a symbol on a neighboring ring, i.e. one energy level up or down. An ability to find the symbol that is in error and correcting it, or at least replacing it with one of the correct length, decreases both symbol and bit error rates. At this point, we draw inspiration from the recently introduced GRAND [9, 26, 8].

First, the demodulator estimates the transmitted symbols via the decision regions given by Bayes' theorem and then converts the resulting symbols to a binary string. By construction, the bit sequence of the last symbol needs to contain a one. If there is none, then the last symbol is demodulated incorrectly. An attempt can be made to correct the error by replacing the last symbol with the second most probable symbol for the last received signal.

The message length that is to be received is known to the receiver. The demodulator compares the length of the message with the known constant. If the lengths are not the same, then demodulated symbols are listed from least reliable to most reliable. Starting with the least reliable symbol, the demodulator examines the length of

Table 4.1: Some Message Lengths for 128-point Constellation Design

E_b/N_0 (dB)	Ave. num. bits (N_b)	Num. symbols (N_s)	Rate
20.00	1594	252	0.998
18.75	771	122	0.995
17.50	177	28	0.979
16.25	56	9	0.934

the bit sequence if it switches the demodulated symbol to a symbol on its own ring or a neighboring ring. If there is a bit sequence of the correct length, then the original demodulated symbol is swapped with the most likely alternate symbol on these specified energy levels and the resulting symbol sequence is returned for demodulation. If there is no bit sequence of the correct length, the demodulator proceeds to the next least reliable demodulated symbol. This process continues until a bit sequence of the correct length is found or all the demodulated symbols are exhausted. In the latter case, the original demodulated message bits are returned without any change.

Consider the Huffman tree and the message bit sequence introduced in the example with Fig. 3-2. Suppose that after detection, the demodulator formed the received symbol sequence as $9-4-15$ which corresponds to the bit sequence "1001011110000". After separating this bit sequence from padding, the remaining bits are "10010111". The expected message length is 7, but the demodulator has 8 bits hence it attempts error correction. The received symbols are ordered according to their reliability. If symbol 9 was labelled as the least reliable symbol, the symbols on the same and the neighboring rings are listed as alternatives. According to the constellation on the left panel of Fig. 3-2, the list in this example is $\{5, 6, 7, 8, 10, 11, 12, 13\}$. The elements of this list are ordered according to their likelihoods and new bit sequences are formed by swapping symbol 9 with the elements of this list according to their likelihoods. When the demodulator finds a bit sequence with length 7, it terminates its search and outputs the resulting bit sequence. In this example swapping symbol 9 with symbol 7 yields the correct length, thus the demodulator outputs the bit sequence "1110111" correctly.

Chapter 5

Overall System Performance

In this section, the performance of the proposed OM constellation design with GRAND-like length correction is presented in comparison with QAM constellations of identical sizes. QAM is a standard in many applications and the BER and SER performances of various-size QAM constellations are readily available in MATLAB using the built-in function “berawgn”. It is useful to provide a definition for the energy per bit over noise spectral density (E_b/N_0) for the upcming results. In a usual modulation scheme, we use the following relation

$$\frac{E_s}{N_0} = \frac{E_b}{N_0} \log_2(K) \tag{5.1}$$

between symbol energy per noise spectral density (E_s/N_0) and bit energy per noise spectral density (E_b/N_0) with K being the size of the constellation. The number of bits used to represent symbols may differ in the designed constellation, and hence, the conventional equation to calculate the E_b/N_0 does not hold. With $N_s(E_s/N_0)$ being the average number of symbols for a message plus padding bits, at most one symbol is used for the padding overhead. Hence, the average energy of a single information symbol is upper bounded by $E_s + \frac{E_s}{N_s(E_s/N_0)}$. As a result the following relation is used to obtain E_b/N_0:

$$\frac{E_s + \frac{E_s}{N_s(E_s/N_0)}}{N_0} = \frac{E_b}{N_0} \sum_{i=1}^{K} -p_i \log_2(p_i), \tag{5.2}$$

where p_i is the probability of the constellation point i.

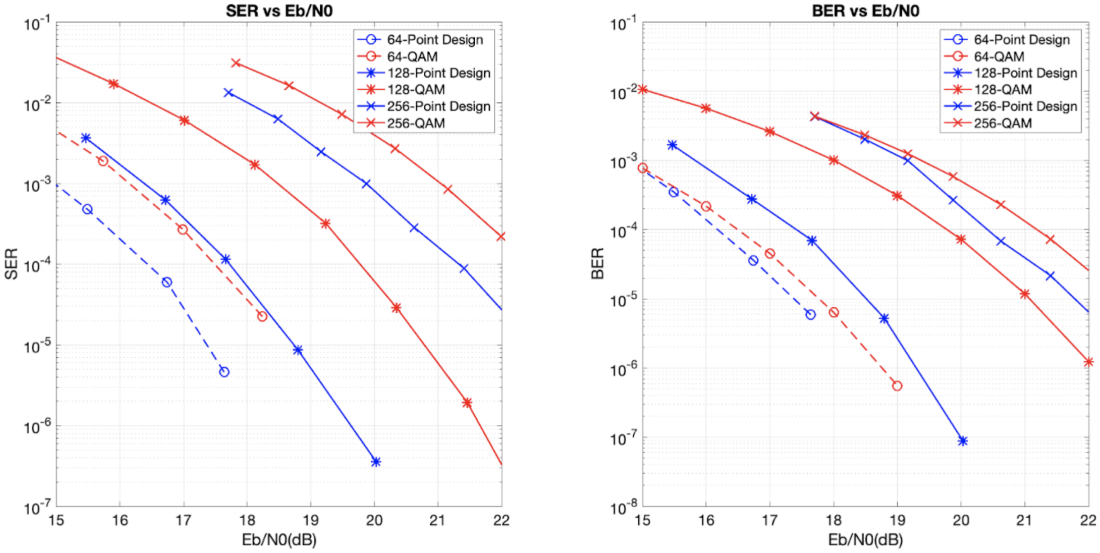

Figure 5-1: SER and BER performances of the overall system in comparison with QAM

Fig. 5-1 displays the overall system performance for some of the commonly used constellation sizes in comparison with the QAM constellations of the same sizes. It can be observed that using OM yields a gain in all the presented constellation sizes compared to the performance of QAM constellations of the same sizes in the SNR region of engineering interest. In particular, the 128-point constellation displays a significant gain of about 2 dB compared to the 64-point and 256-point constellation designs. We observed that the performance of the 64-point and 256-point constellation design suffers from the cutting plane algorithm which determines the optimal continuous probability distribution over a fixed set of energy levels that are provided as an input to the algorithm. The fact that this algorithm does not build an optimal set of energy levels, but instead optimizes the probabilities over the provided fixed set of energy levels may result in suboptimal distributions as seen in the current 64-point and 256-point designs in Fig. 5-1.

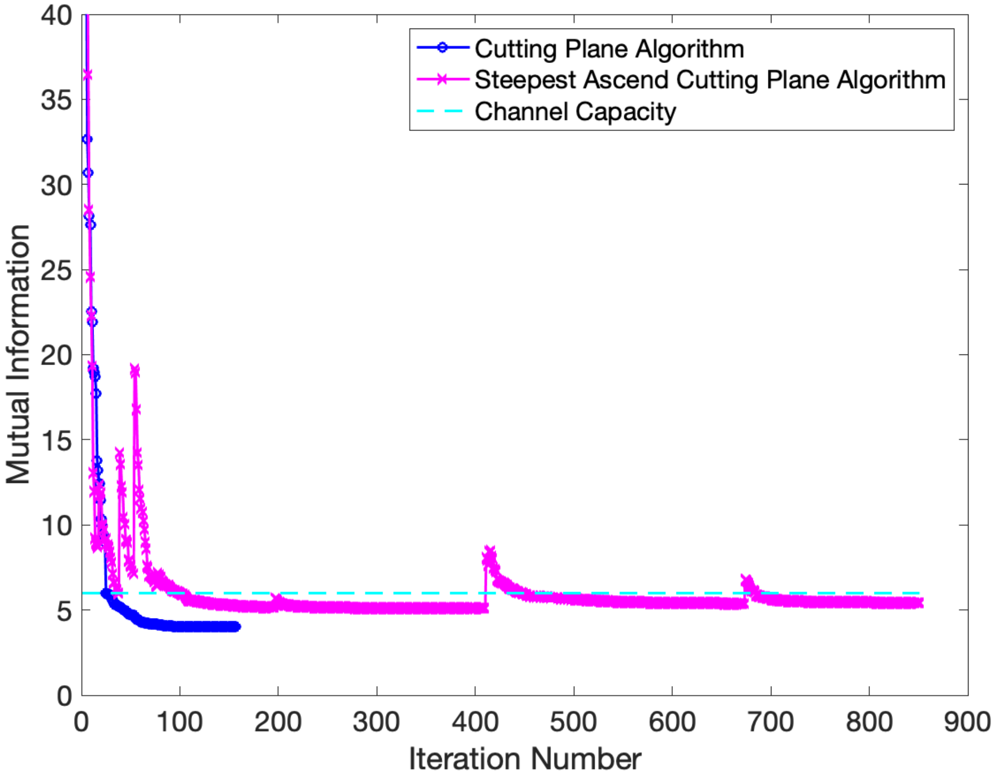

Figure 5-2: Illustration of the change in the attainable mutual information between the channel output and the channel input distribution found during the iterations of the modified cutting plane algorithm and the modified steepest ascend cutting plane algorithm with respect to the iteration number.

Due to this shortcoming of the cutting plane algorithm, some of our future work include enhancing this constellation design process in order to improve the OM gains for all constellation sizes. A promising lead in this direction is presented in Fig. 5-2. Reference [15], introduces another algorithm which is referred to as the "Steepest Ascend Cutting Plane Algorithm" which builds the input alphabet while also optimizing the probability distribution over the dynamic set of energy levels. After performing modifications that are similar to those that we applied to the original cutting plane algorithm, we observed the changes in the mutual information between the channel input distribution found during the iterations of the modified steepest ascend cutting plane algorithm and the channel output. This is depicted in Fig. 5-2 with the magenta curve. We compare this with the the mutual information between the channel input distribution found during the iterations of the modified cutting plane algorithm and the channel output, which is given with the dark blue curve. The cyan dashed-line is the channel capacity at the design SNR. From Fig. 5-2, it is clear that the constellations obtained via the steepest ascend cutting plane algorithm can get closer to the channel capacity, after sufficiently many plane cuts, compared to the constellations designed via the cutting plane algorithm. The only disadvantage of the steepest ascend algorithm is that it takes more iterations for the algorithm to converge, but this optimization is performed only once, hence this property is not a real drawback against the overall system performance. We are currently working on utilizing the new constellation designs obtained this way with the proposed modulator and demodulator.

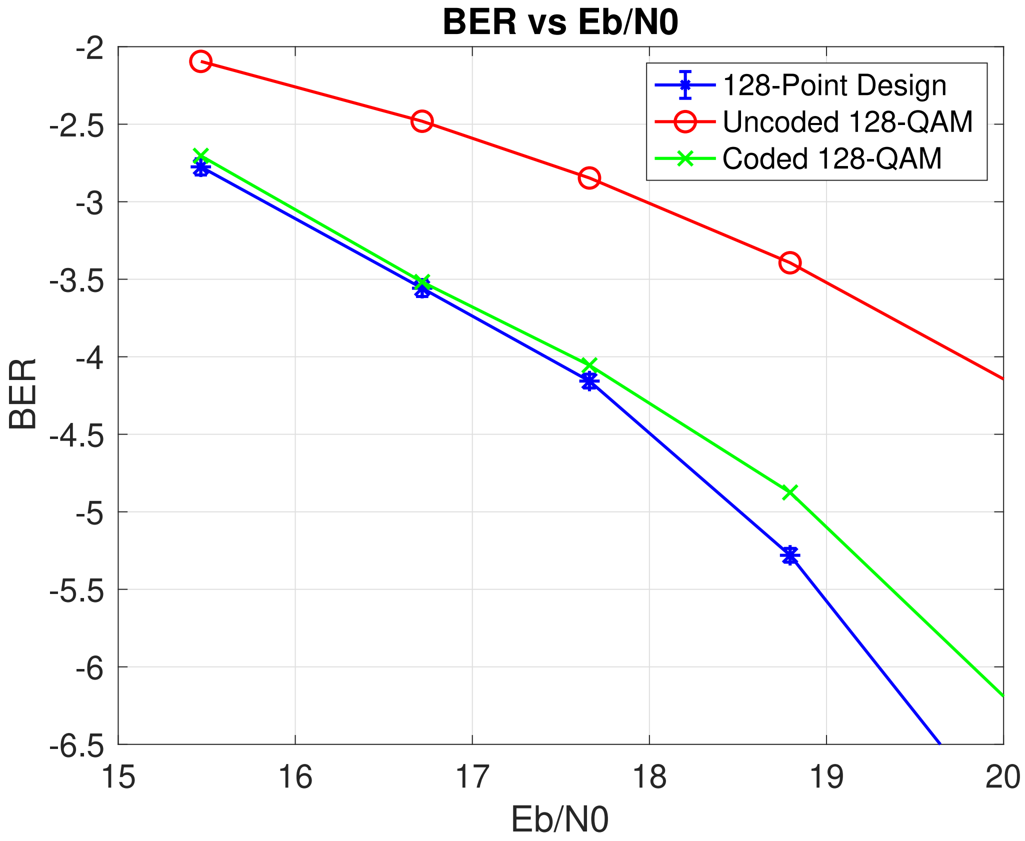

Figure 5-3: BER performance comparison of the 128-point OM design with uncoded 128 QAM and 128-QAM with an additional LDPC code for binary error correction. Due to the varying-length overhead in the proposed scheme, our 128-point design has different rates at different SNRs. The coded 128-QAM results at each SNR are obtained by using LDPC codes of the same average rate as the design at that particular SNR

We note that the 128-point OM constellation design can achieve a gain of about 2 dB both in SER and BER with the proposed length correction demodulator and that this gain is of the order that is typically attained through the use of computationally involved Forward Error Correction (FEC) codes. To further assess that observation, we compared the BER performance of the proposed 128-point OM design and 128-QAM employing Low Density Parity Check (LDPC) codes. As the rate of the proposed scheme depends on the SNR, for the QAM with LDPC to be comparable with the proposed scheme a different LDPC code that has the same rate and the same block length as the proposed scheme is used, at each SNR. For instance, at $E_b/N_0 = 20\ dB$, the LDPC code used in Fig. 5-3 has a message length of 1594 bits and a rate of 0.998 and at $E_b/N_0 = 17.50\ dB$, the LDPC code that is used has a message length of 177 bits and a rate of 0.979. More message lengths and rates can be found in Table 4.2. We use the repeat-accumulate LDPC code design described in [16]. The decoder used with these LDPC codes is the built-in min-sum LDPC decoder of MATLAB. The results displayed in Fig. 5-3 confirm the earlier finding that using OM with a simple padding scheme and GRAND-style length correction results in a final BER performance that is as good as using computationally involved FEC schemes as an outside wrapper to standard modulation.

Another interesting future work direction is what happens when a GRAND-based demodulator that considers all the possible single-symbol-swaps instead of the lightweight version where it only considers the swapping a symbol with another symbol if they are on the same energy level or on the neighbouring energy levels. We have recently implemented this more comprehensive demodulator and currently working on how using this new demodulator affects the complexity of the demodulation and the overall length-correction capability of the system.

Chapter 6

Conclusion

In this thesis, we provide a system for making optimum modulation practical. We present a design procedure to obtain non-uniform constellations according to channel statistics. It was already known in theory that optimal modulation schemes are non-uniform, in the sense that symbol transmission distributions are non-uniform, and that they can perform better in terms of capacity than the commonly used uniform modulation schemes. The proposed design provides modulation and associated demodulation schemes that can significantly surpass the performance of the commonly used modulation schemes such as QAM. While OM is expected to be capable of providing significant SER benefits over commonly used schemes, to translate that to BER gains requires a method that can resolve insertion and deletion errors. Here we establish a simple, low-overhead, and computationally light mechanism to translate that gain to BER. Our method achieves this with a simple padding approach and a novel light-weight GRAND decoder, resulting in a significant improvement of the order of 2 dB that is transparent to the final binary data.

As for the future work, the proposed system will be extended using the Steepest

Ascend Cutting Plane Algorithm which, instead of optimizing over a fixed set of energy levels, builds the set of energy levels while also optimizing the probability over these energy levels. Instead of using a light-weight approach in GRAND-assisted demodulation which considers only a subset of possible symbol swaps based on the energy levels, we are currently working on a GRAND demodulator that considers all length-correcting symbol swaps. In addition to this, the question of whether the proposed system can be extended to multiple-input multiple-output channel models is an interesting question that will be considered in the further stages of the work.

Appendix A

Figures

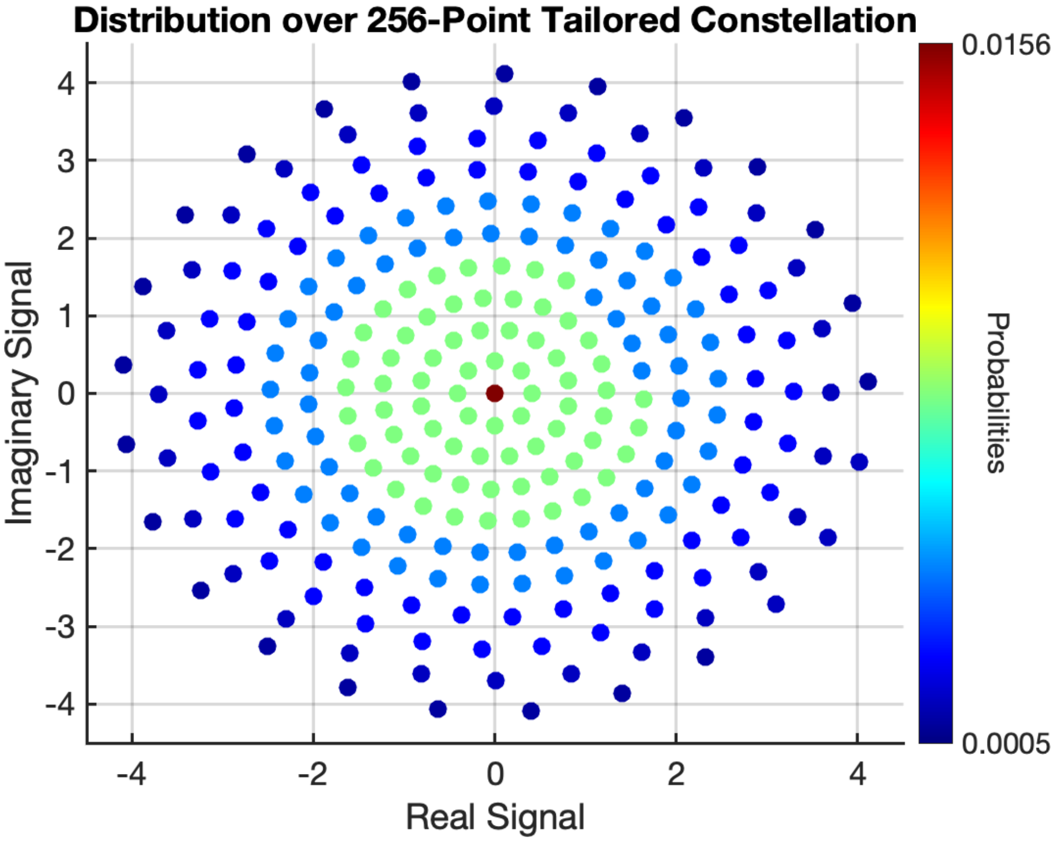

Figure A-1: 256-point constellation design

The parameters to obtain the 64-point constellation in Fig. 1-1:

$$\mathcal{A} = \{x |\ x = 0.7\,k,\ k = 0, \ldots, 10\},\ N_0 = 0.01,\ \sigma_P^2 = 4 \tag{A.1}$$

The parameters to obtain the 256-point constellation in Fig. A-1:

$$\mathcal{A} = \{x |\ x = 0.4\,k,\ k = 0, \ldots, 10\},\ N_0 = 0.01,\ \sigma_P^2 = 4 \tag{A.2}$$

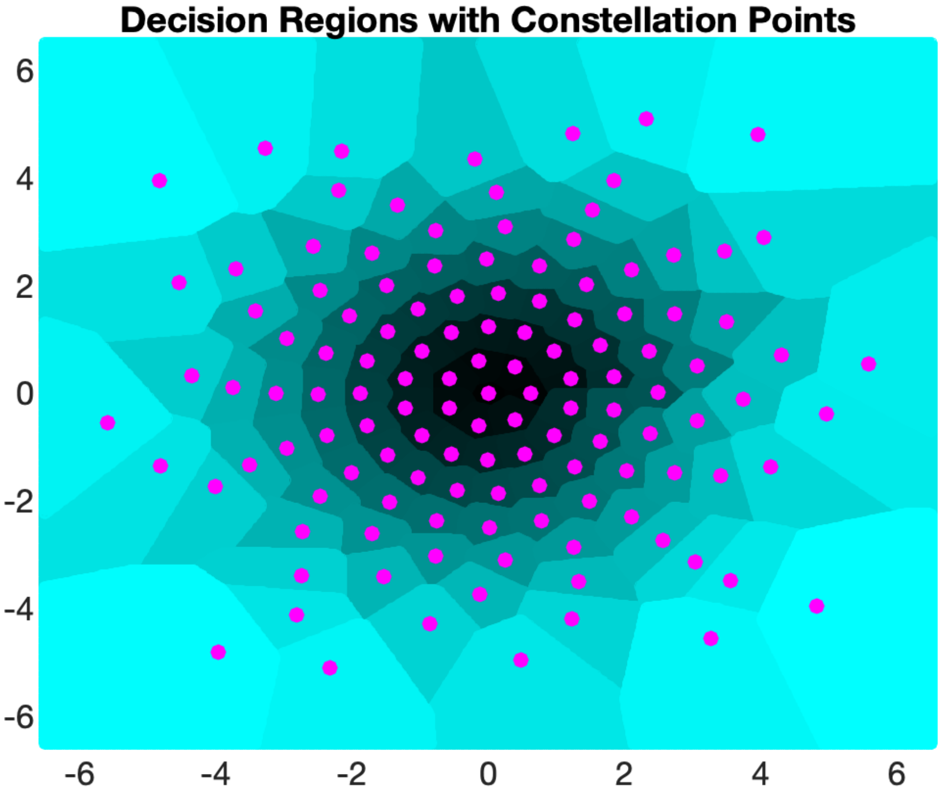

Figure A-2: The decision regions for the 128-point constellation design in Fig. 2-1.

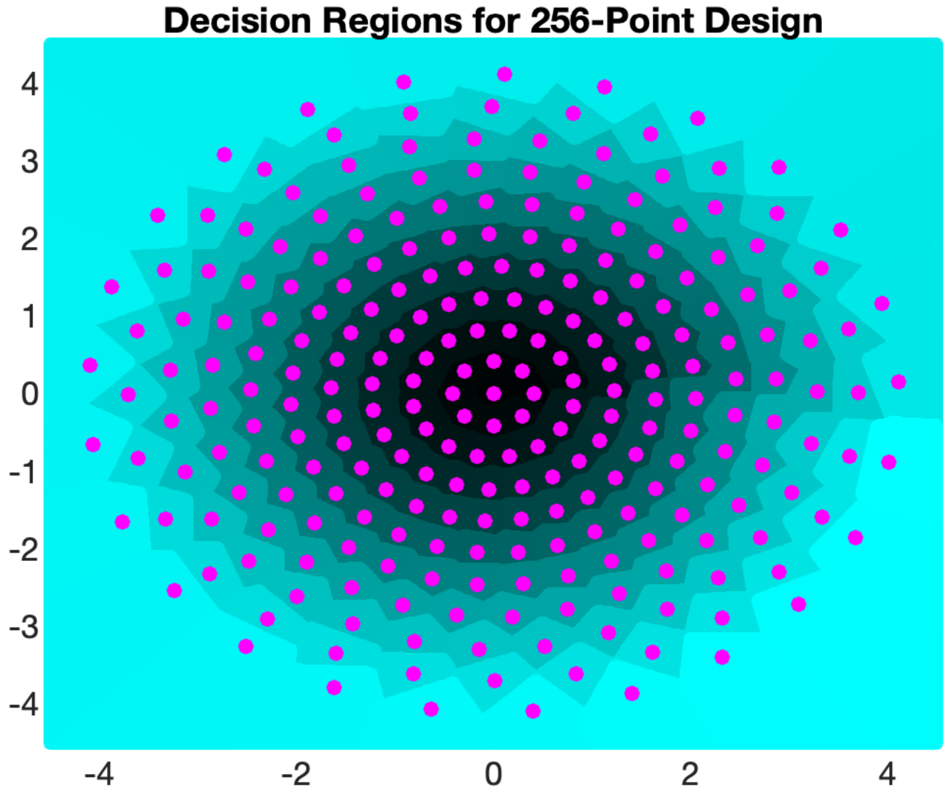

Figure A-3: The decision regions for the 256-point constellation design in Fig. A-1.

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, page 377. Dover, New York City, 1964.

[2] R. A. Amjad and G. Böcherer. Fixed-to-variable length distribution matching. In *IEEE ISIT*, pages 1511–1515, 2013.

[3] S. Baur and G. Boecherer. Arithmetic distribution matching. In *SCC 2015*, pages 1–6, 2015.

[4] G. Böcherer and R. Mathar. Matching dyadic distributions to channels. In *2011 DCC*, pages 23–32, 2011.

[5] T. H. Chan, S. Hranilovic, and F. R. Kschischang. Capacity-achieving probability measure for conditionally Gaussian channels with bounded inputs. *IEEE Trans. Inf. Theory*, 51(6):2073–2088, 2005.

[6] J. Conway and N. Sloane. A fast encoding method for lattice codes and quantizers. *IEEE Trans. Inf. Theory*, 29(6):820–824, 1983.

[7] T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[8] K. R. Duffy. Ordered reliability bits guessing random additive noise decoding. In *IEEE ICASSP*, pages 8268–8272, 2021.

[9] K. R. Duffy, J. Li, and M. Médard. Capacity-achieving guessing random additive noise decoding. *IEEE Trans. Inf. Theory*, 65(7):4023–4040, 2019.

[10] G.D. Forney. Multidimensional constellations. ii. Voronoi constellations. *IEEE J. Sel. Areas Commun.*, 7(6):941–958, 1989.

[11] G.D. Forney. Trellis shaping. *IEEE Trans. Inf. Theory*, 38(2):281–300, 1992.

[12] R. G. Gallager. *Principles of Digital Communication.* Cambridge University Press, 2008.

[13] M. C. Gursoy, H. V. Poor, and S. Verdu. The noncoherent Rician fading channel-part i: structure of the capacity-achieving input. *IEEE Trans. Wirel. Commun.*, 4(5):2193–2206, 2005.

[14] W. Huleihel, Z. Goldfeld, T. Koch, M. Madiman, and M. Médard. Design of discrete constellations for peak-power limited complex Gaussian channels. In *IEEE ISIT*, 2018.

[15] J. Huang and S. P. Meyn. Characterization and computation of optimal distributions for channel coding. *IEEE Trans. Inf. Theory*, 51(7):2336–2351, 2005.

[16] S. J. Johnson. Introducing Low-Density Parity-Check Codes. Tech. Rep., University of Newcastle, 2000.

[17] M. Katz and S. Shamai. On the capacity-achieving distribution of the discrete-time noncoherent and partially coherent AWGN channels. *IEEE Trans. Inf. Theory*, 50(10):2257–2270, 2004.

[18] F.R. Kschischang and S. Pasupathy. Optimal nonuniform signaling for Gaussian channels. *IEEE Trans. Inf. Theory*, 39(3):913–929, 1993.

[19] W. Oettli. Capacity-achieving input distributions for some amplitude-limited channels with additive noise. *IEEE Trans. Inf. Theory*, 20(3):372–374, 1974.

[20] J. G. Proakis and M. Salehi. *Fundamentals of communication systems.* Pearson Prentice Hall, Upper Saddle River, N.J, 2005.

[21] S. Shamai. Capacity of a pulse amplitude modulated direct detection photon channel. *Proc. Inst. Elec. Eng.*, 137:424–430, 1990.

[22] S. Shamai and I. Bar-David. The capacity of average and peak-power-limited quadrature Gaussian channels. *IEEE Trans. Inf. Theory*, 41(4):1060–1071, 1995.

[23] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[24] P. K. Singya, P. Shaik, N. Kumar, V. Bhatia, and M. Alouini. A survey on higher-order QAM constellations: Technical challenges, recent advances, and future trends. *IEEE Open J. Commun. Soc.*, 2:617–655, 2021.

[25] J. G. Smith. The information capacity of amplitude- and variance-constrained scalar Gaussian channels. *Inf. Control.*, 18(3):203–219, 1971.

[26] A. Solomon, K. R. Duffy, and M. Médard. Soft maximum likelihood decoding using GRAND. In *IEEE ICC*, 2020.

[27] A. Tchamkerten. On the discreteness of capacity-achieving distributions. *IEEE Trans. Inf. Theory*, 50(11):2773–2778, 2004.

[28] G. Ungerboeck. *Huffman Shaping*, pages 299–313. Springer US, Boston, MA, 2002.